Proceedings of the Third International Workshop on Computational Social Choice

Vincent Conitzer & Jörg Rothe (eds.)

COMSOC 2010
Düsseldorf

September 13-16, 2010

düsseldorf university press

d|u|p

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf

Proceedings of the Third International Workshop on Computational Social Choice (COMSOC-2010)

Vincent Conitzer & Jörg Rothe (eds.)

Printed by Düsseldorf University Press

Program Committee

Felix Brandt	TU München, Germany
Vincent Conitzer (co-chair)	Duke University, Durham, NC, USA
Edith Elkind	Nanyang Technological University of Singapore, Singapore
Ulle Endriss	University of Amsterdam, The Netherlands
Piotr Faliszewski	AGH University Krákow, Poland
Michael R. Fellows	University of Newcastle, Australia
Marc Kilgour	Wilfried Laurier University, Waterloo, Canada
Jérôme Lang	Université Paris-Dauphine, France
Jean-François Laslier	Ecole Polytechnique, Paris, France
Noam Nisan	Hebrew University, Jerusalem, Israel
Ariel D. Procaccia	Harvard University, Cambridge, MA, USA
Frances Rosamond	University of Newcastle, Australia
Jeffrey S. Rosenschein	Hebrew University, Jerusalem, Israel
Francesca Rossi	University of Padova, Italy
Jörg Rothe (co-chair)	HHU Düsseldorf, Germany
Remzi Sanver	Bilgi University Istanbul, Turkey
Arkadii Slinko	University of Auckland, New Zealand
Kristen Brent Venable	University of Padova, Italy
Toby Walsh	NICTA and University of New South Wales, Sydney, Australia
Michael Wooldridge	University of Liverpool, UK
William S. Zwicker	Union College, Schenectady, NY, USA

Additional Reviewers

Haris Aziz	Chih Chang	Jessica Davies
Felix Fischer	Serge Gaspers	Christophe Gonzales
Umberto Grandi	Mingyu Guo	Paul Harrenstein
Ian Kash	George Katsirelos	Scott Kominers
Sven Krumke	Dmitriy Kvasov	Jean Laine
Andrew Lin	Dipjyoti Majumdar	Reshef Meir
Nina Narodytska	Mahdi Parsa	Evangelia Pyrga
Hans Georg Seedig	Alexander Skopalik	Michael Zuckerman

Sponsors

DFG: Deutsche Forschungsgemeinschaft
ESF: European Science Foundation

Workshop Website

http://ccc.cs.uni-duesseldorf.de/COMSOC-2010/

Preface

Computational social choice, an interdisciplinary field of study at the interface of social choice theory and computer science, promotes a bidirectional exchange of ideas between both fields. On the one hand, techniques developed in computer science (such as complexity analysis or algorithm design) are applied to social choice mechanisms (such as voting procedures or fair division protocols) and problems related to them. On the other hand, concepts from social choice theory are imported into computing. For instance, social welfare orderings originally developed to analyze the quality of resource allocations in human society are equally well applicable to problems in multiagent systems or network design.

Social choice theory is concerned with the design and analysis of methods for collective decision making. Much classical work in the field has focused on establishing abstract results on the existence of procedures meeting certain requirements, but such work has not usually taken computational issues into account. For instance, while it may not be possible to design a voting protocol that makes it impossible for a voter to cheat in one way or another, it may well be the case that cheating successfully turns out to be a computationally intractable problem, which may therefore be deemed an acceptable risk. Examples of topics studied in *computational social choice* include the complexity-theoretic analysis of voting protocols (with respect to both developing computationally feasible mechanisms and exploiting computational intractability as a means against strategic manipulation), and the application of techniques developed in artificial intelligence and logic to the compact representation of preferences in combinatorial domains.

These and other COMSOC topics are well represented in these proceedings of the Third International Workshop on Computational Social Choice (COMSOC-2010), hosted by the Institut für Informatik at Heinrich-Heine-Universität Düsseldorf on September 13–16, 2010. As with the previous two workshops in this biennial series (COMSOC-2006 in Amsterdam and COMSOC-2008 in Liverpool), our aim in organizing COMSOC-2010 has been to bring together different communities: computer scientists interested in computational issues in social choice; people working in artificial intelligence and multiagent systems who are using ideas from social choice to organize societies of artificial software agents; logicians interested in the logic-based specification and analysis of social procedures (social software); and last but not least people coming from social choice theory itself. Moreover, COMSOC-2010 will be held in association and co-located with the COST Action "Algorithmic Decision Theory."

We received 57 submissions,[1] which again represents an increase over the previous COMSOC workshop. Each submission was reviewed by at least two members of the program committee, supported by many additional reviewers. Eventually, 39 papers were accepted to be presented at the workshop and to be included—in revised form—in these proceedings. As with the previous two COMSOC workshops, the Call for Papers explicitly solicited submissions of both original papers and of papers describing recently published work, so some of the papers have recently appeared in other publication venues as well or may be submitted elsewhere soon. The copyright of the articles in this volume lies with the individual authors.

In addition, the proceedings contain short abstracts of talks to be given by our invited speakers: Gabrielle Demange (Paris School of Economics), Matthew O. Jackson (Stanford University), Bettina Klaus (University of Lausanne), Hervé Moulin (Rice University), and Hannu Nurmi (University of Turku). A wide range of COMSOC topics is covered by both the

[1] This is not counting the paper "When Alternatives Vote over Voters" submitted by Marky D. Kondor VII of the University of International Waters, which proposed an "ambitious novel research agenda of *inverted social choice*" (where the alternatives rank the voters). We were about to label the paper as rejected when Marky informed us that due to tight competition from other workshops his paper had to reject COMSOC (but agreed with a preprint in our proceedings), and he encourages us to continue reviewing his papers in the future. Good luck, Marky, with submitting your paper to COSMOC-2011! The COSMOC workshop series—as proposed in Marky's paper—will take place biennially in *odd* years.

invited talks and the contributed papers, spanning complexity issues in winner determination for voting systems and tournament solutions as well as strategic manipulations (both in the term-of-art sense of manipulation and in the related senses of bribery, control, and cloning in elections); multiagent resource allocation, fairness, judgement aggregation, and cake-cutting algorithms; approximating voting rules; determining possible winners in elections and studying single-peaked electorates; coalition formation and cooperative game theory; mechanism design in social choice and mechanism design with payments; and matching problems in social choice as well as pure social choice and political science topics.

COMSOC-2010 continues a "tradition" established at COMSOC-2008: The actual workshop is preceded by a day of tutorials, which will help newcomers to the field to get acquainted with computational social choice in an easily accessible manner. Vincent Conitzer (Duke University) will give a general introductory tutorial and more specific invited tutorials will be presented by Agnieszka Rusinowska (Université Paris 1 Panthéon-Sorbonne), Nicolaus Tideman (Virginia Tech), and Toby Walsh (NICTA and University of NSW). Our tutorial day is called the "LogICCC Tutorial Day"—LogICCC is a EUROCORES program of the European Science Foundation (ESF) that supports several collaborative research projects, including "Computational Foundations of Social Choice" and "Social Software for Elections, the Allocation of Tenders and Coalition/Alliance Formation," which both are closely related to COMSOC. In addition, there will be a special LogICCC session, and the abstracts of the LogICCC tutorials and short talks are also contained in the proceedings.

First and foremost, we thank Ulle Endriss and Jérôme Lang for starting and coordinating the COMSOC workshop series and for their help and advice in organizing COMSOC-2010. We thank the authors for their excellent papers, the workshop participants for attending (at the time of this writing, more than 80 have already registered), and the PC members for their support, advice, and hard work during the preparation for COMSOC-2010. Both our PC members and the additional reviewers wrote high-quality reviews, and they did so under a lot of time pressure. We also thank the many people who have been engaged in the local organization of COMSOC-2010, in particular the Düsseldorf Organizing Team—especially Dorothea Baumeister and Claudia Forstinger for their huge amount of work, Gábor Erdélyi, Claudia Lindner, Magnus Roos, Lena Piras, Anja Rey, Alina Elterman, Florian Klein, Nhan-Tam Nguyen, and Hilmar Schadrack for their organizational help; Isabelle Mehlhorn, Bernd Prümm, and Irene Rothe for the cover design; Heinz Mehlhorn from Düsseldorf University Press for his help and advice; and Eva Hoogland from ESF for her helpful advice and support. Finally, we are grateful to the sponsors of COMSOC-2010 for their generous financial support: the Deutsche Forschungsgemeinschaft and the European Science Foundation.

The topics covered in these proceedings are examples of a wider trend towards interdisciplinary research involving all of decision theory, game theory, social choice, and welfare economics on the one hand, and computer science, artificial intelligence, multiagent systems, operations research, and computational logic on the other. In particular, the mutually beneficial impact of research in microeconomic theory and computer science is already widely recognized and has lead to significant advances in areas such as auction theory (including applications to combinatorial auctions and sponsored search auctions), solving games/equilibrium computation (including applications to the allocation of security assets as well as AI for games such as poker), analysis of strategic behavior in networks, electronic commerce, and negotiation in multiagent systems. What had been missing until 2006 was a forum that specifically addresses computational issues in social choice theory. When the COMSOC workshop series was launched four years ago, the hope was to be able to fill this gap. This hope has been fulfilled by the success of the COMSOC workshop series so far. We are looking forward to an exciting workshop in Düsseldorf.

Durham & Düsseldorf, July 2010 V.C. & J.R.

Table of Contents

LogICCC Tutorials and Talks

A Brief Introductory Tutorial on Computational Social Choice

Vincent Conitzer

Abstract

This is a brief description of the introductory tutorial given at COMSOC 2010.

1 Focus of the Tutorial

This tutorial gives a brief introduction to computational social choice. It is directed especially at the workshop participants who are new to this community, to give them a foothold from which to appreciate the rest of the workshop. Because the workshop program is densely packed, there is too little time to give an exhaustive overview of all the exciting current research topics in computational social choice. Hence, this tutorial focuses strictly on computational aspects of common voting rules. There are two main reasons for this. First, a large fraction of the current research in computational social choice concerns such topics. Second, it gives good insight into the *type* of problem in which the computational social choice community is interested.

2 Topics

In this tutorial, after a quick review of voting rules, we consider some representative problems from computational social choice. For each voting rule, we are confronted with the following computational problems:

1. How hard is it to *execute* the voting rule, that is, to determine the winning alternative(s)?
2. How hard is it to *manipulate* the voting rule by misreporting one's preferences?
3. How hard are other types of undesirable behavior? For example, how hard is it for the chair of the election to *control* the outcome of the election, for instance by introducing additional candidates? How hard is it for an outside party to effectively *bribe* voters?
4. If we have partial information about the votes, how hard is it to determine whether a particular alternative is still a *possible winner*?
5. How can the voters effectively *communicate* their preference information to determine the winning alternative?

It should be noted that for topics 2 and 3 above, computational hardness is *desirable*, because it may prevent the undesirable behavior. This raises interesting questions about whether the worst-case nature of computational complexity theory is appropriate here.

3 Materials and Further Reading

The slides will be made available (at least) on the presenter's website, where the slides of a longer tutorial on the same topic, given jointly with Ariel Procaccia, can also be found.

There are several overview articles of research in this area (e.g., [1, 4, 3, 2, 5]), which also provide references to more focused technical papers.

References

[1] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007)*, 2007.

[2] Vincent Conitzer. Comparing multiagent systems research in combinatorial auctions and voting. *Annals of Mathematics and Artificial Intelligence*. To appear.

[3] Vincent Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.

[4] Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, chapter 14, pages 375–406. Springer, 2009.

[5] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? Working paper.

Vincent Conitzer
Department of Computer Science
Levine Science Research Center, Box 90129
Duke University
Durham, NC 27708, USA
Email: `conitzer@cs.duke.edu`

Different Approaches to Influence in Social Networks

Agnieszka Rusinowska

1 Extended Abstract

The *influence* phenomenon is faced in all kinds of real life situations, and as a consequence it is studied in many scientific areas: in sociology and social psychology, in political science, in economics, in management and business science. Also different approaches are applied to study influence concepts: research is not restricted only to theoretical investigations, but more and more experiments are conducted to get a deeper insight into these phenomena. In the economics literature, studying different concepts related to influence can find its place in several branches of this field, like, e.g., in labor economics, political and public economics, game theory, contract theory, experimental economics, and industrial organization. One of the game theoretic approaches to influence is based on using *social networks* which are particularly suitable to such an analysis. The **aim of this talk** is to deliver a short overview of different approaches to influence applied in the economics and game-theoretic literature, with a particular focus on *studying influence in networks*.

Concerning the game-theoretic literature, both cooperative and noncooperative approaches to influence have been applied; for a short survey, see e.g. [8]. Already more than fifty years ago the concept of influence relation to qualitatively compare the a priori influence of voters in a simple game was introduced [13], and fifty years later this influence relation was extended to voting games with abstention [15]. The cooperative game theoretical approach to interaction is also used in [11, 12], where the authors apply the command structure to model players' interaction relations by simple games.

A very important game theoretic approach to influence is based on using social networks, since they play a central role in the sharing of information and the formation of opinions. Individual decisions and strategic interaction are both embedded in social networks which are therefore particularly useful in analyzing influence. In the decision process the mutual influence does not stop necessarily after one step but may iterate. In this survey, we particularly discuss the iterated models of influence. The seminar network interaction model of information transmission, opinion formation, and consensus formation is presented in [4]; see also e.g. [5, 14]. In [10] the authors consider a social network in which players make an acceptance/rejection decision on a certain proposal, and each of them has an inclination (preliminary opinion) to say either "yes" or "no". It is assumed that players may influence the decisions of others, and consequently the players' decisions may differ from their preliminary inclinations. For further research on this model, see e.g. [6, 7, 9].

Another interesting approach to influence in social networks is based on using relational algebra and RelView [1, 2] which is a BDD-based tool for the visualization and manipulation of relations and for prototyping and relational programming. In [3] the authors apply relation algebra to measure agents' 'strength' (like power, success, and influence) in a social network. This leads to specifications, which can be executed with the help of the BDD-based tool RelView after a simple translation into the tool's programming language. Determining such measures can become quite complex and requires a lot of computations. Hence, using a computer program to compute the measures is extremely useful for real life applications of the concepts in question.

References

[1] R. Behnke, R. Berghammer, E. Meyer, and P. Schneider. RelView — A system for calculation with relations and relational programming. In E. Astesiano, editor, *Fundamental Approaches to Software Engineering*, LNCS 1382, Springer-Verlag, pages 318–321, 1998.

[2] R. Berghammer, G. Schmidt, and M. Winter. RelView and Rath – Two systems for dealing with relations. In H. de Swart, E. Orlowska, G. Schmidt, and M. Roubens, editors, *Theory and Applications of Relational Structures as Knowledge Instruments*, LNCS 2929, Springer, pages 1–16, 2003.

[3] R. Berghammer, A. Rusinowska, and H. de Swart. Applying relational algebra and RelView to measures in a social network. *European Journal of Operational Research*, 202:182–195, 2010.

[4] M.H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69:118–121, 1974.

[5] P. DeMarzo, D. Vayanos, and J. Zwiebel. Persuasion bias, social influence, and unidimensional opinions. *Quarterly Journal of Economics*, 118:909–968, 2003.

[6] M. Grabisch and A. Rusinowska. A model of influence in a social network. *Theory and Decision*, forthcoming, 2009a.

[7] M. Grabisch and A. Rusinowska. Measuring influence in command games. *Social Choice and Welfare*, 33:177–209, 2009b.

[8] M. Grabisch and A. Rusinowska. Different approaches to influence based on social networks and simple games. In A. van Deemen and A. Rusinowska, editors, *Collective Decision Making: Views From Social Choice and Game Theory*. Series Theory and Decision Library C, pages 185–209. Springer, 2010a.

[9] M. Grabisch and A. Rusinowska. Influence functions, followers and command games. *Games and Economic Behavior*, forthcoming, 2010b.

[10] C. Hoede and R. Bakker. A theory of decisional power. *Journal of Mathematical Sociology*, 8:309–322, 1982.

[11] X. Hu and L.S. Shapley. On authority distributions in organizations: equilibrium. *Games and Economic Behavior*, 45:132–152, 2003a.

[12] X. Hu and L.S. Shapley. On authority distributions in organizations: controls. *Games and Economic Behavior*, 45:153–170, 2003b.

[13] J.R. Isbell. A class of simple games. *Duke Mathematical Journal*, 25:423–439, 1958.

[14] M.O. Jackson. Social and Economic Networks. Princeton University Press, 2008.

[15] B. Tchantcho, L. Diffo Lambo, R. Pongou, B. Mbama Engoulou. Voters' power in voting games with abstention: Influence relation and ordinal equivalence of power theories. *Games and Economic Behavior*, 64:335–350, 2008.

Agnieszka Rusinowska
Centre d'Economie de la Sorbonne
CNRS - Université Paris I Panthéon-Sorbonne
106-112 Bd de l'Hôpital
75647 Paris, France
Email: `agnieszka.rusinowska@univ-paris1.fr`

Some Research Problems in Computational Social Choice

Nicolaus Tideman

It is useful to think of a social choice process as composed of a sequence of sub-processes: nomination, evaluation, message submission, message processing, resolution of ties (if any), and authoritative announcement of the result. Researchers with computational expertise might want to apply their efforts to any of the sub-processes, or to questions of design that involve combinations of the sub-processes. But the questions that stand out as calling for the talents of persons with computational expertise are primarily questions of evaluating the feasibility and attractiveness of message processing rules (vote-counting rules).

1 Questions Associated with the Spatial Model of the Election Universe

Many questions with respect to the evaluation of vote-counting rules require a model of the process that generates election outcomes. Recent evidence suggests that a spatial model is appropriate for this purpose. Consider elections in which voters rank the candidates. (That is, the message that the participants in the social choice process must send is a ranking of the options.) For an election with M candidates, define an "election outcome" as a vector of with $M!$ components, in which each component is the number of voters who placed the candidates in one of the $M!$ possible orders. Three candidates span a space of two dimensions. In this space, assume that voters have ideal points that have a bivariate normal distribution, and that they have circular indifference contours. The space is then divided into six wedges assigned to the six orderings of the candidates. There are five degrees of freedom in the shares of votes going to the different orderings of the candidates, but only four degrees of freedom in the spatial model, so the spatial model is refutable. Evidence indicates that deviations from the spatial model can entirely or nearly entirely be explained by sampling variability. Research questions: Will the results hold for additional data sets? The proportion of voters who know the candidates appears to correlate with how well the spatial model explains the outcome. Can other correlates be identified? What happens when you look at elections with four candidates? With five? With M? Are other versions of the spatial model better? What is the best way to deal with ties that arise in survey data? Implication: Modeling of the consequences of alternative voting rules should be done with the spatial model.

2 Questions Associated with Identifying the Outcome under Rules for Selecting One Candidate from More than Two

A number of voting rules have been proposed for elections with more than two candidates. Some of these rules pose computational problems. Examples: The Condorcet-Kemeny-Young rule potentially requires the evaluation of $M!$ sums. The Ranked Pairs rule (which I devised) poses computational challenges that I could imagine solving only in a very crude and time-consuming way. Are there computationally efficient ways of dealing with the difficult cases that could occasionally arise under these voting rules? What about the "estimated centrality" rule, which selects the candidate whose estimated spatial location is closest to the center of the distribution of voters' ideal points. Is that rule computationally feasible for more than three candidates? Would someone like to offer a general program that counted votes by a wide variety of rules?

3 Questions Associated with Evaluating the Susceptibility of Voting Rules to Strategizing

The Gibbard-Satterthwaite theorem tells us that all reasonable voting rules are subject to strategy in some instances. There are a number of ways in which the susceptibility of voting rules to strategizing might be measured. What is the best way to measure the susceptibility of voting rules to strategizing? How do different rules compare?

4 Questions Associated with the Single Transferable Vote Form of Proportional Representation

The Single Transferable Vote (STV) is a form of proportional representation in which voters submit rankings of candidates, and votes are counted by a complex algorithm that is intended to identify a winning set of candidates of a specified size that reflects the diversity of preferences in the electorate. There are a number of versions of STV, varying in their sophistication and in their susceptibility to different concerns. There are at least two proposed versions of STV that may be so sophisticated that they might require an unacceptably long time to determine the winners. Thus it is interesting to ask: What are the best computational algorithms for identifying the winning sets of candidates under the highly sophisticated versions of STV? What are the resulting computational times with specified hardware? If the most sophisticated versions of STV pose computational problems that make it impossible to guarantee computability, what are the closest approximations that do permit guarantees of computability?

Nicolaus Tideman
Department of Economics
Virginia Tech
Blacksburg VA 24060, USA
Email: ntideman@vt.edu

Where Are the Hard Manipulation Problems?

Toby Walsh

Abstract

One possible escape from the Gibbard-Satterthwaite theorem is computational complexity. For example, it is NP-hard to compute if the STV rule can be manipulated. However, there is increasing concern that such results may not reflect the difficulty of manipulation in practice. In this tutorial, I survey recent results in this area.

The Gibbard Satterthwaite theorem proves that, under some simple assumptions, a voting rule can always be manipulated. A number of possible escapes have been suggested. For example, if we relax the assumption of an universal domain and replace it with single peaked preferences, then strategy free voting rules exist. In an influential paper [1], Bartholdi, Tovey and Trick proposed that complexity might offer another escape: perhaps it is computationally so difficult to find a successful manipulation that agents have little option but to report their true preferences? Many voting rules have subsequently been shown to be NP-hard to manipulate [3]. However, NP-hardness only dictates the worst-case and may not reflect the difficulty of manipulation in practice. Indeed, a number of recent theoretical results suggest that manipulation can often be easy (e.g. [19]).

I argue here that we can study the hardness of manipulation empirically [17, 18]. There are several reasons why empirical analysis is useful. For example, theoretical analysis is usually restricted to simple distributions like uniform votes. Votes in real elections may be very different due, for instance, to correlations between votes. As a second example, theoretical analysis is often asymptotic so does not reveal the size of hidden constants. Such constants may be important to the actual computational cost. In addition, elections are typically bounded in size so asymptotic results may be uninformative. Such experiments suggest different behaviour occurs in the problem of computing manipulations of voting rules than in other NP-hard problems like propositional satisfiability [2, 13], constraint satisfaction [4, 9], number partitioning [6, 8], and other NP-hard problems [7, 14, 15]. For instance, many transitions seen in our experiments appear smooth, as seen in polynomial problems [16].

Another problem in which manipulation may be an issue is the stable marriage problem. Can agents be married to a more preferred partner by mis-reporting their preferences? Unfortunately, Roth [11] proved that *all* stable marriage procedures can be manipulated. We might hope that computational complexity might also be a barrier to manipulate stable marriage procedures. In joint work with Pini, Rossi and Venable, I have proposed a new stable marriage procedures based on voting that is NP-hard to manipulate [10]. This procedure has other desirable properties like gender neutrality.

A third domain in which manipulation may be an issue is sporting tournaments [12]. Manipulating a sporting tournament is slightly different to manipulating an election. In a sporting tournament, the voters are also the candidates. Since it is hard (without bribery or similar mechanisms) for a team to play better than it can, we consider just manipulations where the manipulators can throw games. We show, for example, that we can decide how to manipulate round robin and cup competitions, two of the most popular sporting competitions in polynomial time.

References

[1] J.J. Bartholdi, C.A. Tovey, and M.A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[2] P. Cheeseman, B. Kanefsky, and W.M. Taylor. Where the really hard problems are. In *Proc. of the 12th IJCAI*, pages 331–337, 1991.

[3] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate. *Journal of the Association for Computing Machinery*, 54, 2007.

[4] I.P Gent, E. MacIntyre, P. Prosser, B.M. Smith, and T. Walsh. Random constraint satisfaction: Flaws and structure. *Constraints*, 6(4):345–372, 2001.

[5] I.P. Gent, H. Hoos, P. Prosser, and T. Walsh. Morphing: Combining structure and randomness. In *Proc. of the 16th National Conf. on AI.* AAAI, 1999.

[6] I.P. Gent and T. Walsh. Phase transitions and annealed theories: Number partitioning as a case study. In *Proc. of 12th ECAI*, 1996.

[7] I.P. Gent and T. Walsh. The TSP phase transition. *Artificial Intelligence*, 88:349–358, 1996.

[8] I.P. Gent and T. Walsh. Analysis of heuristics for number partitioning. *Computational Intelligence*, 14(3):430–451, 1998.

[9] E. MacIntyre, P. Prosser, B.M. Smith, and T. Walsh. Random constraint satisfaction: Theory meets practice. In *4th Int. Conf. on Principles and Practices of Constraint Programming (CP-98)*, pages 325–339. Springer, 1998.

[10] M.S. Pini, F. Rossi, K.B. Venable, and Walsh T. Manipulation and gender neutrality in stable marriage procedures. In *8th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 665–672, 2009.

[11] A. E. Roth. The economics of matching: Stability and incentives. *Mathematics of Operations Research*, 7:617–628, 1982.

[12] T. Russell and T. Walsh. Manipulating tournaments in cup and round robin competitions. In *Algorithmic Decision Theory, First Int. Conf., ADT 2009*, pages 26–37. Springer, 2009.

[13] T. Walsh. The constrainedness knife-edge. In *Proc. of the 15th National Conf. on AI.* AAAI, 1998.

[14] T. Walsh. Search in a small world. In *Proc. of 16th IJCAI.* 1999.

[15] T. Walsh. Search on high degree graphs. In *Proc. of 17th IJCAI.* 2001.

[16] T. Walsh. From P to NP: COL, XOR, NAE, 1-in-k, and Horn SAT. In *Proc. of the 17th National Conf. on AI.* AAAI, 2002.

[17] T. Walsh. Where are the really hard manipulation problems? The phase transition in manipulating the veto rule. In *Proc. of 21st IJCAI*, pages 324–329. 2009.

[18] T. Walsh. An empirical study of the manipulability of single transferable voting. In *Proc. of the 19th ECAI.* IOS Press, 2010.

[19] L. Xia and V. Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *EC '08: Proc. of the 9th ACM conference on Electronic commerce*, pages 99–108, 2008.

Toby Walsh
NICTA and UNSW
Sydney, Australia
Email: toby.walsh@nicta.com.au

Problem Solving on Simple Games via BDDs

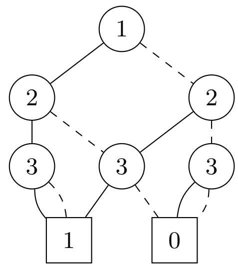
Rudolf Berghammer and Stefan Bolus

Abstract

Simple games are yes/no cooperative games which arise in many practical applications, especially in political life and the formation of alliances and coalitions. Binary decision diagrams (BDDs) can be used to represent, for instance, Boolean function, sets of subsets and relations. They are extensively studied and were applied to various research problems. In this extended abstract we'll give a motivation why it is a good idea to consider BDDs as another representation for simple games.

1 Motivation

A simple game (see e.g. [4]) is a pair $(N, \mathcal{W})$ where N is a set of so called *players* and $\mathcal{W} \subseteq 2^N$ is an up-set (with respect to set inclusion) of so called *winning coalitions*. Elements not in $\mathcal{W}$ are called *losing* and elements in 2^N are called *coalitions*. Binary decision diagrams (see e.g. [2]) are directed, labeled and acyclic graphs with a root and two designated sinks (1-/0-sink) such that each non-sink has two outgoing edges. As one can see from Fig. 1, they can be used to represent Boolean function in a very natural way. Each path corresponds to an assignment and the sink determines the outcome. Because simple games are technically a set of subsets they can easily be represented by their characteristic function $\chi : \{0,1\}^{|N|} \to \{0,1\}$ where the first player corresponds to the first Boolean variable and so one.



variable 1	2	3	f-value
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Figure 1: A BDD for a Boolean function. Numbers inside circles (labels) correspond to Boolean variables. The rectangular nodes are the 1- and 0-sink, respectively. Edges are directed downwards. Solid/dashed edges are 1-/0-edges.

So called *quasi-reduced and ordered binary decision diagrams* (QOBDDs) are BDDs that share sub-BDDs whenever possible. E.g., in Fig. 1 the center node with label 3 is shared. QOBDDs are often small in practice. In general, however, they can grow exponentially in the number of Boolean variables. The same holds for monotone Boolean functions and even threshold functions[1] where in the latter case the bounds for the number of nodes are $\mathcal{O}(2^{n/2})$ and $\mathcal{O}(|N|Q)$ if Q is the threshold (see [3, 1]), but even the latter bound is rarely reached in practice. A similar bound can be shown for multiple weighted voting games (MWVG; see again [1]). Additionally, different classes of QOBDDs (WVG, MWVG, any) can exhibit useful properties which perhaps can be exploited to derive efficient algorithms. For instance, building the QOBDD for the minimal winning coalitions of a WVG from the QOBDD of its winning coalition is a linear time algorithm in the number of QOBDD nodes.

[1]Threshold functions correspond exactly to characteristic functions of weighted voting games (WVGs).

The use of QOBDDs offers not only another representation of simple games, WVG and MWVG, but due to its relatively compact representation of simple games it also allows to solve problems for real world instances. A feature which is offered by other explicit representation just to a very limited degree. Moreover, QOBDDs can be manipulated like sets as long as they represent an up-set. For instance, constraints for the winning coalitions can be applied. Winning coalitions of multiple games can be combined not only using conjunction but also using other operations like disjunction to model multiple opportunities for a coalition to win. For instance, the US Federal Legal System and Taylor's and Zwicker's Magic Squares can be modeled using that.

Despite the very famous problem of computing different power indices for simple games, the computation of the desirability relation on the players and the test for dummy players are two basic problems which appear in some other more complex problems like the test to be a WVG or not . Here, one can profit from the fact that BDDs were already applied to many problems from different areas and many problems have been solved in a slightly different notion. For instance, dummy players in simple games correspond exactly to redundant variables in Boolean functions. Other problems can be solved using existing operations on QOBDDs and some simple algorithms like the following one to compute the QOBDD for the blocking coalitions (and thus the dual game) from the winning coalition of a simple game:

Compls(v) $\equiv$
  if v is a sink **then return** v
  elsif v was already visited with result r **then return** r
  else r := $\mathsf{ite}(i, \mathit{Compls}(\mathsf{else}(v)), \mathit{Compls}(\mathsf{then}(v)))$
    mark v as visited with result r and **return** r

Graphically, the algorithm just exchanges each node's 1- and 0-edge. Thus, it has a running time linear in the number of nodes. This allows to handle even larger real world problems like the International Monetary Fund with 186 players which has about 16 mil. nodes.

Our research in this direction has two main objectives. The first one is to study the complexity of known problems using the BDD representation. This is especially interesting since QOBDDs can have exponential size in general but have a bounded size for special classes like WVGs. The second objective is to develop and provide applicable methods which can be used not only by computer scientists and maybe serve as a foundation for new questions.

References

[1] S. Bolus. Power indices of simple games and vector-weighted majority games by means of binary decision diagrams. 2010. Submitted.

[2] R.E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3):293–318, 1992.

[3] K. Hosaka, Y. Takenaga, and S. Yajima. On the size of ordered binary decision diagrams representing threshold functions. *Algorithms and Computation*, pages 584–592, 1994.

[4] A.D. Taylor and W.S. Zwicker. *Simple Games: Desirability Relations, Trading, Pseudoweightings.* Princeton University Press, 1999.

Rudolf Berghammer and Stefan Bolus
Institut für Informatik, Universität Kiel
Olshausenstraße 40
24098 Kiel, Germany
Email: `{rub|stb}@informatik.uni-kiel.de`

Consensus Measures Generated by Weighted Kemeny Distances on Linear Orders

José Luis García-Lapresta and David Pérez-Román

Extended Abstract

In the field of Social Choice, Bosch [4] introduced the notion of *consensus measure* as a mapping that assigns a number between 0 and 1 to every profile of linear orders, satisfying three properties: *unanimity* (in every subgroup of agents, the highest degree of consensus is only reached whenever all individuals have the same ranking), *anonymity* (the degree of consensus is not affected by any permutation of agents) and *neutrality* (the degree of consensus is not affected by any permutation of alternatives).

In García-Lapresta and Pérez-Román [8] we extended Bosch's notion of consensus measure to the context of weak orders (indifference among different alternatives is allowed) and we consider some additional properties that such measures could fulfill: *maximum dissension* (in each subset of two agents, the minimum consensus is only reached whenever preferences of agents are linear orders and each one is the inverse of the other), and *reciprocity* (if all individual weak orders are reversed, then the consensus does not change). After that, a class of consensus measures based on the distances among individual weak orders were introduced and analyzed. See also García-Lapresta and Pérez-Román [7].

In this contribution, we consider the above mentioned framework and properties for the case of linear orders. However, we now deal with the possibility of weighting discrepancies among linear orders by taking into account where these discrepancies appear. Since in some decision problems it is not the same to have differences in the top alternatives than in the bottom ones (see Baldiga and Green [3]), we introduce weights for distinguishing where these differences occur. To do this, we consider a class of consensus measures generated by weighted Kemeny distances, and we analyze some of their properties. The Kemeny metric was initially defined on linear orders by Kemeny [9], as the number of pairs where the orders' preferences disagree. We note that the Kemeny distance is a metric, but the introduced weighted Kemeny distances are not metrics in the sense of Deza and Deza [5]. On the the use of Kemeny and other metrics in the field of Social Choice see Eckert and Klamler [6].

Recently, Alcalde-Unzu and Vorsatz [1, 2] have introduced some consensus measures in the context of linear orders –related to some rank correlation indices– and they provide some axiomatic characterizations. It is important to note that both papers introduce a preliminary analysis to the weighting approach of consensus measures in the context of linear orders. See also Baldiga and Green [3].

It is interesting to note that the introduced consensus measures generated by weighted Kemeny distances can be used for designing appropriate decision making processes that require a minimum agreement among agents. For instance, in García-Lapresta and Pérez-Román [7] we propose a voting system where agents' opinions are weighted by the marginal contributions to consensus.

With respect to the computational aspect, we are preparing a computer program to obtain the consensus in real decisions when agents rank order the feasible alternatives. We are also working in an extension of the weighted consensus measures to the framework of weak orders.

Acknowledgements

The financial support of the Junta de Castilla y León (Consejería de Educación, Projects VA092A08 and GR99), the Spanish Ministerio de Ciencia e Innovación (Project ECO2009–07332) and ERDF are gratefully acknowledged.

References

[1] J. Alcalde-Unzu, M. Vorsatz. Do we agree? Measuring the cohesiveness of preferences. *University of the Basque Country Working Paper 2010/28*, 2010

[2] J. Alcalde-Unzu, M. Vorsatz. Measuring Consensus: Concepts, Comparisons, and Properties. In: *Consensual Processes* (eds. E. Herrera-Viedma, J.L García-Lapresta, J. Kacprzyk, H. Nurmi, M. Fedrizzi, S. Zadrozny). Springer, forthcoming.

[3] K.A. Baldiga, J.R. Green. Choice-based measures of conflict in preferences. Harvard University, 2010.

[4] R. Bosch. *Characterizations of Voging Rules and Consensus Measures*. Ph. D. Dissertation, Tilburg University, 2005.

[5] M.M. Deza, E. Deza. *Enciclopedia of Distances*. Springer-Verlag, 2009.

[6] D. Eckert, C. Klamler. Distance-Based Aggregation Theory. In: *Consensual Processes* (eds. E. Herrera-Viedma, J.L García-Lapresta, J. Kacprzyk, H. Nurmi, M. Fedrizzi, S. Zadrozny). Springer, forthcoming.

[7] J.L. García-Lapresta, D. Pérez-Román. Some consensus measures and their applications in group decision making. In: *Computational Intelligence in Decision and Control* (eds. D. Ruan, J. Montero, J. Lu, L. Martínez, P. D'hondt, E.E. Kerre). World Scientific, Singapore, pp. 611–616, 2008.

[8] J.L. García-Lapresta, D. Pérez-Román. Measuring Consensus in Weak Orders. In: *Consensual Processes* (eds. E. Herrera-Viedma, J.L García-Lapresta, J. Kacprzyk, H. Nurmi, M. Fedrizzi, S. Zadrozny). Springer, forthcoming.

[9] J.G. Kemeny. Mathematics without numbers. *Daedalus* 88, pp. 571–591, 1959.

José Luis García-Lapresta and David Pérez-Román
PRESAD Research Group
University of Valladolid
Avenida Valle Esgueva 6, 47011 Valladolid, Spain
Email: lapresta@eco.uva.es,david@emp.uva.es

Strategy and Manipulation in Medieval Elections

Sara L. Uckelman[1] and Joel Uckelman

There are many goals in developing electoral protocols, including a desire for a system which is transparent, in that it is clear what the rule or procedure to follow is; non-manipulable, in that it is not in a person's best interest to misrepresent their preferences; honest, in the sense that it elects the 'right' candidate; and not open to strategizing, i.e., bribery or collusion. However, these desiderata are in tension with each other: Often, transparent electoral procedures are the least strategy resistant, and many honest procedures encourage manipulation. Thus a balance between these different goals must be sought. In modern times, since the seminal result on vote manipulation, the Gibbard-Satterthwaite Theorem [5, 6], much attention has been devoted to developing voting rules where manipulation is never in the best interest of the voters [4] or which are computationally too complex for the average bounded agent to be able to manipulate [1]. This focus on computational aspects of electoral methods is one of the hallmarks of modern studies on voting.

But pursuit of these goals is not restricted to modern times: Those participating in elections in the Middle Ages also sought transparency, non-manipulability, honesty, and strategyproofness in so far as these properties can be consistently expressed in a single procedure. However, given the lack of computational sophistication in the Middle Ages, alternate approaches were needed in order to promote honesty, discourage strategizing, etc. These approaches can be classified as either external (constraints introduced outside of the electoral procedure, such as incentives for coming to consensus quickly) or internal (constraints introduced within the electoral procedure, such as voting rules which cannot be manipulated without adverse effects, or which are too difficult for the average bounded agent to manipulate). Surveying examples of both approaches in the context of medieval ecclesiastical and secular elections provides an interesting comparison to modern electoral procedures.

Elections in the Middle Ages were used for the same reasons that they are today: To select suitable candidate(s) for a particular office, duty, or obligation. However, it is important to note that the term *electio* was used in the Middle Ages in a broader sense than our modern 'election'. Its primary sense was 'selection' or 'choice', and only secondarily 'election' in the modern sense. Thus, many records which purportedly discuss elections are not discussing elections of the type which interests us. We can identify four categories of medieval electoral processes: (1) Election by an external authority having no direct interest in the election; (2) Indirect election, where electors name other electors who then select or elect the officials; (3) Election by lot; and (4) Election by ballot. Elections of the first and third types are generally computationally uninteresting; the first type corresponds to dictatorial voting rules, and the third type collapses to probability theory. In general, interesting voting methods are found only in the fourth type, election by ballot, though they can also occur in indirect election.

Ecclesiastical elections

In ideal circumstances, the election of popes, bishops, and abbots and abbesses required unanimous consent for a candidate to win. These elections were "conceived as a way to discover God's will. It was guided by the unanimity rule, the only rule that could assure

[1]This author was funded by the NWO project "Dialogical Foundations of Semantics" (DiFoS) in the ESF EuroCoRes programme LogICCC (LogICCC-FP004; DN 231-80-002; CN 2008/08314/GW).

the participants that their decision was right" [3, p. 3]. However, most cases were not ideal: the electorate, being fallible humans, did not have direct access to the will of God, and furthermore, they were often driven by wholly different motivations, such as desire for political influence, knowledge of ecclesiastical favor or reward if their candidate was elected, etc. In such cases, reaching consensus was extremely difficult, if not impossible, resulting in schisms and impasses, and thus alternative methods had to be used.

We consider methods introduced in the election of each of the three types of officials. In papal elections, the use of majority voting was in use from the late 5th C onwards; in later periods, a modified notion of approval voting was also implemented. We highlight three trends in archepiscopal elections: election by *fiat*, election by lots, and dual postulation. The third is the most interesting, as it can be understood as an early example of a "cut and choose" method, one which predates by nearly 500 years the legislative method proposed by James Harrington (1611–77), which is cited by Brams and Taylor as the first example of cut and choose in the political arena [2, p. 12]. The most interesting data comes from the elections of abbots and abbesses, in particular the case study of the abbatial electoral procedure used by the convent of San Zaccaria in Venice at the beginning and the end of the of the 16th C, which is neither anonymous nor consistent.

Secular elections

In secular contexts, votes were used to elect officials to public office (e.g., sheriff, member of parliament, etc.), and to decide upon matters of policy. Quite often, the electoral procedures and voting methods used in these contexts are more sophisticated, and hence more interesting, than in the ecclesiastical contexts, in part because secular elections were not intended to reveal God's will. Secular nevertheless elections faced similar problems of deadlock, and we consider requirements put in place intended to reduce this occurrence. We also look at various methods which were implemented to make the cost of influencing the result of an election prohibitive, including the code of Vicenza for 1264 and the voting systems used in Cambridge from 18 Edward III to 10 Elizabeth I and in Newcastle-upon-Tyne in 1345. These are but a few examples of medieval electoral processes which were safe-guarded against manipulation and strategizing by increasing the actual, monetary cost of such manipulation, rather than the computational cost.

References

[1] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.

[2] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[3] J. M. Colomer and I. McLean. Electing popes: Approval balloting and qualified-majority rule. *Journal of Interdisciplinary History*, 29(1):1–22, 1998.

[4] U. Endriss. Vote manipulation in the presence of multiple sincere ballots. In D. Samet, editor, *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2007)*, pages 125–134. Presses Universitaires de Louvain, 2007.

[5] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[6] M. A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10: 187–217, 1975.

Sara L. Uckelman and Joel Uckelman
Institute for Logic, Language & Computation, Universiteit van Amsterdam
Science Park 904, PO Box 94242, 1090GE Amsterdam, The Netherlands
Email: S.L.Uckelman@uva.nl,J.D.Uckelman@uva.nl

Invited COMSOC-2010 Talks

Collective Attention and Ranking Methods

Extended Abstract

Gabrielle Demange

The use of rankings is becoming pervasive in many areas including academia for ranking researchers, journals, universities, and the Web environment for ranking Internet pages. The public good aspect of information explains the use of rankings. Rankings are based on a costly process of gathering and summarizing some relevant information on the alternatives in a particular topic. When such information is relevant to anyone, the publication of rankings avoids each individual to pay the search and processing costs. For that very reason, rankings have some influence on the attention that is devoted to the various alternatives. In recurrent situations, attention will, in turn, alter the new statements on which subsequent rankings will be based. This paper proposes an analysis of the feedback between rankings, attention intensities, and statements by studying some reasonable dynamics.

A ranking problem is described by a set of items to be ranked and a set of 'experts' who provide some statements on which the ranking will be based. Rankings here are cardinal, meaning that relative scores are assigned to items. In some situations, as in the ranking of Web pages based on the link structure, the items to be ranked coincide with the experts. These situations are sometimes referred to as the judgment by 'peers'.

The analysis bears on ranking methods that satisfy two important properties. The first property, *intensity invariance*, has been introduced for dealing with the situations in which the 'intensity' of statements is not controlled. In such situations, one may not want an expert to increase its impact on the final ranking by an inflation in its statements (there are other justifications, as explained in the paper). An 'intensity invariant' ranking method is obtained by factoring out the intensity of experts' statements. For example, the 'invariant' method, which serves as a basis to PageRank of Google, factors out the intensity of outward links to avoid pages to increase their score by inflating the number of these links.

The second property, that of *supporting weights* views a method as simultaneously assigning scores to the items and weights to the experts. Given the experts' statements, the ranking writes as a weighted combination of the experts' statements in which furthermore the scores and the weights form some sort of an equilibrium relationship. The property is satisfied by most current methods - e.g. the counting method, the invariant method, the Hits method- although it has not be made explicit so far. This property is useful for various reasons. In particular, it helps us to define new methods through alternative equilibrium relationships and to give a precise definition to what a peers' method is.

The first part of the paper considers static problems, in which the experts' statements are given. I introduce a new ranking method that is both intensity invariant and supported by equilibrium weights. The equilibrium is based on the notion of handicaps. There are indeed strong relationships between rankings and handicaps. Since the purpose of handicaps is to adjust the marks received by items so as to equalize their 'strength', rankings and handicaps are inversely related to each other. The method, called the *handicap-based method*, is characterized by simple properties. The computation of the handicap-based ranking relies on a well-known procedure of matrix scaling, called RAS method or iterative proportional fitting procedure.

The second part of the paper studies a recurrent framework to analyze the influence of rankings. This influence is driven by their impact on attention intensities. In a context in which the number of alternatives to consider is huge, experts cannot carefully assess each one and tend to pay more attention to those whose score is higher. For example, while working on a paper, a researcher who uses rankings tends to read more the journals whose ranks are higher. An 'influence function' describes how the current ranking modifies attention intensities. This generates a joint dynamics on rankings and statements because statements depend on both preferences and attention: the current ranking modifies attention intensities, hence the next statements on which next ranking is based. An intuition is that, as past statements have an impact on future statements through rankings computation, we might expect 'the rich to get richer'. However, the impact of such self-enforcing mechanism may differ according to the ranking method. Our aim is to investigate more precisely this link between a ranking method and the dynamics, starting with a simple linear form for the influence function. Contrasted results are obtained for two different classes of methods.

The first class, called the generalized handicap-based methods, is obtained from the handicap-based method by modifying the experts' weights. The class includes both the handicap-based and the counting methods. These methods guarantee stability in the sense that, given preferences for the experts, the sequence of rankings converges towards a unique rest point.

The second class is the class of peers' methods. The rationale behind a peers' method is that the ability of an individual to perform (measured by his score) is correlated with his ability to judge others' performance. In particular, for a method supported by weights, a minimal requirement is that an individual who receives a small score is also assigned a small expert's weight. This defines a peers' method. I show that whatever peers' method, the dynamics may admit multiple limit points for some preferences, each one corresponding to a different support (the support is the subset of items that keep a positive score). Furthermore, the supports of the limit points are independent of the peers' method. Such result illustrates the self-sustaining aspect of a peers' method. Self-sustainability here is not obtained through plain manipulation but through the coordination device induced by the influence of the ranking.[1]

This paper is about the convergence of behaviors and statements. This is also the concern of the large literature that analyzes the influence of opinions channelled by 'neighbors' in a partially connected network. This literature analyzes situations in which individuals receive private signals about a state of the world. One main question is whether (non-strategic) communication will lead opinions to converge to a common belief and, if convergence occurs, how this common belief relates to the initial opinions and the network structure. Instead here information -the ranking- is made public and influences all experts in an identical way. The impact however differs across experts because they differ in their preferences.The analysis shows that the interplay of preferences and the ranking method may induce a variety of different outcomes.

Gabrielle Demange
Paris School of Economics
48 bd Jourdan, 75014 Paris, France
Email: `demange@pse.ens.fr`.

[1]Researchers in computer science have also concerns about the influence of the rankings provided by search engines. The main criticism is that rankings are biased towards already popular webpages, thus preventing the rise in popularity of recently created high quality pages. There has been some proposals to correct the bias, such as introducing some randomness in the rankings, or to account of the date of creation of a page in the computation of the ranking.

Collective Time Preferences

Matthew O. Jackson and Leeat Yariv

We examine collective decisions over streams of consumption. Agents all consume the same stream and evaluate it according to time discounted and smooth utility functions. We show that if agents differ in their time discount factors, then the only way to aggregate their preferences while satisfying unanimity and time-consistency conditions is by appointing a dictator, even when all agents have exactly the same instantaneous utility function. This implies that decision makers embodying several different "personalities" must be time inconsistent. We also show that aggregation via voting results in choices that violate transitivity despite the highly structured space of alternatives.

JEL Classification Numbers: D72, D71, D03, D11, E24

Keywords: Aggregating preferences, time inconsistent preferences, intransitivities, voting

Matthew O. Jackson
Department of Economics
Stanford University
& the Santa Fe Institute
Email: `jacksonm@stanford.edu`

Leeat Yariv
Division of the Humanities and Social Sciences
Caltech
Email: `lyariv@hss.caltech.edu`

Allocation via Deferred-Acceptance under Responsive Priorities

Lars Ehlers and Bettina Klaus

This extended abstract summarizes Ehlers and Klaus (2009)

1 Extended Abstract

We study the allocation of indivisible objects with capacity constraints to a set of agents when each agent receives at most one object and monetary compensations are not possible. Important applications of this model are the assignment of students to public schools, university admissions, and university housing allocation. We assume that students in these situations have strict preferences over the (object) types (e.g., admission to a specific school or university or dormitory rooms of a certain type) and that (object) types might come with a capacity constraint (the maximal number of students a school or university can admit or the maximal number of dormitory rooms of the same type). An allocation rule is a systematic way of solving any allocation problem (with capacity constraints).

In most papers that study the allocation of indivisible objects with capacity constraints, externally prescribed priorities are also specified; this class of problems is usually referred to as "school choice problems" or "student placement problems". Balinski and Sönmez (1999) were the first to formulate the allocation problem based on priorities, which in many real life situation naturally arise, e.g., in school choice students who live closer to a school and/or have siblings attending a school have higher priority at that school. The agents' priorities for a certain type are captured by an ordering of the agents: a priority structure. Given agents' priorities, it is natural to require that the allocation is "stable" with respect to the priorities. This means that there should be no agent who—conditional on higher priority—envies another agent (for receiving a better object). Given a priority structure, Gale and Shapley's (1962) famous deferred acceptance algorithm (an algorithm which has been extensively applied in practice, see Roth, 2008) can be used to find the agent-optimal stable allocation for any problem with capacity constraints and responsive priorities. We call a rule which is based on the agents-proposing deferred-acceptance algorithm with responsive priorities a *responsive DA-rule*.

Note that we do not a priori assume that priorities are externally given. Two other papers that consider this more general model of object allocation with multiple copies of each type and capacity constraints are Ehlers and Klaus (2006) and Kojima and Manea (2009). Kojima and Manea (2009) point out that "Despite the importance of deferred acceptance rules in both theory and practice, no axiomatization has yet been obtained in an object allocation setting with unspecified priorities." Then, they proceed to provide two characterizations of deferred acceptance rules with so-called acceptant substitutable priorities (a larger class of rules than the class of responsive DA-rules which is based on priorities that are determined by a choice function that reflects substitutability in preferences over sets of agents).

We consider situations where resources may change, i.e., it could be that additional objects are available. When the change of the environment is exogenous, it would be unfair if the agents who were not responsible for this change were treated unequally. We apply this idea of solidarity and require that if additional resources become available, then all agents (weakly) gain. This requirement is called *resource-monotonicity*. Next, we add the mild efficiency requirement of *weak non-wastefulness* as well as the very basic and intuitive properties of *individual rationality* and *unavailable type invariance*. We also impose the invariance property *truncation invariance*. Our last property is the well-known strategic robustness condition of *strategy-proofness*. First, we show that these elementary and intuitive properties characterize, for so-called house allocation problems (quotas at most one), the class of responsive DA-rules that are based on the agent-proposing deferred-acceptance algorithm with responsive priority structures (Theorem 1). Second, we extend this characterization to the class of all problems with capacity constraints, by replacing *resource-monotonicity* with the new property of *two-agent consistent conflict resolution* (Theorem 2).

Another situation of interest is the change of the set of agents and objects because agents leave with their allotments. *Consistency* requires that the allocation for the "reduced economy" allocates the remaining objects to the remaining agents in the same way as before. Since many rules do not satisfy *consistency*, we introduce *weak consistency*, which only requires that agents who received the null object in the original economy still receive the null object in any reduced economy. We obtain a third characterization of the class of responsive DA-rules by *unassigned type invariance*, *individual rationality*, *weak non-wastefulness*, *weak consistency*, and *strategy-proofness* (Theorem 3).

References

[1] Balinski, M., and T. Sönmez (1999): A Tale of Two Mechanisms: Student Placement, *Journal of Economic Theory* 84, 73–94.

[2] Ehlers, L., and B. Klaus (2006): Efficient Priority Rules, *Games and Economic Behavior* 55, 372–384.

[3] L. Ehlers and B. Klaus (2009): "Allocation via Deferred Acceptance," CIREC Cahier 17-2009 (`http://ideas.repec.org/p/mtl/montec/17-2009.html`).

[4] Gale, D., and L. Shapley (1962): College Admissions and the Stability of Marriage, *American Mathematical Monthly* 69, 9–15.

[5] Kojima, F., and M. Manea (2009): Axioms for Deferred Acceptance, *Econometrica*, forthcoming.

[6] Roth, A.E. (2008): Deferred Acceptance Algorithm: History, Theory, Practice, and Open Questions, *International Journal of Game Theory* 36, 537–569.

Lars Ehlers
Département de Sciences Économiques and CIREQ
Université de Montréal
Montréal, Québec H3C 3J7, Canada
Email: `lars.ehlers@umontreal.ca`

Bettina Klaus
Faculty of Business and Economics
University of Lausanne
Internef 538, CH-1015, Lausanne, Switzerland
Email: `bettina.klaus@unil.ch`

Impartial Peer Evaluation

Hervé Moulin

Peer evaluation is a central institution of many communities of experts. Evaluating the relative merits of specialized pieces of work requires knowledge that can only be found among those experts, thus it cannot be entrusted to an impartial outside observer. But peer evaluation is plagued by conflicts of interest, a difficulty only partially alleviated by the confidentiality of reports: even protected by the veil of anonymity, evaluator Smith may and will take into account how her message about Jones' work affects Smith's standing within the peer group. Although it is clearly impossible to eliminate entirely the inherent *partiality* of peer evaluation[1], we can nevertheless design group decision rules for specific, limited choice problems, that systematically avoid any conflict of interest.

In a general group decision problem, we call a decision rule *impartial* if an agent's message never has any influence on the aspects of the collective decision that matter to this agent; thus I have no way to use my message strategically, because I am indifferent between all outcomes in my option set.

A family of impartial rules for allocating a divisible commodity is the subject of [1]: a group of four or more partners must divide a bonus (or a malus) among themselves, and each partner has a well formed subjective opinion about the relative contributions of the *other* partners to the bonus, which the rule asks him to report. The key assumption is that he cares only about his own share, not about the distribution among others of the money he does not get. Impartiality means that his report has *no impact* on his final share.

The paper explores impartial rules in two simple problems involving no money, one akin to voting and one to assignment. In the first problem, a group of agents must choose one of them to receive a prize, or undertake a task (not necessarily a desirable one). Each agent cares about receiving the prize or not, but is indifferent about who among the others gets the prize. In the second problem, the agents must be assigned to a given set of indivisible objects (private goods or bads), and each one cares only about which object she gets. A prime example of the second problem is the collective determination of a strict ranking of the agents, based on these agents' messages only, when we assume that each participant only cares about her own rank. Think of a ranking of undergraduate programs by polls of their alumni.

We look for "reasonable" impartial decision rules in these two problems, where "reasonableness" conveys other, more familiar, desirable properties of a rule.

In the first problem, we must assign a purely private commodity called a prize. We look for impartial voting rules: everyone votes for someone other than herself, and whether or not she get the prize is completely independent of her own message (but this message does influence who gets it if not her).

The set of agents is N; agent i's message space is $N\diagdown i$: everyone nominates one of the other agents to be the winner. We interpret $m_i = j$ as supporting the choice of agent j for the winner, which requires the rule to be *monotonic* in the sense that additional votes for a given agent cannot reverse the decision to make her the winner.

With the notation $\mathcal{D} = \sqcap_{i\in N} (N\diagdown i)$, with generic element $x = (x_i)$, a **voting rule** is a mapping $\varphi : \mathcal{D} \to N$, and we want such a rule to satisfy

- **Impartiality**: for all i, x_i, x'_i, x_{-i}, $\varphi(x_i, x_{-i}) = i \Leftrightarrow \varphi(x'_i, x_{-i}) = i$;
- **Unanimity**: for all i, x, $\{x_j = i \text{ for all } j \in N\diagdown i\} \Rightarrow \varphi(x) = i$;

[1] For a formal statement we can invoke the Gibbard-Satterthwaite impossibility result.

- **Monotonicity**: for all $i, j, i \neq j$, all x, $\varphi(x) = i \Rightarrow \varphi(i, x_{-j}) = i$;
- **No Dummy**: for all i, $\varphi(x_i, x_{-i}) \neq \varphi(x'_i, x_{-i})$ for some x_i, x'_i, x_{-i}.

We show that these four requirements are incompatible for $n \leq 4$, but they are compatible for $n \geq 5$ or more agents. The proof is constructive.

In the second problem we must determine a strict ordering of the agents (with respect to some given criteria), when everyone cares only about his own rank (i.e., how many are above but not who). There too it seems possible to design a reasonable mechanism where agent i's report is a strict ranking of agents other than i, and i's actual ranking is independent of his own report.

References

[1] de Clippel, G., Moulin, H. and N. Tideman, 2008, Impartial division of a dollar, *Journal of Economic Theory*, 139, 176-191.

Hervé Moulin
Department of Economics
Rice University
Email: moulin@rice.edu

Nice, but Are They Relevant? A Political Scientist Looks at Social Choice Results

Hannu Nurmi

The motivation for introducing a new voting system or criticizing an old one is often a counterintuitive or unexpected voting outcome. A case in point is Borda's memoir where he criticized the plurality voting and suggested his own method of marks [2]. With time this approach focusing on a specific flaw of a system has given way to studies dealing with a multitude of systems and their properties. An example of such studies (e.g. [3]) is summarized in Table 1.

Here criterion a denotes the Condorcet winner criterion, b the Condorcet loser one, c strong Condorcet criterion, d monotonicity, e Pareto, f consistency, g Chernoff property, h independence of irrelevant alternatives and i invulnerability to the no-show paradox. A "1" ("0", respectively) in the table means that the system represented by the row satisfies (violates) the criterion represented by the column.

A more "graded" approach to comparing two systems with respect to one criterion has also been suggested [1]. The superiority of system A with respect to system B takes on degrees from strongest to weakest as follows:

1. A satisfies the criterion, while B doesn't, i.e. there are profiles where B violates the criterion, but such profiles do not exist for B.
2. in every profile where A violates the criterion, also B does, but not vice versa.
3. in *practically all profiles* where A violates the criterion, also B does, but not vice versa ("A dominates B almost everywhere").
4. in a plausible probability model B violates the criterion with higher probability than A.
5. in those political cultures that we are interested in, B violates the criterion with higher frequency than A.

Comparing systems with respect to just one criterion is, however, not plausible since criteria tend to be contested not only among the practitioners devising voting systems, but also within the scholarly community. Suppose instead that one takes a more holistic view of Table 1 and gives some consideration to all criteria. A binary relation of dominance could then be defined as follows: A system A (strictly) dominates system B in terms of a set of criteria, if and only if whenever B satisfies a criterion, so does A, but not the other way around.

But all criteria are not of equal importance. Nor are they unrelated. Moreover, Table 1 tells very little – in fact nothing – about the likelihood of criterion violations in those cases where those violations are possible. To find out how often a given system violates a criterion – say, elects a Condorcet loser – one has to know how often various preference profiles occur and how these are mapped into voting strategies by voters. Once we know these two things we can apply the system to the voting strategy n-tuples (if the number of voters is n), determine the outcomes, and, finally, compare these with preference profile to find out whether the choices dictated by the criterion contradict those resulting from the profile, e.g. if an eventual Condorcet loser was chosen. Traditionally, two methods have been resorted in estimating the frequency of criterion violations: (i) probability modeling, and (ii)

	Criterion								
Voting system	a	b	c	d	e	f	g	h	i
Amendment	1	1	1	1	0	0	0	0	0
Copeland	1	1	1	1	1	0	0	0	0
Dodgson	1	0	1	0	1	0	0	0	0
Maximin	1	0	1	1	1	0	0	0	0
Kemeny	1	1	1	1	1	0	0	0	0
Plurality	0	0	1	1	1	1	0	0	1
Borda	0	1	0	1	1	1	0	0	1
Approval	0	0	0	1	0	1	1	0	1
Black	1	1	1	1	1	0	0	0	1
Pl. runoff	0	1	1	0	1	0	0	0	0
Nanson	1	1	1	0	1	0	0	0	0
Hare	0	1	1	0	1	0	0	0	0

Table 1: A Comparison of voting procedures

computer simulations. Both are based on generating artificial electorates and calculating how frequently the criterion is violated or some other incompatibility is encountered in these electorates.

A consideration not disclosed by Table 1 is the intuitive difficulty of finding examples demonstrating criterion violations. In some cases such examples are rather straight-forward, while in others one has to work them out. We shall discuss some of these and dwell on their implications for voting system choice.

The mainstream social choice theory is based on the assumption that the individuals are endowed with complete and transitive preference relations over choice alternatives. Since there are circumstances under which non-transitive preferences make perfect sense, it is worthwhile to find out whether plausible alternatives to the ranking assumption exist. Towards the end of the paper we shall briefly outline some of these.

References

[1] E. Lagerspetz. Evaluating voting systems. In M. Wiberg, editor. *Reasoned Choices*, The Finnish Political Science Association, 2004.

[2] I. McLean and A. B. Urken, editors. *Classics of Social Choice*, The University of Michigan Press, 1995.

[3] H. Nurmi. *Voting Procedures under Uncertainty*, Springer Verlag, 2002.

Hannu Nurmi
Public Choice Research Centre
University of Turku
FI-20014 TURKU
Finland
Email: `hnurmi@utu.fi`

Contributed COMSOC-2010 Papers

Partial Kernelization for Rank Aggregation: Theory and Experiments

Nadja Betzler[1], Robert Bredereck[1], and Rolf Niedermeier

Abstract

RANK AGGREGATION is important in many areas ranging from web search over databases to bioinformatics. The underlying decision problem KEMENY SCORE is NP-complete even in case of four input rankings to be aggregated into a "median ranking". We study efficient polynomial-time data reduction rules that allow us to find *optimal* median rankings. On the theoretical side, we improve a result for a "partial problem kernel" from quadratic to linear size. On the practical side, we provide encouraging experimental results with data based on web search and sport competitions, e.g., computing optimal median rankings for real-world instances with more than 100 candidates within milliseconds.

1 Introduction

We investigate the effectiveness of data reduction for computing *optimal* solutions of the NP-hard RANK AGGREGATION problem. Kemeny's corresponding voting scheme goes back to the year 1959 [14] and was later specified by Levenglick [16]. It can be described as follows. An *election* (V, C) consists of a set V of n votes and a set C of m candidates. A vote or a *ranking* is a total order of all candidates. For instance, in case of three candidates a, b, c, the order $c > b > a$ means that candidate c is the best-liked one and candidate a is the least-liked one. For each pair of votes v, w, the *Kendall-Tau distance* between v and w is defined as

$$\text{KT-dist}(v, w) = \sum_{\{c,d\} \subseteq C} d_{v,w}(c, d),$$

where $d_{v,w}(c, d)$ is set to 0 if v and w rank c and d in the same order, and is set to 1, otherwise. The *score* of a ranking l with respect to an election (V, C) is defined as $\sum_{v \in V} \text{KT-dist}(l, v)$. A ranking l with a minimum score is called a *Kemeny ranking* of (V, C) and its score is the *Kemeny score* of (V, C). The central problem considered in this work is as follows:

RANK AGGREGATION: Given an election (V, C), find a Kemeny ranking of (V, C).

Its decision variant KEMENY SCORE asks whether there is a Kemeny ranking of (V, C) with score at most some additionally given positive integer k. The RANK AGGREGATION problem has numerous applications, ranging from building meta-search engines for the web or spam detection [10] over databases [11] to the construction of genetic maps in bioinformatics [12]. Kemeny rankings are also desirable in classical voting scenarios such as the determination of a president (see, for example, `www.votefair.org`) or the selection of the best qualified candidates for job openings. The wide range of applications is due to the fulfillment of many desirable properties from the social choice point of view [23], including the *Condorcet property*: if there is a candidate (*Condorcet winner*) who is better than every other candidate in more than half of the votes, then this candidate is also ranked first in every Kemeny ranking.

Previous work. First computational complexity studies of KEMENY SCORE go back to Bartholdi et al. [3], showing its NP-hardness. Dwork et al. [10] showed that the problem remains NP-hard even in the case of four votes. Moreover, they identified its usefulness in aggregating web search results

[1] Supported by the DFG, research project PAWS, NI 369/10.

and provided several approximation and heuristic algorithms. Recent papers showed constant-factor approximability [2, 22] and an (impractical) PTAS [15]. Schalekamp and van Zuylen [20] provided a thorough experimental study of approximation and heuristic algorithms. Due to the importance of computing optimal solutions, there have been some experimental studies in this direction [8, 9]: An integer linear program and a branch-and-bound approach were applied to random instances generated under a noise model (motivated by the interpretation of Kemeny rankings as maximum likelihood estimators [8]). From a parameterized complexity perspective, the following is known. First fixed-parameter tractability results have been shown with respect to the single parameters number of candidates, Kemeny score, maximum range of candidate positions, and average KT-distance d_a [4]. The *average KT-distance*

$$d_a := \sum_{v,w\in V, v\neq w} \text{KT-dist}(v,w)\big)/(n(n-1))$$

will also play a central role in this work. Moreover, KEMENY SCORE remains NP-hard when the average range of candidate positions is two [4], excluding hope for fixed-parameter tractability with respect to this parameterization. Simjour [21] further introduced the parameter "Kemeny score divided by the number of votes" (also showing fixed-parameter tractability) and improved the running times for the fixed-parameter algorithms corresponding to the parameterizations by average KT-distance and Kemeny score. Recently, Karpinski and Schudy [13] devised subexponential-time fixed-parameter algorithms for the parameters Kemeny score, d_a, and Kemeny score divided by the number of votes. Mahajan et al. [17] studied above guarantee parameterization with respect to the Kemeny score. Introducing the new concept of partial kernelization, it has been shown that with respect to the average KT-distance d_a one can compute in polynomial time an equivalent instance where the number of candidates is at most $162d_a^2 + 9d_a$ [5]. This equivalent instance is called *partial kernel*[2] with respect to the parameter d_a because it only bounds the number of candidates but not the number of votes instead of bounding the total instance size (as one has in classical problem kernels). Finally, it is interesting to note that Conitzer [7] developed a powerful preprocessing technique for solving a similar rank aggregation problem (Slater ranking). His concept of similar candidates is related to our approach.

Our contributions. On the theoretical side, we improve the previous partial kernel from $162d_a^2 + 9d_a$ candidates [5] to $11d_a$ candidates. Herein, the central point is to exploit "stronger majorities", going from "$>_{2/3}$-majorities" as used before [5] to "$\geq_{3/4}$-majorities". In this line, we also prove that the consideration of "$\geq_{3/4}$-majorities" is optimal in the sense that "$\geq_s$-majorities" with $s < 3/4$ do not suffice.

On the practical side, we provide strong empirical evidence for the usefulness of data reduction rules associated with the above mentioned kernelization. An essential property of our data reduction rules is that they can break instances into several subinstances to be handled independently, that is, the relative order between the candidates in two different subinstances in a Kemeny ranking is already determined. This also means that for hard instances which we could not completely solve, we were still able to compute "partial rankings" of the top and bottom ranked candidates. Finally, we employ some of the known fixed-parameter algorithms and integer linear programming to solve sufficiently small parts of the instances remaining after data reduction.

Due to the lack of space, several details are deferred to the full version of the paper.

2 Majority-based data reduction rules

We start with some definitions and sketch some relevant previous results [5]. Then we show how to extend the previous results to obtain a linear partial kernel for the parameter average KT-distance by

[2]A formal definition of partial kernels appears in the upcoming journal version of [5].

value of s	**partial kernel result**	**sp. case: no dirty pairs**
$2/3 < s < 3/4$	quadratic partial kernel w.r.t. n_d ([5, Theorem 5])	polynomial-time solvable ([5, Theorem 4])
$3/4 \leq s \leq 1$	linear partial kernel w.r.t. n_d (Theorem 1)	

Table 1: Partial kernelization and polynomial-time solvability. The term dirty refers to the $\geq_s$-majority for the respective values of s. The number of dirty pairs is n_d. A linear partial kernel w.r.t. the average KT-distance follows directly from the linear partial kernel w.r.t. n_d (Theorem 1).

providing a new reduction rule. We also show the "limits" of our new reduction rule. Finally, we provide two more reduction rules of practical relevance.

Definitions and previous results. The data reduction framework from previous work [5] introduces a "dirtiness concept" and shows that one can delete some "non-dirty candidates" by a data reduction rule leading to a partial kernel with respect to the average KT-distance. The "dirtiness" of a pair of candidates is measured by the amount of agreement of the votes for this pair. To this end, we introduce the following notation. For an election (V, C), two candidates $c, c' \in C$, and a rational number $s \in]0.5, 1]$, we write

$$c \geq_s c'$$

if at least $\lceil s \cdot |V| \rceil$ of the votes prefer c to c'. A candidate pair $\{c, c'\}$ is *dirty according to the $\geq_s$-majority* if neither $c \geq_s c'$ nor $c' \geq_s c$. All remaining pairs are *non-dirty according to the $\geq_s$-majority*. This directly leads to the parameter number n_d of dirty pairs according to the $\geq_s$-majority. Previous work only considered $>_{2/3}$-majorities[3] and provided a reduction rule such that the number of candidates in a reduced instance is at most quadratic in n_d as well as in d_a [5]. In this work, we provide a linear partial kernel with respect to n_d according to the $\geq_s$-majority for $s \geq 3/4$ and show that this leads to a linear partial kernel with respect to d_a.

We say that c and c' are *ordered according to the $\geq_s$-majority* in a preference list l if $c \geq_s c'$ and $c > c'$ in l. If all candidate pairs are non-dirty with respect to the $\geq_s$-majority for an $s > 2/3$, then there exists a *$\geq_s$-majority order*, that is, a preference list in which all candidate pairs are ordered according to the $\geq_s$-majority [5]. Furthermore, such a $>_{2/3}$-majority can be found in polynomial time and is a Kemeny ranking [5]. Candidates appearing only in non-dirty pairs are called *non-dirty candidates* and all remaining candidates are *dirty candidates*. Note that with this definition a non-dirty pair can also be formed by two dirty candidates. See Table 1 for an overview of partial kernelization and polynomial-time solvability results.

We end with some notation needed to state our data reduction rules. For a candidate subset $C' \subseteq C$, a ranking fulfills the condition $C' > C \setminus C'$ if every candidate from C' is preferred to every candidate from $C \setminus C'$. A subinstance of (V, C) *induced* by a candidate subset $C' \subseteq C$ is given by (V', C') where every vote in V' one-to-one corresponds to a vote in V keeping the relative order of the candidates from C'.

2.1 New results exploiting $\geq_{3/4}$-majorities

We improve the partial kernel upper bound [5] for the parameter d_a from quadratic to linear, presenting a new data reduction rule. The crucial idea for the new reduction rule is to consider $\geq_{3/4}$-majorities instead of $>_{2/3}$-majorities. We further show that the new reduction rule is tight in the sense that it does not work for $>_{2/3}$-majorities.

[3] To simplify matters, we write "$>_{2/3}$" instead of "$\geq_s$ with $s > 2/3$", and if the value of s is clear from the context, then we speak of "dirty pairs" and omit "according to the $\geq_s$-majority".

value of s	properties
$1/2 \leq s \leq 2/3$	a $\geq_s$-majority order does not necessarily exist (Example 1)
$2/3 < s < 3/4$	a $\geq_s$-majority order exists (follows from [5, Theorem 4]) **but** a non-dirty candidate and a dirty candidate do not have to be ordered according to the $\geq_s$-majority in a Kemeny ranking (Theorem 2)
$3/4 \leq s \leq 1$	a $\geq_s$-majority order exists (follows from [5, Theorem 4]) **and** in every Kemeny ranking every non-dirty candidate is ordered according to the $\geq_s$-majority with respect to all other candidates (Lemma 1)

Table 2: Properties "induced" by $\geq_s$-majorities for different values of s.

Reduction rule. The following lemma allows us to formulate a data reduction rule that deletes all non-dirty candidates and additionally may break the remaining set of dirty candidates into several subsets to be handled independently from each other.

Lemma 1. *Let $a \in C$ be a non-dirty candidate with respect to the $\geq_{3/4}$-majority and $b \in C \setminus \{a\}$. If $a \geq_{3/4} b$, then in every Kemeny ranking one must have "$a > \cdots > b$"; if $b \geq_{3/4} a$, then in every Kemeny ranking one must have "$b > \cdots > a$".*

As a direct consequence of Lemma 1 we can partition the candidates of an election (V, C) as follows. Let $N := \{n_1, \ldots, n_s\}$ denote the set of non-dirty candidates with respect to the $\geq_{3/4}$-majority such that $n_i \geq_{3/4} n_{i+1}$ for $1 \leq i \leq s-1$. Then,

$$D_0 := \{d \in C \setminus N \mid d \geq_{3/4} n_1\},$$
$$D_i := \{d \in C \setminus N \mid n_i \geq_{3/4} d \text{ and } d \geq_{3/4} n_{i+1}\} \text{ for } 1 \leq i \leq s-1, \text{ and}$$
$$D_s := \{d \in C \setminus N \mid n_s \geq_{3/4} d\}.$$

3/4-Majority Rule. *Let (V, C) be an election and N and $D_0, \ldots, D_s$ be the sets of non-dirty and dirty candidates as specified above. Replace the original instance by the $s+1$ subinstances induced by D_i for $i \in \{0, \ldots, s\}$.*

The soundness of the $3/4$-Majority Rule follows directly from Lemma 1 and it is straightforward to verify its running time $O(nm^2)$. An instance reduced by the 3/4-Majority Rule contains only dirty candidates with respect to the original instance. Making use of a simple relation between the number of dirty candidates and the average KT-distance as also used previously [5], one can state the following.

Theorem 1. *For* KEMENY SCORE *a partial kernel with less than $11 \cdot d_a$ candidates and less than $2n_d$ candidates can be computed in $O(nm^2)$ time.*

Tightness results. We investigate to which $\geq_s$-majorities the results obtained for $\geq_{3/4}$-majorities extend. An overview of properties for a Kemeny ranking for different values of s is provided in Table 2.

For the $>_{2/3}$-majority, instances without dirty candidates are polynomial-time solvable [5]. More precisely, the $>_{2/3}$-majority order is a Kemeny ranking. A simple example shows that for any $s \leq 2/3$ a $\geq_s$-majority order does not always exist:

Example 1. Consider the election consisting of the three candidates $a, b,$ and c and the three votes "$a > b > c$", "$b > c > a$", and "$c > a > b$". Here, $a \geq_{2/3} b$, $b \geq_{2/3} c$, and $c \geq_{2/3} a$. Then, no linear order fulfills all three relations.

The existence of a data reduction rule analogously to the $3/4$-Majority Rule for $\geq_s$-majorities for $s < 3/4$ would be desirable since such a rule might be more effective: There are instances for which a candidate is dirty according to the $\geq_{3/4}$-majority but non-dirty according to a $\geq_s$-majority with $s < 3/4$. Hence, for many instances, the number n_d of dirty pairs according to the $\geq_{3/4}$-majority assumes higher values than it does according to smaller values of s. In the following, we discuss why an analogous s-Majority Rule with $s < 3/4$ cannot exist. The decisive point of the $3/4$-Majority Rule is that, in a Kemeny ranking, every non-dirty candidate must be ordered according to the $\geq_{3/4}$-majority with respect to *every* other candidate. The following theorem shows that this is not true for $\geq_s$-majorities with $s < 3/4$.

Theorem 2. *Consider a $\geq_s$-majority for any rational $s \in]2/3, 3/4[$. For a non-dirty candidate x and a dirty candidate y, $x \geq_s y$ does not imply $x > y$ in a Kemeny ranking.*

Proof. Let s_1 and s_2 be two positive integers such that $s = s_1/s_2$. We construct an election such that there is a non-dirty candidate x with $x \geq_s y$ but "$y > \cdots > x$" in every Kemeny ranking. The set of candidates is $\{x, y, a_1, a_2\}$ and there are the following $n = s_1 \cdot s_2$ votes:

- $s_1 \cdot s_2 - s_1^2$ votes of type 1: $x > y > a_1 > a_2$,
- $2s_1^2 - s_1 \cdot s_2$ votes of type 2: $a_1 > a_2 > x > y$,
- $s_1 \cdot s_2 - s_1^2$ votes of type 3: $y > a_1 > a_2 > x$.

We first show that there is a positive number of votes of every type:

Considering the number of votes of types 1 and 3, recall that $3/4 > s_1/s_2$ and thus $s_2 > 4/3 \cdot s_1$. Hence, it is easy to see that their number is $s_1 \cdot s_2 - s_1^2 > s_1 \cdot (4/3 \cdot s_1 - s_1) > 0$. Regarding votes of type 2, we use the trivial bound that $s_1/s_2 > 1/2$ and thus their number is $2s_1^2 - s_1 \cdot s_2 > s_1 \cdot (2s_1 - 2s_1) = 0$.

Now, we show that x is non-dirty and $x \geq_s y$. The number of votes with $a > x$ for $a \in \{a_1, a_2\}$ is $2s_1^2 - s_1 \cdot s_2 + s_1 \cdot s_2 - s_1^2 = s_1^2 = s \cdot n$ and the number of votes with $x > y$ is $s_1 \cdot s_2 - s_1^2 + 2s_1^2 - s_1 \cdot s_2 = s_1^2 = s \cdot n$ and thus x is non-dirty according to the $\geq_s$-majority and $x \geq_s y$.

In the following, we show that the score of "$y > a_1 > a_2 > x$" is smaller than the score of every other preference list and, hence, there is no Kemeny ranking in which x and y are ordered according to the $\geq_s$-majority.

Since "$a_1 > a_2$" in every vote, "$a_1 > a_2$" in every Kemeny ranking (see e.g. [4]). Distinguishing three cases, we first show that in every Kemeny ranking "$a_1 > x$" if and only if "$a_2 > x$", and "$a_1 > y$" if and only if "$a_2 > y$". After this, we can treat a_1 and a_2 as one candidate of "weight" two and thus with this argument there remain only six preference lists for which the score has to be investigated to show that "$y > a_1 > a_2 > x$" is the only preference list with minimum score.

Case 1: Consider a preference list with "$a_1 > x > a_2$" where y is placed either before or after all other three candidates. This preference list cannot have minimum score since swapping x and a_2 leads to a preference list with smaller score since $a_2 \geq x$ in more than $sn > 2/3 \cdot n$ votes.

Case 2: Consider a preference list with "$a_1 > y > a_2$" where x is placed either before or after all three other candidates. This preference list cannot have minimum score since swapping a_1 and y leads to a preference list with smaller score. This can be seen as follows. Since $s_1 < 3/4 \cdot s_2$, the number of votes with "$y > a_1$" is

$$2s_1s_2 - 2s_1^2 > 2s_1(s_2 - 3/4 \cdot s_2) = 1/2 \cdot s_1s_2 = n/2.$$

Case 3: Consider the preference list "$a_1 > x > y > a_2$". Note that the same preference list with x and y swapped would clearly have a larger score. We show that "$a_1 > a_2 > x > y$" has a smaller score than "$a_1 > x > y > a_2$". The only pairs that change the score are $\{a_2, y\}$ and $\{a_2, x\}$. These pairs contribute with

$$\#_v(a_2 > y) + \#_v(a_2 > x) = 2s_1^2 - s_1s_2 + 2s_1^2 - s_1s_2 + s_1s_2 - s_1^2 = 3s_1^2 - s_1s_2$$

to the old score and with $2n - \#_v(a_2 > y) - \#_v(a_2 > x)$ to the "new" score. Hence, it remains to show that the difference between the old and new score is positive, that is,

$$3s_1^2 - s_1s_2 - 2s_1s_2 + 3s_1^2 - s_1s_2 = 6s_1^2 - 4s_1s_2 > 6 \cdot 2/3 \cdot s_1s_2 - 4s_1s_2 = 0.$$

Finally, we consider the scores of all possible remaining six preference lists $r_1, \dots, r_6$ with a standing for "$a_1 > a_2$":

$r_1: \quad a > x > y$	$r_3: \quad x > a > y$	$r_5: \quad y > a > x$
$r_2: \quad a > y > x$	$r_4: \quad x > y > a$	$r_6: \quad y > x > a$

Let $t(r)$ denote the score of a preference list r. It is easy to verify that $t(r_1) < t(r_2)$, $t(r_1) < t(r_3)$, and $t(r_4) < t(r_6)$. Hence, it remains to compare the score of r_5 with the score of r_1 and r_4. Since a represents two candidates, we count the corresponding pairs twice in the following computations.

$$\begin{aligned}
&t(r_1) - t(r_5) \\
&= 2\#_v(x > a) + 2\#_v(y > a) + \#_v(y > x) - 2\#_v(a > y) - 2\#_v(x > a) - \#_v(x > y) \\
&= 2s_1s_2 - 2s_1^2 + 4s_1s_2 - 4s_1^2 + s_1s_2 - s_1^2 - 4s_1^2 + 2s_1s_2 - 2s_1s_2 + 2s_1^2 - s_2s_1 + s_1s_2 \\
&= 7s_1s_2 - 5s_1^2 > 7s_1 \cdot 4/3 \cdot s_1 - 5s_1^2 = 13/3 \cdot s_1^2 > 0 \\
&t(r_4) - t(r_5) \\
&= \#_v(y > x) + 2\#_v(a > x) + 2\#_v(a > y) - 2\#_v(a > y) - 2\#_v(x > a) - \#_v(x > y) \\
&= s_1s_2 - s_1^2 + 2 \cdot s_1^2 - 2 \cdot (s_1s_2) + 2 \cdot s_1^2 - s_1^2 \\
&= 2s_1^2 - s_1s_2 > 2/3 \cdot s_1^2 > 0
\end{aligned}$$

This shows that r_5 has a smaller score than r_1 and r_4.

Altogether, we showed that r_5 is the only Kemeny ranking. Thus, there is an election with $x \geq_s y$ for every $s \in]2/3, 3/4[$ such that every Kemeny ranking has $y > x$. □

2.2 Exploiting the Condorcet property

We present a well-known data reduction rule of practical relevance and show that it reduces an instance at least as much as the 3/4-Majority Rule. The reduction rule is based on the following easy-to-verify observation.

Observation 1. *Let $C' \subseteq C$ be a candidate subset with $c' \geq_{1/2} c$ for every $c' \in C'$ and every $c \in C \setminus C'$. Then there must be a Kemeny ranking fulfilling $C' > C \setminus C'$.*

To turn Observation 1 into a reduction rule, we need a polynomial-time algorithm to identify appropriate "winning subsets" of candidates. We use the following simple strategy, called *winning subset routine*: For every candidate c, compute a minimal winning subset M_c by iteratively adding every candidate c' with $c' >_{1/2} c''$, $c'' \in M_c$, to M_c. After this, we choose a smallest winning subset.

Condorcet-Set Rule. *If the winning subset routine returns a subset C' with $C' \neq C$, then replace the original instance by the two subinstances induced by C' and $C \setminus C'$.*

It is easy to see that the Condorcet-Set Rule can be carried out in $O(nm^3)$ time. The following proposition shows that the Condorcet-Set Rule is at least as powerful as the $3/4$-Majority Rule, implying that the Condorcet-Set Rule provides a partial kernel with less than $11d_a$ candidates.

Proposition 1. *An instance reduced by the Condorcet-Set Rule cannot be further reduced by the* $3/4$*-Majority Rule.*

Proposition 1 shows that the $3/4$-Majority Rule cannot lead to a "stronger" reduction of an instance than the Condorcet-Set Rule does. However, since the Condorcet-Set Rule has a higher running time, that is $O(nm^3)$ compared to $O(nm^2)$, applying the $3/4$-Majority Rule before the Condorcet-Set Rule may lead to an improved running time in practice. This is also true for the consideration of the following "special case" of the Condorcet-Set Rule also running in $O(nm^2)$ time.

Condorcet Rule. *If there is a candidate* $c \in C$ *with* $c \geq_{1/2} c'$ *for every* $c' \in C \setminus \{c\}$*, then delete* c*.*

Indeed, our experiments will show that combining the Condorcet-Set Rule with the other rules significantly speeds up the practical running times for many instances.

3 Experimental results

To solve sufficiently small remaining parts of the instances left after the application of our data reduction rules, we implemented three exact algorithms. First, an extended version of the search tree algorithm showing fixed-parameter tractability with respect to the Kemeny score [4, 6]. Second, a dynamic programming algorithm running in $O(2^m \cdot nm^2)$ time for m candidates and n votes [4, 19]. Third, the integer linear program [8, Linear Program 3] which was the fastest exact algorithm in previous experimental studies [8, 20]. We use the freely available ILP-solver GLPK[4] to solve the ILP.[5]

Our algorithms are implemented in C++ using several libraries of the boost package. Our implementation consists of about 4000 lines of code. All experiments were carried out on a PC with 3 GHz and 4 GB RAM (CPU: Intel Core2Quad Q9550) running under Ubuntu 9.10 (64 bit) Linux. Source code and test date are available under the GPL Version 3 license under http://theinf1.informatik.uni-jena.de/kconsens/.

We start to describe our results for two different types of web search data (Sections 3.1 and 3.2) followed by instances obtained from sport competitions (Section 3.3).

3.1 Search result rankings

A prominent application of Rank Aggregation is the aggregation of search result rankings obtained from different web search engines. We queried the same 37 search terms as Dwork et al. [10] and Schalekamp and van Zuylen [20] to generate rankings. We used the search engines Google, Lycos, MSN Live Search, and Yahoo! to generate rankings of 1000 candidates. We consider two search results as identical if their URL is identical up to some canonical form (cutting after the top-level domain). Results not appearing in all rankings are ignored. Ignoring the term "zen budism" with only 18 candidates, this results in 36 instances having between 55 and 163 candidates. We start with a systematic investigation of the performance of the individual reduction rules followed by describing our results for the web instances.

We systematically applied all combinations of reduction rules, always sticking to the following rule ordering: If applied, the Condorcet-Set Rule is applied last and the $3/4$-Majority Rule is applied

[4]http://www.gnu.org/software/glpk/

[5]We omit a detailed discussion about the performance of the single algorithms. A systematic comparison of the three algorithms will be provided in the full version of this work.

	blues		gardening		classical guitar	
	time	profile	time	profile	time	profile
001	0.03	$1^2 > 5 > 1 > 101 > 1 > 2$	0.01	$1 > 2 > 1 > 102$	0.03	$1 > 114$
010	0.10	$1^{74} > 9 > 1^{29}$	0.05	$1^{54} > 43 > 1^9$	0.06	$1^6 > 92 > 1^{17}$
011	0.10	$1^{74} > 9 > 1^{29}$	0.05	$1^{54} > 43 > 1^9$	0.07	$1^6 > 92 > 1^{17}$
100	0.84	$1^{74} > 9 > 1^{29}$	0.95	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	1.89	$1^6 > 7 > 1^{50} > 35 > 1^{17}$
101	0.10	$1^{74} > 9 > 1^{29}$	1.03	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	2.03	$1^6 > 7 > 1^{50} > 35 > 1^{17}$
110	0.10	$1^{74} > 9 > 1^{29}$	0.10	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	0.19	$1^6 > 7 > 1^{50} > 35 > 1^{17}$
111	0.10	$1^{74} > 9 > 1^{29}$	0.11	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	0.18	$1^6 > 7 > 1^{50} > 35 > 1^{17}$

Figure 1: The first column encodes the combination of reduction rules used: the first digit is "1" if the Condorcet-Set Rule is applied, the second if the Condorcet Rule is applied and the last digit is "1" if the $3/4$-Majority Rule is applied. For the three instances corresponding to the search terms "blues", "gardening", and "classical guitar" we give the running times in seconds and the profiles describing the result of the data reduction process.

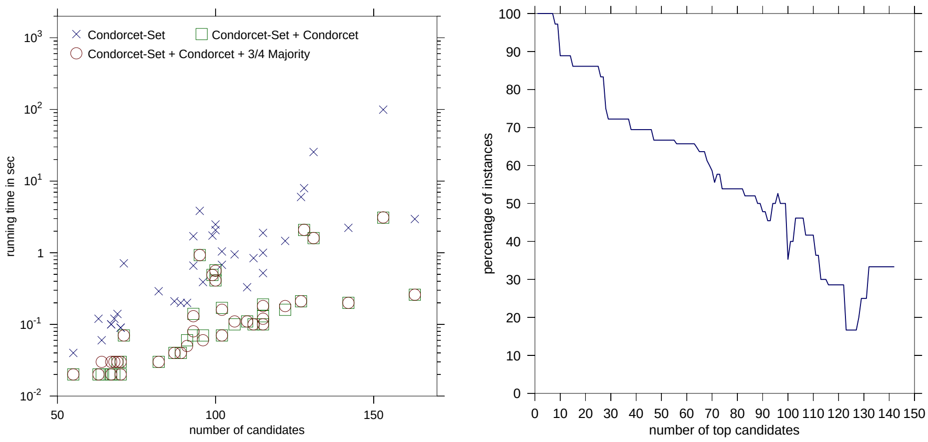


Figure 2: Left: Running times of different combinations of reduction rules. To improve readability, we omitted the data points for the Condorcet-Set Rule combined with the $3/4$-Majority Rule which was usually worse and in no case outperformed the best running times for the other combinations. Right: Percentage of the web search instances for which the x top candidates could be determined by data reduction and dynamic programming within five minutes. For a given number x of top positions, we only considered instances with at least x candidates.

first. After a successful application of the Condorcet-Set Rule, we "jump" back to the other rules (if "activated"). Examples are given in Fig. 1. This led to the following observations.

First, surprisingly, the Condorcet Rule alone led to a stronger reduction than the $3/4$-Majority Rule in most of the instances whereas the $3/4$-Majority Rule never led to a stronger reduction than the Condorcet Rule. Second, for several instances the Condorcet-Set Rule led to a stronger reduction than the other two rules, for example, for gardening and classical guitar (see Fig. 1). It led to a stronger reduction for 14 out of the 36 instances and restricted to the 15 instances with more than 100 candidates (given in Table 3), it led to a stronger reduction for eight of them. Finally, the running times for the Condorcet-Set Rule in combination with the other rules are given in the left part of Fig. 2. Applying the Condorcet Rule before the Condorcet-Set Rule led to a significant speed-up. Additionally applying the $3/4$-Majority Rule changes the running time only marginally. Note that jumping back to the "faster" rules after applying the Condorcet-Set Rule is crucial to obtain the given running times. In the following, by "our reduction rules", we refer to all three rules applied in the order: Condorcet Rule, $3/4$-Majority Rule, and Condorcet-Set Rule.

Table 3: Web data instances with more than 100 candidates. The first column denotes the search term, the second the number of candidates, the third the running time in seconds, and the last column the "profiles" remaining after data reduction to read as follows. Every "1" stands for a position for which a candidate was determined in a Kemeny ranking and higher numbers for groups of candidates whose "internal" order could not be determined by the data reduction rules. Sequences of i ones are abbreviated by 1^i. For example, for the search term "architecture", we know the order of the best 36 candidates, then we know the set of candidates that must assume positions 37– 48 without knowledge of their relative orders, and so on.

search term	# cand.	time	structure of reduced instance		
affirmative action	127	0.21	1^{27}	$> 41 >$	1^{59}
alcoholism	115	0.10	1^{115}		
architecture	122	0.16	1^{36}	$> 12 > 1^{30} > 17 >$	1^{27}
blues	112	0.10	1^{74}	$> 9 >$	1^{29}
cheese	142	0.20	1^{94}	$> 6 >$	1^{42}
classical guitar	115	0.19	1^{6}	$> 7 > 1^{50} > 35 >$	1^{17}
Death+Valley	110	0.11	1^{15}	$> 7 > 1^{30} > 8 >$	1^{50}
field hockey	102	0.17	1^{37}	$> 26 > 1^{20} > 4 >$	1^{15}
gardening	106	0.10	1^{54}	$> 20 > 1 > 1 > 9 > 1^{8} > 4 >$	1^{9}
HIV	115	0.13	1^{62}	$> 5 > 1^{7} > 20 >$	1^{21}
lyme disease	153	3.08	1^{25}	$> 97 >$	1^{31}
mutual funds	128	2.08	1^{9}	$> 45 > 1^{9} > 5 > 1 > 49 >$	1^{10}
rock climbing	102	0.07	1^{102}		
Shakespeare	163	0.26	1^{100}	$> 10 > 1^{25} > 6 >$	1^{22}
telecommuting	131	1.60	1^{9}	$> 109 >$	1^{13}

For all instances with more than 100 candidates, the results of our reduction rules are displayed in Table 3: the data reduction rules are not only able to reduce candidates at the top and the last positions but also partition some instances into several smaller subinstances. Out of the 36 instances, 22 were solved directly by the reduction rules and one of the other algorithms in less than five minutes. Herein, the reduction rules always contributed with less than four seconds to the running time. For all other instances we still could compute the "top" and the "flop" candidates of an optimal ranking. For example, for the search term "telecommuting" there remains a subinstance with 109 candidates but we know the best nine candidates (and their order). The effectiveness in terms of top candidates of our reduction rules combined with the dynamic programming algorithm is illustrated in Fig. 2. For example, we were able to compute the top seven candidates for all instances and the top 40 candidates for 70 percent of the instances.

3.2 Impact rankings

We generated rankings that measure the "impact in the web" of different search terms. For a search engine, a list of search terms is ranked according to the number of the hits of each single term. We used Ask, Google, MSN Live Search, and Yahoo! to generate rankings for all capitals (240 candidates), all nations (242 candidates), and the 103 richest people of the world.[6] Our biggest instance is built from a list of 1349 mathematicians.[7]

As to the capitals, in less than a second, our algorithms (reduction rules and any of the other algorithms for solving subinstances up to 11 candidates) computed the following "profile" of a Kemeny ranking: $1^{45} > 34 > 1^{90} > 43 > 1^{26}$ (see Table 3 for a description of the profile concept). The final Kemeny ranking starts as follows: London $>$ Paris $>$ Madrid $>$ Singapore $>$ Berlin $> \cdots$.

[6] http://en.wikipedia.org/wiki/List_of{capitals_by_countries, richest_people}
[7] http://aleph0.clarku.edu/~djoyce/mathhist/chronology.html

For aggregating the nation rankings, our algorithms were less successful. However, we could still compute the top 6 and the flop 12 candidates. Surprisingly, the best represented nation in the web seems to be Indonesia, followed by France, the United States, Canada, and Australia. The instance consisting of the 103 richest persons could be solved exactly in milliseconds by the data reduction rules. In contrast, for the mathematicians we could only compute the top 31 and flop 31 candidates but could not deal with a subinstance of 1287 candidates between. For the mathematicians instance, the search strategy for minimal subsets for the Condorcet-Set Rule as given in Section 2 led to a running time of more than a day. Hence, we used a cutoff of 20 candidates for the size of the minimal subsets. This decreased the running time to less than one hour.

3.3 Sport competitions

Formula 1. The winner determination of a Formula 1 season can be considered as an election where the candidates are the drivers and the votes are the single races. Currently, the winner determination is based on a "scoring rule", that is, in a single race every candidate gets some points depending on the outcome and the candidate with highest total score wins. We computed Kemeny winners for the seasons from 1970 till 2008. Since currently our implementation cannot handle ties, we only considered candidates that have competed in all races. Candidates that dropped out of a race are ordered according to the order determined by how long the drivers participated in the race. The generated instances have about 16 votes and up to 28 candidates.

Without data reduction, the ILP-approach was the most successful algorithm. It could solve all instances in less than 31 seconds whereas the dynamic programming algorithm could not solve the two instances with the highest number of candidates within 5 minutes. All search tree variants performed even worse. The Condorcet and the Condorcet-Set Rule partitioned nearly all instances in very small components such that a Kemeny ranking could be computed for all years except 1983 in few milliseconds. For 1983 (24 candidates), a remaining component with 19 candidates could be solved in less than one minute by the dynamic programming algorithm.

The Kemeny winner in most of the considered seasons is the same as the candidate selected by the used scoring rule. However, in 2008, Lewis Hamilton was elected as world champion (beating Felipe Massa by only one point) whereas Massa was the "Condorcet driver" and thus the first candidate in every Kemeny ranking. Since in contrast to Kemeny's voting system there is no scoring rule fulfilling the Condorcet property [23], this is no complete surprise.

Winter sport competitions. For ski jumping and cross skiing, we considered the world cup rankings from the seasons 2005/2006 to 2008/2009,[8] ignoring candidates not appearing in all four rankings. Without data reduction, the ski jumping instance, consisting of 33 candidates, was solved by the ILP-solver GLPK in 103 seconds whereas the search tree and dynamic programming algorithms did not find a solution within five minutes. In contrast, the instance was solved in milliseconds by only applying the reduction rules. The cross skiing instance, consisting of 69 candidates, could not be solved without data reduction within five minutes by any of our algorithms but was reduced in 0.04 seconds such that one component with 12 and one component with 15 candidates were left while all other positions could be determined by the reduction rules. The remaining parts could be solved, for example by the dynamic programming algorithm, within 0.12 and 0.011 seconds.

4 Conclusion

Our experiments showed that the described data reduction rules allow for the computation of optimal Kemeny rankings for real-world instances of non-trivial sizes within seconds. For instance, all of our larger now solved instances (with more than 50 candidates) could not be solved by the ILP,

[8] Obtained from `http://www.sportschau.de/sp/wintersport/`

the previously fastest exact algorithm [8], or the two other implemented fixed-parameter algorithms directly. A key-feature of the data reduction rules is to break instances into smaller, independent instances. A crucial observation in the experiments with the different data reduction rules regards certain cascading effects, that is, jumping back to the faster-to-execute rules after a successful application of the Condorcet-Set Rule significantly improves the running time. This shows that the order of applying data reduction rules is important. We could not observe a specific behavior of our data reduction rules for the different types of data under consideration. However, a further extension of the data sets and experiments in this direction are clearly of interest.

On the theoretical side, we improved the previous partial kernel [5] with respect to the parameter average KT-distance from quadratic to linear size. Despite the negative results from Theorem 2, there is still room for improving the $>_{2/3}$-majority based results. In particular, is there a linear partial kernel with respect to the $\geq_s$-majority for any $s < 3/4$? A natural step in answering this question seems to investigate whether for two *non-dirty* candidates a, b, there must be a Kemeny ranking with $a > b$ if $a \geq_s b$. An important extension of RANK AGGREGATION is to consider "constraint rankings", that is, the problem input additionally contains a prespecified order of some candidate pairs in the consensus list [22]. Here, our data reduction rules cannot be applied anymore. New reduction rules for this scenario could also be used in "combination" with the search tree algorithm [4] in an "interleaving mode" [18]. Other challenging variants of RANK AGGREGATION of practical interest are investigated by Ailon [1].

References

[1] N. Ailon. Aggregation of partial rankings, p-ratings, and top-m lists. *Algorithmica*, 57(2):284–300, 2010.

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5), 2008. Article 23 (October 2008).

[3] J. Bartholdi III, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[4] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny rankings. *Theoretical Computer Science*, 410(45):4554–4570, 2009.

[5] N. Betzler, J. Guo, C. Komusiewicz, and R. Niedermeier. Average parameterization and partial kernelization for computing medians. In *Proc. of 9th LATIN*, volume 6034 of *LNCS*, pages 60–71. Springer, 2010. Long version to appear in *Journal of Computer and System Sciences*.

[6] R. Bredereck. *Fixed-Parameter Algorithms for Computing Kemeny scores – Theory and Practice*. Studienarbeit, Universität Jena, 2010. CoRR abs/1001.4003.

[7] V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proc. of 21st AAAI '06*, pages 613–619. AAAI Press, 2006.

[8] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proc. of 21st AAAI*, pages 620–626. AAAI Press, 2006.

[9] A. Davenport and J. Kalagnanam. A computational study of the Kemeny rule for preference aggregation. In *Proc. of 19th AAAI*, pages 697–702. AAAI Press, 2004.

[10] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. of 10th WWW*, pages 613–622, 2001.

[11] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proc. of 22nd ACM SIGMOD*, pages 301–312. ACM, 2003.

[12] B. N. Jackson, P. S. Schnable, and S. Aluru. Consensus genetic maps as median orders from inconsistent sources. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):161–171, 2008.

[13] M. Karpinski and W. Schudy. Approximation schemes for the betweenness problem in tournaments and related ranking problems. Technical report, CoRR abs/0911.2214, 2009.

[14] J. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959.

[15] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proc. of 39th STOC*, pages 95–103. ACM, 2007.

[16] A. Levenglick. Fair and reasonable election systems. *Behavioral Science*, 20(1):34–46, 1975.

[17] M. Mahajan, V. Raman, and S. Sikdar. Parameterizing above or below guaranteed values. *Journal of Computer and System Sciences*, 75:137–153, 2009.

[18] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[19] V. Raman and S. Saurabh. Improved fixed parameter tractable algorithms for two “edge” problems: MAXCUT and MAXDAG. *Information Processing Letters*, 104(2):65–72, 2007.

[20] F. Schalekamp and A. van Zuylen. Rank aggregation: Together we’re strong. In *Proc. of 11th ALENEX*, pages 38–51. SIAM, 2009.

[21] N. Simjour. Improved parameterized algorithms for the Kemeny aggregation problem. In *Proc. of 4th IWPEC*, volume 5917 of *LNCS*, pages 312–323. Springer, 2009.

[22] A. van Zuylen and D. P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34:594–620, 2009.

[23] H. Young and A. Levenglick. A consistent extension of Condorcet’s election principle. *SIAM J. Appl. Math.*, 35(2):285–300, 1978.

Nadja Betzler, Robert Bredereck, and Rolf Niedermeier
Institut für Informatik
Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2
D-07743 Jena, Germany
Email: `{nadja.betzler,robert.bredereck,rolf.niedermeier}@uni-jena.de`

On the Fixed-Parameter Tractability of Composition-Consistent Tournament Solutions

Felix Brandt, Markus Brill, and Hans Georg Seedig

Abstract

Tournament solutions, i.e., functions that associate with each complete and asymmetric relation on a set of alternatives a non-empty subset of the alternatives, play an important role within social choice theory and the mathematical social sciences at large. Laffond et al. have shown that various tournament solutions satisfy *composition-consistency*, a strong structural invariance property based on the similarity of alternatives. We define the *decomposition degree* of a tournament as a parameter that reflects its decomposability and show that computing any composition-consistent tournament solution is fixed-parameter tractable with respect to the decomposition degree. This is of particular relevance for tournament solutions that are known to be computationally intractable such as the Banks set and the tournament equilibrium set, both of which have been proposed in the context of social choice. Finally, we experimentally investigate the decomposition degree of two natural distributions of tournaments.

1 Introduction

Many problems in multiagent decision making can be addressed using tournament solutions, i.e., functions that associate with each complete and asymmetric relation on a set of alternatives a non-empty subset of the alternatives. Tournament solutions are most prevalent in social choice theory, where the binary relation is typically assumed to be given by the simple majority rule (Moulin, 1986; Laslier, 1997). Other application areas include multi-criteria decision analysis (Arrow and Raynaud, 1986; Bouyssou et al., 2006), zero-sum games (Fisher and Ryan, 1995; Laffond et al., 1993; Duggan and Le Breton, 1996), coalition formation (Brandt and Harrenstein, 2011), and argumentation theory (Dung, 1995; Dunne, 2007).

Recent years have witnessed an increasing interest in the computational complexity of tournament solutions by the multiagent systems and theoretical computer science communities. A number of concepts such as the Banks set (Woeginger, 2003), the Slater set (Alon, 2006; Conitzer, 2006), and the tournament equilibrium set (Brandt et al., 2010) have been shown to be computationally intractable. For others, including the minimal covering set and the bipartisan set, algorithms that run in polynomial time but are nevertheless computationally quite demanding because they rely on linear programming, have been provided (Brandt and Fischer, 2008). The class of all tournaments is excessively rich and it is well-known that only a fraction of these tournaments occur in realistic settings (see, e.g., Feld and Grofman, 1992). Therefore, an important question is whether there are natural classes or distributions of tournaments that admit more efficient algorithms for computing specific tournament solutions. In this paper, we study tournaments that are *decomposable* in a certain well-defined way. A set of alternatives forms a component if all alternatives in this set bear the same relationship to all outside alternatives. Elements of a component can thus be seen as variants of the same type of an alternative. Laslier (1997) has shown that every tournament admits a unique natural decomposition into components, which may themselves be decomposable into subcomponents. A tournament solution is *composition-consistent* if it chooses the best alternatives of the best components (Laffond et al., 1996).[1] In other words, a composition-consistent tournament solution can be computed by recursively determining the winning components. All of the tournament solutions mentioned earlier except the Slater set are composition-consistent.

In this paper, we provide a precise formalization of the recursive decomposition of tournaments and a detailed analysis of the speed-up that can be achieved when computing composition-consistent tournament solutions. In particular, we define the *decomposition degree* of a tournament as a parameter that reflects its decomposability. Intuitively, a low decomposition degree indicates that the tournament admits a particularly well-behaved decomposition and therefore allows the efficient computation of composition-consistent tournament solutions. Within our analysis, we leverage a recently proposed linear-time algorithm for the modular decomposition of directed graphs (McConnell and de Montgolfier, 2005; Capelle et al., 2002).

In related work, Betzler et al. (2010) proposed data reduction rules that facilitate the computation of *Kemeny rankings*. One of these rules, the "Condorcet-set rule", corresponds to a (rather limited) special case of composition-consistency where tournaments are decomposed into exactly *two* components. Furthermore, a preprocessing technique that resembles the one proposed in this paper has been used by Conitzer (2006) to speed up the computation of *Slater rankings*. Interestingly, even though Slater's solution is *not* composition-consistent, decompositions of the tournament can be exploited to identify a *subset* of the optimal rankings.

Our results, on the other hand, allow us to compute *complete choice sets* and are applicable to *all composition-consistent tournament solutions*, including the uncovered set (Fishburn, 1977; Miller, 1980), the minimal covering set (Dutta, 1988), the bipartisan set (Laffond et al., 1993), the Banks set (Banks, 1985), the tournament equilibrium set (Schwartz, 1990), and the minimal extending set (Brandt, 2009). The former three admit polynomial-time algorithms whereas the latter three are computationally intractable. None of the concepts is known to admit a linear-time algorithm.

We show that computing any composition-consistent tournament solution is *fixed-parameter tractable* with respect to the decomposition degree of the tournament, i.e., there are algorithms that are only superpolynomial in the decomposition degree. We conclude the paper with an extensive investigation of the decomposition degree of two natural distributions of tournaments. The first one is a well-studied model model that assumes the existence of a true linear ordering of the alternatives that has been perturbed by binary random inversions. The other one is a spatial voting model based on the proximity of voters and alternatives in a multi-dimensional space.

2 Preliminaries

In this section, we provide the terminology and notation required for our results (see Laslier (1997) for an excellent overview of tournament solutions and their properties).

2.1 Tournaments

Let X be a universe of alternatives. For notational convenience we assume that $\mathbb{N} \subseteq X$. The set of all non-empty finite subsets of X will be denoted by $\mathcal{F}(X)$. A (finite) *tournament* T is a pair $(A, \succ)$, where $A \in \mathcal{F}(X)$ and $\succ$ is an asymmetric and complete (and thus irreflexive)

[1] Composition-consistency is related to *cloning-consistency*, which was introduced by Tideman (1987) in the context of social choice.

binary relation on X, usually referred to as the *dominance relation*.[2] Intuitively, $a \succ b$ signifies that alternative a is preferable to b. The dominance relation can be extended to sets of alternatives by writing $A \succ B$ when $a \succ b$ for all $a \in A$ and $b \in B$.[3] We further write $\mathcal{T}(X)$ for the set of all tournaments on X. The *order* $|T|$ of a tournament $T = (A, \succ)$ refers to its number of alternatives $|A|$. Finally, a *tournament isomorphism* of two tournaments $T = (A, \succ)$ and $T' = (A', \succ')$ is a bijective mapping $\pi : A \to A'$ such that $a \succ b$ if and only if $\pi(a) \succ' \pi(b)$.

2.2 Components and Decompositions

An important structural concept in the context of tournaments is that of a *component*. A component is a subset of alternatives that bear the same relationship to all alternatives not in the set.

Definition 1. Let $T = (A, \succ)$ be a tournament. A non-empty subset B of A is a *component* of T if for all $a \in A \setminus B$ either $B \succ a$ or $a \succ B$. A *decomposition* of T is a set of pairwise disjoint components $\{B_1, \dots, B_k\}$ of T such that $A = \bigcup_{i=1}^{k} B_i$.

The *null decomposition* of a tournament $T = (A, \succ)$ is $\{A\}$; the *trivial decomposition* consists of all singletons of A. Any other decomposition is called *proper*. A tournament is said to be *decomposable* if it admits a proper decomposition. Given a particular decomposition, the *summary* of a tournament is defined as the tournament on the individual components rather than the alternatives.

Definition 2. Let $T = (A, \succ)$ be a tournament and $\tilde{B} = \{B_1, \dots, B_k\}$ a decomposition of T. The *summary* of T with respect to $\tilde{B}$ is defined as $\tilde{T} = (\{1, \dots, k\}, \tilde{\succ})$, where

$$i \mathrel{\tilde{\succ}} j \quad \text{if and only if} \quad B_i \succ B_j.$$

A tournament is called *reducible* if it admits a decomposition into *two* components. Otherwise, it is *irreducible*. Laslier (1997) has shown that there exist a natural unique way to decompose any tournament. Call a decomposition $\tilde{B}$ *finer* than another decomposition $\tilde{B}'$ if $\tilde{B} \neq \tilde{B}'$ and for each $B \in \tilde{B}$ there exists $B' \in \tilde{B}'$ such that $B \subseteq B'$. $\tilde{B}'$ is said to be *coarser* than $\tilde{B}$. A decomposition is *minimal* if its only coarser decomposition is the null decomposition.

Proposition 1 (Laslier (1997))**.** *Every irreducible tournament with more than one alternative admits a unique minimal decomposition.*

This is obviously not true for *reducible* tournaments, as witnessed by the tournament $T = (\{1, 2, 3\}, \succ)$ with $1 \succ 2$, $1 \succ 3$, and $2 \succ 3$, which admits two minimal decompositions, namely $\{\{1\}, \{2, 3\}\}$ and $\{\{1, 2\}, \{3\}\}$. Nevertheless, there is a unique way to decompose any reducible tournament. A *scaling decomposition* is a decomposition with a transitive summary.

Proposition 2 (Laslier (1997))**.** *Every reducible tournament admits a unique scaling decomposition such that each component is irreducible.*

This scaling decomposition into irreducible components is also the finest scaling decomposition.

[2] This definition slightly diverges from the common graph-theoretic definition where $\succ$ is defined on A rather than X. However, it facilitates the sound definition of tournament solutions.

[3] To avoid cluttered notation, we omit the curly braces if one of the sets is a singleton, i.e., we write $a \succ B$ instead of the more cumbersome $\{a\} \succ B$.

2.3 Tournament Solutions

A *maximal* element of a tournament $T = (A, \succ)$ is an alternative that is not dominated by any other alternative. Due to the asymmetry of the dominance relation, there can be at most one maximal element, which then also constitutes a *maximum*. Let $\max(T)$ denote the function that yields the empty set or the maximum whenever one exists, i.e.,

$$\max(T) = \{a \in A : a \succ b \text{ for all } b \in A \setminus \{a\}\}.$$

In social choice theory, the maximum of a tournament given by a majority relation is commonly referred to as the *Condorcet winner*.

Since the dominance relation may contain cycles and thus fail to have a maximal element, a variety of concepts have been suggested to take over the role of singling out the "best" alternatives of a tournament. Formally, a *tournament solution* S is defined as a function that associates with each tournament $T = (A, \succ)$ a non-empty subset $S(T)$ of A. Following Laslier (1997), we require a tournament solution to be independent of alternatives outside the tournament, invariant under tournament isomorphisms, and to select the maximum whenever it exists.

Definition 3. A *tournament solution* is a function $S : \mathcal{T}(X) \to \mathcal{F}(X)$ such that

(*i*) $S(T) \subseteq A$ for all tournaments $T = (A, \succ)$;

(*ii*) $S(T) = S(T')$ for all tournaments $T = (A, \succ)$ and $T' = (A, \succ')$ such that $T|_A = T'|_A$;

(*iii*) $S((\pi(A), \succ')) = \pi(S((A, \succ)))$ for all tournaments $(A, \succ)$, $(A', \succ')$, and every tournament isomorphism $\pi : A \to A'$ of $(A, \succ)$ and $(A', \succ')$; and

(*iv*) $S(T) = \max(T)$ whenever $\max(T) \neq \emptyset$.

A tournament solution is composition-consistent if it chooses the "best" alternatives from the "best" components (Laffond et al., 1996).

Definition 4. A tournament solution S is *composition-consistent* if for all tournaments T and $\tilde{T}$ such that $\tilde{T}$ is the summary of T with respect to some decomposition $\{B_1, \dots, B_k\}$,

$$S(T) = \bigcup_{i \in S(\tilde{T})} S(T|_{B_i}).$$

2.4 Fixed-Parameter Tractability and Parameterized Complexity

We briefly introduce the most basic concepts of parameterized complexity theory (see, e.g., Downey and Fellows, 1999; Niedermeier, 2006). In contrast to classical complexity theory, where the size of problem instances is the only measure of importance, parameterized complexity analyzes whether the hardness of a problems only depends on the size of certain parameters. A problem with parameter k is said to be *fixed-parameter tractable* (or to belong to the class FPT) if there exists an algorithm that solves the problem in time $f(k) \cdot poly(|I|)$, where $|I|$ is the size of the input and f is some computable function independent of $|I|$.

For example, each (computable) problem is trivially fixed-parameter tractable with respect to the parameter $|I|$. The crucial point is to identify a parameter that is reasonably small in realistic instances and to devise an algorithm that is only superpolynomial in this parameter.

3 The Decomposition Tree of a Tournament

Propositions 1 and 2 offer a straightforward method to iteratively decompose tournaments. If the tournament is reducible, take the scaling decomposition with irreducible components. If it is irreducible, take the minimal decomposition. The repeated application of these decompositions leads to the *decomposition tree* of a tournament.

Definition 5. The *decomposition tree* $D(T)$ of a tournament $T = (A, \succ)$ is defined as a rooted tree whose nodes are non-empty subsets of A. The root of $D(T)$ is A and for each node $B \in C$ with $|B| \geq 2$, the children of B are defined as follows:

- If $T|_B$ is reducible, the children of B are the components of a finest scaling decomposition of $T|_B$.
- If $T|_B$ is irreducible, the children of B are the components of a minimal decomposition of $T|_B$.

It also follows from Propositions 1 and 2 that every tournament has a *unique* decomposition tree. By definition, each node in $D(T)$ is a component of T and each leaf is a singleton. However, not all components of T need to appear as nodes in $D(T)$. An example of a decomposition tree is provided in Figure 1.

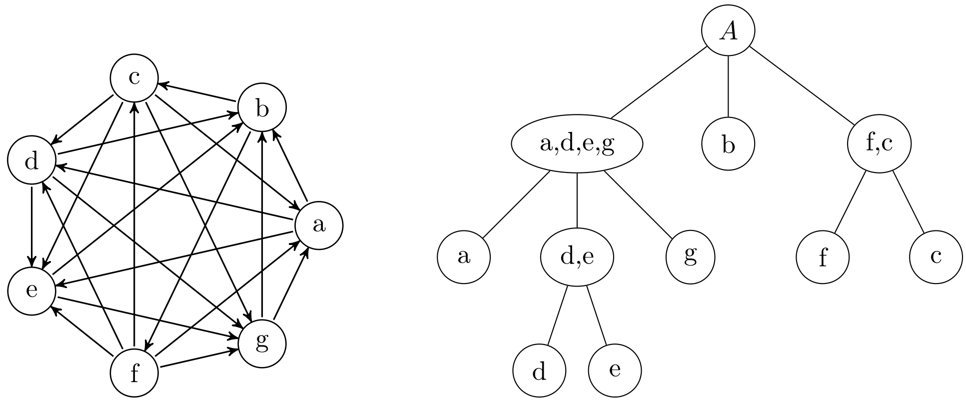


Figure 1: Example tournament with corresponding decomposition tree. Nodes $\{f, c\}$ and {d,e} are reducible, all other nodes are irreducible. Curly braces are omitted to improve readability.

An internal (i.e., non-leaf) node B of $D(T)$ with children $B_1, \dots, B_k$ corresponds to the tournament $T_B = (\{1, \dots, k\}, \tilde{\succ})$ where $i \mathrel{\tilde{\succ}} j$ if and only if $B_i \succ B_j$, i.e., T_B is the summary of $T|_B$ with respect to the decomposition $\{B_1, \dots, B_k\}$. The order of T_B is thus equal to the number of children of node B. Moreover, we call an internal node B *reducible* (respectively, *irreducible*) if the tournament T_B is reducible (respectively, irreducible).[4] If B is reducible, we assume without loss of generality that the children $B_1, \dots, B_k$ are labelled according to their transitive summary, i.e., $B_i \succ B_j$ if and only if $i < j$. In particular, $\max(T_B) = \{1\}$.

Recent results on the modular decomposition of directed graphs (Capelle et al., 2002; McConnell and de Montgolfier, 2005) imply that the decomposition tree of a tournament can be computed in linear time.[5]

Proposition 3. *The decomposition tree of a tournament T can be computed in time $O(|T|^2)$.*

[4] $T|_B$ is reducible (respectively, irreducible) if and only if its summary T_B is.

[5] The representation of a tournament is quadratic in the number of its alternatives.

The proof consists of two steps. In the first step, a *factorizing permutation* of the tournament is constructed. A factorizing permutation of $T = (A, \succ)$ is a permutation of the alternatives in A such that each component of T is a contiguous interval in the permutation. McConnell and de Montgolfier (2005) provide a simple algorithm that computes a factorizing permutation of a tournament in linear time. Furthermore, there exists a fairly complicated linear-time algorithm by Capelle et al. (2002) that, given a tournament T and a factorizing permutation of T, computes the decomposition tree $D(T)$. Since the literature on composition-consistency in social choice and on modular decompositions in graph theory is unfortunately not well-connected and for reasons of completeness, we outline both algorithms in the Appendix.

The concept of a factorizing permutation also yields a simple way to bound the number of nodes in the decomposition tree.

Lemma 1. *The number of* internal *nodes in the decomposition tree of a tournament T is at most $|T| - 1$.*

Proof. Let $\sigma(T)$ be a factorizing permutation of T and consider a node B in $D(T)$. Decomposing B into new components (the children of B in $D(T)$) corresponds to making "cuts" in $\sigma(T)$. Furthermore, each cut generates at most two new components.[6] As there are only $|T| - 1$ possible positions for such a cut, the maximum number of nodes in $D(T)$ is $1 + 2(|T| - 1) = 2|T| - 1$. The bound follows from the observation that $D(T)$ has exactly $|T|$ leaves. □

4 Computing Solutions via the Decomposition Tree

Let S be a composition-consistent tournament solution and consider an arbitrary tournament $T = (A, \succ)$ together with its decomposition tree $D(T)$. Composition-consistency implies that

$$S(T|_B) = \bigcup_{i \in S(T_B)} S(T|_{B_i}) \tag{1}$$

for each internal node B in $D(T)$ with children $B_1, \dots, B_k$. The solution set $S(T)$ can thus be computed by starting at the root of $D(T)$ and iteratively applying equation 1. If B is reducible, we immediately know that $S(T|_B) = S(T|_{B_1})$, since 1 is the maximum in the transitive tournament T_B. A straightforward implementation of this approach is given in Algorithm 1.

Algorithm 1 visits each node of $D(T)$ at most once. The algorithm for computing S is only invoked for tournaments T_B for which B is irreducible. The order of such a tournament T_B is equal to the number of children of the node B in $D(T)$. The *decomposition degree* of T is defined as an upper bound of this number.

Definition 6. The *decomposition degree* $\delta(T)$ of a tournament T is given by

$$\delta(T) = max\{|T_B| : B \text{ is an irreducible internal node in } D(T)\}.$$

Proposition 3 implies that $\delta(T)$ can be computed efficiently. The decomposition degree of the example tournament in Figure 1 is 3.

Let $f(n)$ be an upper bound on the running time of an algorithm that computes $S(T)$ for tournaments of order $|T| \leq n$. Then, the running time of Algorithm 1 can be upper-bounded by $f(\delta(T))$ times the number of irreducible nodes of $D(T)$. We thus obtain the following theorem.

[6] Cuts can be made simultaneously, in which case the number of new components *per cut* is smaller.

Algorithm 1 Compute $S(T)$ via decomposition tree

Compute $D(T)$
$S, S' \leftarrow \emptyset$
$Q \leftarrow (A)$
while $Q \neq ()$ **do**
    $B \leftarrow$ Dequeue(Q)
    if $|B| = 1$ **then**
        $S \leftarrow S \cup B$
    else
        if B is reducible **then**
            Enqueue(Q, B_1)
        else // B is irreducible
            for all $i \in S(T_B)$ **do**
                Enqueue(Q, B_i)
return S

Theorem 1. *Let S be a composition-consistent tournament solution and let $f(k)$ be an upper bound on the running time of an algorithm that computes S for tournaments of order at most k. Then, $S(T)$ can be computed in $O(n^2) + f(\delta) \cdot (n-1)$ time, where δ is the decomposition degree of T and n is the order or T.*

Proof. Let T be a tournament and $n = |T|$. Computing $D(T)$ requires time $O(n^2)$ (Proposition 3). We now show that Algorithm 1 computes $S(T)$ in time $f(\delta(T)) \cdot (n-1)$. Correctness follows from composition-consistency of S. The running time can be bounded as follows. During the execution of the while-loop, each node B of $D(T)$ is visited at most once. If B is reducible or a singleton, there is no further computation. If B is irreducible, $S(T_B)$ is computed. As $|T_B|$ is upper-bounded by $\delta(T)$, this can be done in $f(\delta(T))$ time. Finally, Lemma 1 shows that the number of (internal) nodes of $D(T)$ is at most $n-1$. Summing up, this yields a running time of $O(n^2) + f(\delta(T)) \cdot (n-1)$. ☐

In particular, Theorem 1 shows that the computation of $S(T)$ is fixed-parameter tractable with respect to the parameter $\delta(T)$.

To get a better understanding of this theorem, consider a composition-consistent tournament solution S such that $f(n)$ is in E = DTIME$(2^{O(n)})$. This holds, for example, for the Banks set. For given tournaments T of order n, Theorem 1 then implies that $S(T)$ can be computed efficiently (i.e., in time polynomial in n) whenever $\delta(T)$ is in $O(\log^k n)$. Theorem 1 is also applicable to tractable tournaments solutions such as the minimal covering set and the bipartisan set. Although computing these solutions is known to be in P, existing algorithms rely on linear programming and may be too time-consuming for very large tournaments. For both concepts, a significant speed-up can be expected for distributions of tournaments that admit a small decomposition degree.

Generally, decomposing a tournament asymptotically never harms the running time, as the time required for computing the decomposition tree is only linear in the input size.[7]

5 Experimental Results

It has been shown in the previous section that computing composition-consistent tournament solutions is fixed-parameter tractable with respect to the decomposition degree of a

[7] Checking whether there exists a maximum already requires $O(n^2)$ time.

tournament. While the clustering of alternatives within components has some natural appeal by itself, an important question concerns the value of the decomposition degree for reasonable and practically motivated distributions of tournaments. In this section, we will explore this question experimentally using two probabilistic models from social choice theory. Both models are based on a set of *voters* who entertain preferences over *candidates*. Given a finite set of candidates C and an odd number of voters with linear preferences over C, the *majority tournament* is defined as the tournament $(C, \succ)$, where $a \succ b$ if and only if the number of voters preferring a to b is greater than the number of voters preferring b to a.

Noise model The first model we consider is a standard model in social choice theory where it is usually attributed to Condorcet (see, e.g., Young, 1988). Condorcet assumed that there exists a "true" ranking of the candidates and that the voters possess noisy estimates of this ranking. In particular, he assumed that there is a probability $p > \frac{1}{2}$, such that for each pair a, b of candidates, each voter ranks a and b according to the true ranking with probability p and ranks them incorrectly with probability $1 - p$.

Spatial Model Spatial models of voting are well-studied objects in social choice theory (see, e.g., Austen-Smith and Banks, 2000). For a fixed natural number d of issues, we assume that candidates (i.e., alternatives) as well as voters are located in the space $[0, 1]^d$. The position of candidates and voters can be thought of as their stance on the d issues. Voters' preferences over candidates are given by the proximity to their own position according to the Euclidian distance. We generate tournaments by drawing the positions of candidates and voters uniformly at random from $[0, 1]^d$.

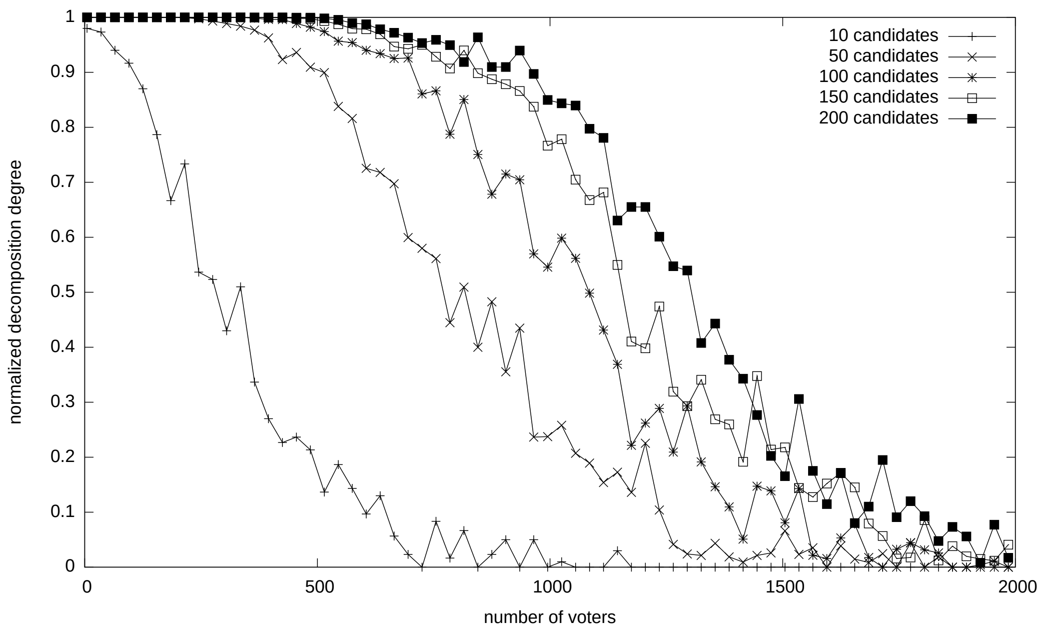


Figure 2: Noise model with $p = 0.55$

The results of our experiments are presented in Figures 2, 3, and 4. The x-axis shows the number of voters, which goes from 5 to 1985 in increments of 30. In order to facilitate the comparison of results for a varying number of candidates, the y-axis is labelled with the *normalized* decomposition degree, i.e., the decomposition degree divided by the number of candidates. Each graph shows the results for a fixed number of candidates, and each data point corresponds to the average value of 30 instances. Whenever the normalized decomposition degree is less than one, composition-consistency can be exploited, even for tournament solutions that already admit fast (say, linear-time) algorithms. The slower the original algorithm, the more dramatic is the speedup obtained by capitalizing on the decomposition tree.

Figure 2 shows the results for the noise model with parameter $p = 0.55$. For any number of candidates, the decomposition degree goes to zero when the number of voters grows. This is not surprising because the probability that the tournament is transitive tends to 1 for any $p > \frac{1}{2}$ (and a transitive tournament T has $\delta(T) = 0$). Interestingly, the decomposition degree drops abruptly when a certain number of voters is reached.

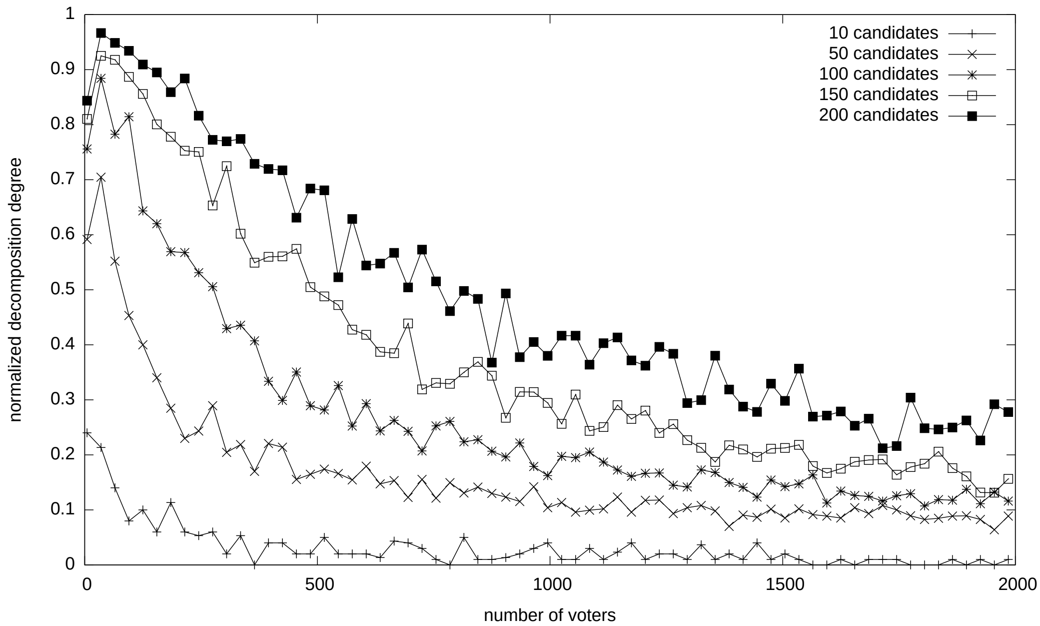


Figure 3: Spatial model with $d = 2$

Figures 3 and 4 show the results for the spatial model for dimensions $d = 2$ and $d = 20$. Surprisingly, the decomposition degree does not significantly increase when moving to a higher-dimensional space. Similar to the noise model discussed above, δ tends to 0 for growing n because a population of voters that is evenly distributed in $[0, 1]^d$ tends to produce transitive tournaments.

The results of our experiments show that, even for moderately-sized electorates, tournaments in both distributions are highly decomposable and therefore allow significantly faster algorithms for computing composition-consistent tournament solutions. For example, consider the two-dimensional spatial model with 150 candidates and some tournament solution that can be computed in time 2^n. For 500 voters, the (average) normalized decomposition degree is approximately 0.5. When assuming for simplicity that the decomposition tree is already given, the speed-up factor (i.e., the running time of the original algorithm divided by the running time of the algorithm that exploits composition-consistency) is $\frac{2^{150}}{2^{75} \cdot (150-1)} \approx 2.5 \cdot 10^{20}$.

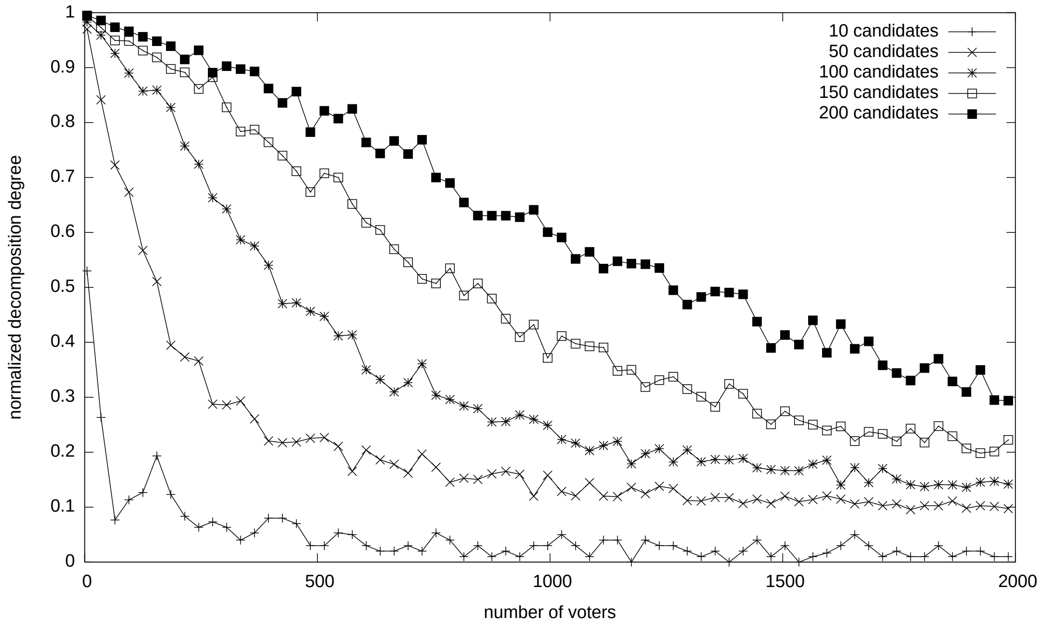


Figure 4: Spatial model with $d = 20$

6 Conclusion

In this paper, we studied the algorithmic benefits of composition-consistent tournament solutions. We defined the decomposition degree of a tournament as a parameter that reflects its decomposability. Intuitively, a low decomposition degree indicates that the tournament admits a particularly well-behaved decomposition. Our main result states that computing any composition-consistent tournament solution is fixed-parameter tractable with respect to the decomposition degree. This is of particular relevance for tournament solutions that are known to be computationally intractable such as the Banks set and the tournament equilibrium set. For example, one corollary of our main result is that the Banks set of a tournament can be computed efficiently whenever the decomposition degree is polylogarithmic in the number of alternatives. We experimentally determined the decomposition degree of two natural distributions of tournaments stemming from social choice theory and found that the decomposition degree in many realistic instances is surprisingly low. As a consequence, the speedup obtained by exploiting composition-consistency when computing tournament solutions for these instances will be quite substantial.

In future work, it would be interesting to measure the concrete effect of capitalizing on composition-consistency on the running time of existing algorithms for specific tournament solutions. Since computing a decomposition tree requires only linear time, it is to be expected that decomposing a tournament never hurts, and often helps. Composition-consistency can be further exploited by parallelization and storing the solutions of small tournaments in a lookup table.

Acknowledgments This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR-2312/6-1 (within the European Science Foundation's EUROCORES program LogICCC), BR 2312/7-1, and BR 2312/3-3.

References

N. Alon. Ranking tournaments. *SIAM Journal on Discrete Mathematics*, 20(1):137–142, 2006.

K. J. Arrow and H. Raynaud. *Social Choice and Multicriterion Decision-Making.* MIT Press, 1986.

D. Austen-Smith and J. S. Banks. *Positive Political Theory I: Collective Preference.* University of Michigan Press, 2000.

J. S. Banks. Sophisticated voting outcomes and agenda control. *Social Choice and Welfare*, 3:295–306, 1985.

N. Betzler, R. Bredereck, and R. Niedermeier. Exact rank aggregation based on effective data reduction. Unpublished manuscript, 2010.

D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukiàs, and P. Vincke. *Evaluation and Decision Models: Stepping Stones for the Analyst.* Springer-Verlag, 2006.

F. Brandt. Minimal stable sets in tournaments. Technical report, http://arxiv.org/abs/0803.2138, 2009. Presented at the 9th International Meeting of the Society of Social Choice and Welfare.

F. Brandt and F. Fischer. Computing the minimal covering set. *Mathematical Social Sciences*, 56(2):254–268, 2008.

F. Brandt and P. Harrenstein. Characterization of dominance relations in finite coalitional games. *Theory and Decision*, 2011. Forthcoming.

F. Brandt, F. Fischer, P. Harrenstein, and M. Mair. A computational analysis of the tournament equilibrium set. *Social Choice and Welfare*, 34(4):597–609, 2010.

C. Capelle, M. Habib, and F. de Montgolfier. Graph decompositions and factorizing permutations. *Discrete Mathematics and Theoretical Computer Science*, 5:55–70, 2002.

V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 613–619. AAAI Press, 2006.

R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

J. Duggan and M. Le Breton. Dutta's minimal covering set and Shapley's saddles. *Journal of Economic Theory*, 70:257–265, 1996.

P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

P. E. Dunne. Computational properties of argumentation systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15):701–729, 2007.

B. Dutta. Covering sets and a new Condorcet choice correspondence. *Journal of Economic Theory*, 44:63–80, 1988.

S. L. Feld and B. Grofman. Who's afraid of the big bad cycle? Evidence from 36 elections. *Journal of Theoretical Politics*, 4:231–237, 1992.

P. C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

D. C. Fisher and J. Ryan. Tournament games and positive tournaments. *Journal of Graph Theory*, 19(2):217–236, 1995.

G. Laffond, J.-F. Laslier, and M. Le Breton. The bipartisan set of a tournament game. *Games and Economic Behavior*, 5:182–201, 1993.

G. Laffond, J. Lainé, and J.-F. Laslier. Composition-consistent tournament solutions and social choice functions. *Social Choice and Welfare*, 13:75–93, 1996.

J.-F. Laslier. *Tournament Solutions and Majority Voting.* Springer-Verlag, 1997.

R. M. McConnell and F. de Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):198–209, 2005.

N. R. Miller. A new solution set for tournaments and majority voting: Further graph-theoretical approaches to the theory of voting. *American Journal of Political Science*, 24 (1):68–96, 1980.

H. Moulin. Choosing from a tournament. *Social Choice and Welfare*, 3:271–291, 1986.

R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.

T. Schwartz. Cyclic tournaments and cooperative majority voting: A solution. *Social Choice and Welfare*, 7:19–29, 1990.

T. N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

G. J. Woeginger. Banks winners in tournaments are difficult to recognize. *Social Choice and Welfare*, 20:523–528, 2003.

H. P. Young. Condorcet's theory of voting. *The American Political Science Review*, 82(4): 1231–1244, 1988.

Felix Brandt, Markus Brill, and Hans Georg Seedig
Institut für Informatik
Technische Universität München
85748 Garching bei München, Germany
`{brandtf,brill,seedigh}@in.tum.de`

Budgeted Social Choice: A Framework for Multiple Recommendations in Consensus Decision Making

Tyler Lu and Craig Boutilier

Abstract

We develop a new framework for social choice problems, *budgeted social choice*, in which a *limited* number of alternatives can be recommended/prescribed to a population of agents. This limit is determined by some form of budget. Such problems naturally arise in a variety of contexts. Our model is general, spanning the continuum from pure consensus decisions (i.e., standard social choice) to fully personalized recommendation. Our results show that standard rank aggregation rules are not appropriate for such tasks and that good solutions typically involve picking diverse alternatives tailored to different agent types. The corresponding optimization problems are shown to be NP-complete, but we develop fast greedy algorithms with some theoretical guarantees. Experimental results on real-world datasets (APA election and sushi) show some interesting patterns and the prove the effectiveness of our greedy algorithms.

1 Introduction

Social choice has received considerable attention in AI and computer science in recent years [10, 13, 7]. This is in part due to technological advances that have facilitated an explosion in the availability of (sometimes implicit) ranking or preference data. Users can, with increasing ease, rate, compare or rank products (e.g., movies, consumer goods, neighborhoods) and information (e.g., clicking on search responses or ads, linking to data sources in social media). This has allowed a great degree of personalization in product recommendation and information provision.

Despite this trend, tailoring the alternatives presented or recommended to specific users can be difficult for any of a number of reasons, among them privacy concerns (actual or perceived), scarce data, or the infeasibility of complete personalization. For example, decisions regarding certain types of public projects (such as highway placement, or park design) may force the choice of a single option: one cannot build different projects to meet the desires of different individuals. Similarly, a company designing a product to meet consumer demand must find a single product that maximizes consumer satisfaction across its target market (assuming sufficient correlation between satisfaction and revenue/profit). In such settings, a single "consensus" recommendation must be made for the population as a whole. If such consensus recommendations are made in a way that is sensitive to the preferences of individuals, we land squarely in the realm of social choice.

There is, of course, a middle ground between pure personalization and pure consensus recommendation. For example, suppose the company can configure its manufacturing facility to produce three variants of the product in question. Then its aim should be to determine three products that *jointly* maximize consumer satisfaction. In the case of public projects, perhaps a small number of projects can be chosen. In domains like web search, if one has insufficient data about an individual making a query (or is reluctant to use it because of privacy concerns), a small number of responses can be presented if browser "real estate" is limited. In the design of pension plan options, there are many reasons to limit the number of offerings available to encourage meaningful choice. In these and numerous other examples, we fall somewhere between making a single consensus recommendation and making fully personalized recommendations for individuals. Some (perhaps implicit) aggregation of users must take place—we cannot offer fully personalized offerings to each individual—placing us in the realm of social choice; but at the same time, we have an opportunity to do some tailoring of the decisions to the preferences of the aggregated groups, and indeed, make choices about the precise form of this aggregation to optimize some social choice function.

In this paper, we develop a general model for just such settings. We call the problem at hand one of *budgeted social choice*. Unlike the usual social choice models, in which a single outcome is selected (or single consensus ranking determined), we allow for the possibility that more than one option can be offered, and assume that each user will *benefit from the best option, according to her own preferences, among those presented*. However, the number of options offered is constrained by a *budget*; this is the key factor that prevents us from exploiting pure personalization to meet the desires of individual users. This budget can take a variety of forms, and we explore several of them in this work. The budget could be a strict limit on the *number* of options (e.g., at most three products can be manufactured, or at most 10 web links can be presented on a page), or on *their cost* (e.g., the total expenditure on city parks cannot exceed $3M). We can also adopt a more nuanced perspective in which the cost of allowing additional options is traded off against the benefit to the target population (e.g., add a fourth product option if increase in consumer satisfaction outweighs the cost of a fourth production line; or extend the city parks budget if increase in social welfare is sufficiently high). Finally, we can consider settings in which the budget is not just a function of the options "created," but also of their overall usage or uptake in the population. Our general framework allows for a fixed charge (e.g., configuring and staffing an assembly line) and per-unit cost (e.g., the marginal cost of producing a unit of product for a specific individual).

Though the motivations are different, multiple-winner models in voting theory [4, 20] can be viewed as an instance of our model. In such systems, the goal is to determine a collection of candidates (e.g., a parliament) that best represents the "collective interests" of the voters (e.g., based on principles of proportional representation). Indeed, our "limited choice" model with Borda scoring corresponds directly to Chamberlin and Courant's [4] proportional representation scheme; in this way, our budgeted choice model can be used to motivate the application of such proportional models to ranking and recommendation, under certain assumptions. Also related is the combinatorial public project problem [19] where given each agent's valuation over all subsets of alternatives, a limited number of alternatives must be chosen for everyone. The focus is more on the tension between approximating social welfare and incentivizing truthfulness (requiring payments from agents).

We begin by outlining a simple model of budgeted social choice in which there is a strict limit K on the number of candidates that can be made available. We do this to illustrate the general principles and intuitions underlying our approach and draw connection to proportional representation schemes. We show that for various social choice objectives, computing the optimal set of K candidates for a set of preferences in this *limited choice model* is NP-hard. However, the induced objective is submodular, and a simple greedy algorithm produces candidate sets whose deviation from optimal is bounded. Computational experiments on various preference data sets show that the greedy algorithm is, in fact, very close to optimal in practice.

We then present our general model in which adding alternatives to the available set is costly (allowing both fixed and per-unit charges) and subject to some form of budget. The limited choice model is a special case of this *costly choice model*. The costly choice model with only fixed charges remains submodular, but when per-unit costs are included, submodularity vanishes. We develop an integer programming formulation of the general optimization problem (which applies directly to the limited choice model). We again provide a greedy heuristic algorithm for solving the general problem which runs in polynomial time. Computational experiments verify its efficacy in practice, but we have no theoretical bounds on its performance currently.

2 Background

We first review some basic concepts from social choice before defining the class of budgeted social choice problems (see [11] for further background). We assume a set of *agents* (or *voters*) $N = \{1, \ldots, n\}$ and a set of *alternatives* (or *candidates*) $A = \{a_1, \ldots, a_m\}$. Let Γ_A be the set of *rankings* (or *votes*) over A (i.e., permutations over A). Alternatives can represent any outcome space over which the voters have preferences (e.g., product configurations, restaurant dishes, candidates for office, public projects, etc.) and for which a single collective choice must be made. Agent ℓ's preferences are represented by a ranking $v_\ell \in \Gamma_A$, where ℓ prefers a_i to a_j, denoted as $a_i \succ_{v_\ell} a_j$, if

$v_\ell(a_i) < v_\ell(a_j)$. We refer to a collection of votes $V = (v_1, \ldots, v_n) \in \Gamma_A^n$ as a *preference profile*.

Given a preference profile, there are two main problems in social choice. The first is selecting a *consensus alternative*, requiring the design of a *social choice function* $f : \Gamma_A^n \to A$ which selects a "winner" given voter rankings/votes. The second is selecting a *consensus ranking* [2], requiring a *rank aggregation function* $f : \Gamma_A^n \to \Gamma_A$. The consensus ranking can be used for many purposes; e.g., the top-ranked alternative can be taken as the consensus winner, or we might select the top k alternatives in the consensus ranking in settings where multiple candidates can be chosen (say, parliamentary seats, or web search results [10]). *Plurality* is the simplest, most common approach for consensus alternatives: the alternative with the greatest number of "first place votes" wins (various tie-breaking schemes can be adopted). However, plurality fails to account for a voter's relative preferences for any alternative other than its top ranked (assuming sincere voting). Other schemes, e.g., *Borda count* or *single transferable vote*, produce winners that are more sensitive to relative preferences. Among schemes that produce consensus rankings, the *Borda ranking* [8] and the *Kemeny consensus* [15] are especially popular.

Definition 1. *Given a ranking v, the* Borda count *of alternative a is $\beta(a, v) = m - v(a)$. The* Borda count *of a relative to preference profile V is $\beta(a, V) = \sum_{v \in V} \beta(a, v)$. A* Borda ranking *$r^*_\beta = r^*_\beta(V)$ is any ranking that orders alternatives from highest to lowest Borda count.*

One can generalize the Borda count by assigning arbitrary scores to the rank positions:

Definition 2. *A* positional scoring function (PSF) *$\alpha : \{1, \ldots, m\} \to \mathbb{R}_{\geq 0}$ maps ranks onto scores s.t. $\alpha(1) \geq \cdots \geq \alpha(m) \geq 0$. Given a ranking v_ℓ and alternative a, let $\alpha_\ell(a) = \alpha(v_\ell(a))$. The* α-score *of a relative to profile V is $\alpha(a, V) = \sum_{v_\ell \in V} \alpha_\ell(a)$. An* α-ranking *$r^*_\alpha = r^*_\alpha(V)$ is any ranking that orders alternatives from highest to lowest α-score.*

Definition 3. *Let $\mathbf{1}$ be the indicator function,* sgn *the sign function and r, v two rankings. The* Kendall-tau *metric is $\tau(r, v) = \sum_{1 \leq i < j \leq m} \mathbf{1}[\mathrm{sgn}[(v(a_i) - v(a_j))(r(a_i) - r(a_j))] < 0]$. Given a profile V, the* Kemeny cost *of a ranking r is $\kappa(r, V) = \sum_{v_\ell \in V} \tau(r, v_\ell)$. The* Kemeny consensus *is any ranking $r^*_\kappa = r^*_\kappa(V)$ that minimizes the Kemeny cost.*

Intuitively, Kendall-tau distance measures the number of pairwise relative misorderings between an output ranking r and a vote v, while the Kemeny consensus minimizes the total number of such misorderings across profile V. While positional scoring is easy to implement, much work in computational social choice has focused on NP-hard schemes like Kemeny [10, 3].

Rank aggregation has interesting connections to work on *rank learning*, much of which concerns aggregating (possibly noisy) preference information from agents into full preference rankings. For example, Cohen et al. [6] focus on learning rankings from (multiple user) pairwise comparison data, while *label ranking* [13] considers constructing personalized rankings from votes. Often unanalyzed is why specific rank aggregations should be chosen for particular settings such as these. One can think of some schemes as a maximum likelihood estimator of some underlying objective ranking (e.g., for Kemeny [22] and positional scoring rules [7]).

3 The Limited Choice Model

While the use of social choice techniques in applications like web search and recommender systems is increasingly common, the motivations for producing consensus recommendations for users with different preferences often varies. Consider, for instance, the motivation for "budgeted" consensus recommendation discussed in our introduction. If a decision maker can provide a limited set of K choices to a population of users to best satisfy their preferences, methods like Kemeny, Borda, etc. *could* be used to produce an aggregate ranking from which the top K alternatives are taken. However, there is little rationale for doing so without a deeper analysis of what it means to "satisfy" the preferences of the user population. In the spirit of our recent work on rank aggregation [17], we develop a precise decision-theoretic formulation of the budgeted social choice problem. Rather than applying existing social choice schemes directly, we derive optimal consensus decisions from decision-theoretic principles and show how these differ (and relate to) classic aggregation rules.

We first introduce the *limited choice problem*, a simple version of budgeted social choice in which one must choose a slate of K alternatives that maximizes some notion of total satisfaction among a group of agents. We develop the more general budgeted model in the next section. Assume a set of n voters with preferences over alternatives A as above. Rather than selecting a single consensus alternative, a decision maker is allowed to recommend K alternatives. Each voter realizes benefit commensurate with its most preferred alternative among the K recommended. For example, a company may be limited to offering K products to its target market, where the products are substitutes (so no consumer will use more than one); or a municipality may have budget for K new parks and citizens draw enjoyment from their most preferred park.

While our goal is to find the best set of K alternatives, the formalization of this model depends on two key choices: how voter satisfaction with a slate is measured; and how we measure social welfare. Our general framework can accommodate many measures of utility and social welfare, but for concreteness we focus on (a) positional scoring (such as Borda) to quantify voter satisfaction; and (b) the sum of such voter "utilities" as our social welfare metric. In other words, our aim is to find a slate of size K that maximizes the sum of the positional scores of each voter's most preferred candidate in the slate:

Definition 4. *Given alternatives A, preference profile V, and PSF α, a K-*recommendation set *is any set of alternatives $\Phi \subseteq A$ of size K. The α-*score *of Φ is:*

$$S_\alpha(\Phi, V) = \sum_{\ell \in N} \max_{a \in \Phi} \alpha_\ell(a) \ . \tag{1}$$

The optimal K-recommendation set *w.r.t. α is:*

$$\Phi^*_\alpha = \operatorname*{argmax}_{|\Phi|=K} S_\alpha(\Phi, V) \ . \tag{2}$$

We use $S_\alpha(\Phi, v)$ to denote the score w.r.t. a single vote/ranking v. We drop the subscript α from S_α when it is evident from the context, and use S_β to denote the special case of Borda scoring.

The objective in Eq. 2 is identical to the Chamberlin and Courant [4] scheme of proportional representation and results for that scheme apply directly to this variant of the limited choice model, as we discuss below. While we focus on total positional scoring as our optimization criterion, the general budgeted framework allows other measures of utility and social desiderata. For example, we can use maximin-fairness (w.r.t. positional scoring) encoded as:

$$\Phi^*_{\text{fair}} = \operatorname*{argmax}_{|\Phi|=K} \min_{\ell \in N} S_\alpha(\Phi, v_\ell) \ . \tag{3}$$

Setting $\alpha(i) = \mathbf{1}[i = 1]$ corresponds to a binary satisfaction measure in which a voter is satisfied with Φ only if its top alternative is made available. In this case, the optimal Φ^*_α corresponds to selecting the K alternatives with the highest "plurality" score (i.e., greatest number of first-place "votes"). However, choosing the top K candidates from a consensus ranking using positional scoring is, in general, not appropriate. For any ranking r, let $r|K$ denote the K top-ranked alternatives in r. The Borda ranking r^*_β can produce slates $r^*_\beta|K$ that are a factor of 2 from optimal using our limited-choice measure, while the α-ranking for arbitrary PSFs can be as much as a factor of K from optimal.

Proposition 5. *For any K we have: (a) $\inf_{(m,n,V)} \frac{S_\beta(r^*_\beta|K,V)}{S_\beta(\Phi^*,V)} = 1/2$; and (b) $\inf_{(\alpha,m,n,V)} \frac{S_\alpha(r^*_\alpha|K,V)}{S_\alpha(\Phi^*,V)} \leq 1/K$.*

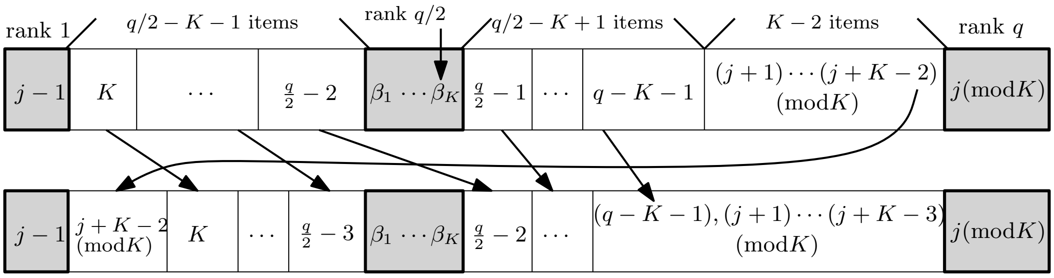


Fig. 1: Example showing that $r^*_\beta|K$ can be factor of 2 worse than optimal. Assume q items $\{0, 1, \ldots, q-K-1, \beta_1, \ldots, \beta_K\}$, and $n = K(q-K-2)$ votes. The votes are divided into K blocks, each containing $q-K-2$ votes. For each block $j \leq K$, item $j-1$ is always the top alternative in each vote, and item $j \pmod K$ is the worst. This means the optimal recommendation set is $\Phi^* = \{0, \ldots, K-1\}$, with $S_\beta(\Phi^*, V) = (q-1)n$. The jth block of votes has a structure illustrated in the figure, with two example votes shown: the items j and $j \pmod K$ are fixed in the top/bottom spots and items $\beta_1, \ldots, \beta_K$ are also fixed in positions $q/2 - K + 1, \ldots, q/2$. (Fixed items are shaded.) The remaining items are arranged in the other positions in the first vote (the non-shaded positions). Starting with one such arrangement (e.g., the top vote in the figure), each candidate is "rotated downward" one non-shaded position (with wrap around) to produce the next vote in the block. This is repeated until $q-K-2$ votes are constructed for block j (i.e., one vote for each non-shaded position). Thus, any non-fixed item occupies each non-shaded rank position in exactly one vote in this block j. Thus, the *average* score of a non-shaded item is $\sum_{i \in [q-2] \setminus \{q/2, \ldots, q/2+K-1\}} i = \frac{-q^2+3\,q-2+qK+K^2-K}{-2q+2K+4} < q/2$ (whenever $q > K+2$, which always holds). Hence the average score of any item in $\{K, \ldots, q-K-1\}$ (which occupy only unshaded positions in all blocks) across all blocks is less than $q/2$. Also observe that the average score of any item in Φ^* is less than $q/2$: item $j-1$ has score $q-1$ in block j but has score 0 in block $j-2 \pmod K$ (giving average $(q-1)/2$ in these two blocks) and has average less than $q/2$ across all other blocks (since it is an unshaded item in those blocks). But the average score of β_i is at least $q/2$ (since its position is fixed in all blocks). Hence the top K items of the Borda ranking r^*_β are $\beta_1, \ldots, \beta_K$. But $S_\beta(r^*_\beta|K, V) = (q/2+K-1)n$, so $S(r^*_\beta|K, V)/S(\Phi^*, V) = (q/2+K-1)/(q-1)$, which approaches $1/2$ from above as $q \to \infty$.

Proof Sketch. (a) To obtain a lower bound, we note that the total Borda score of all alternatives is $\sum_{a \in A} \beta(a, V) = n(0+1+2+\cdots+m-1) = nm(m-1)/2$. The item a^*_β with the highest Borda count must have a count at least the average, over the alternatives, $nm(m-1)/2/m = n(m-1)/2$. Since a^*_β is the highest-ranked element in r^*_β, we have $S_\beta(r^*_\beta|K, V) \geq n(m-1)/2$. By contrast, the score of the optimal set Φ^* is at most $n(m-1)$. Hence $r^*_\beta|K$ has score that is no worse than a factor of $[n(m-1)/2]/[n(m-1)] = 1/2$ from optimal. We demonstrate an upper bound realizing this worst-case error using the example described in Fig. 1.

(b) An upper bound can be demonstrated using an example somewhat similar in spirit to that for the Borda count as in (a); we omit it due to lack of space. It remains open whether $r^*_\alpha|K$ can indeed be worse than a factor of K from optimal. $\square$

These results illustrate that care must be taken in the application of rank aggregation methods to novel social choice problems. In our limited choice setting, the use of positional scoring rules (e.g., Borda) to determine the K most "popular" alternatives can perform extremely poorly. Intuitively, the optimal slate appeals to the *diversity* of the agent preferences in a way that is not captured by "top K" methods. Indeed, this is one of the motivations for the proportional schemes [4, 20]. More importantly, the underlying preference aggregation scheme is defined relative to an explicitly articulated decision criterion. We defer a detailed discussion for lack of space, but we note that STV, often used for proportional representation [21] can perform poorly w.r.t. our criterion as well. Specifically, we can show that the slate produced by STV can be a factor of 2 worse than optimal.

The examples above suggest that determining optimal recommendation sets in the limited choice model may be computationally difficult. This is the case: the problem is NP-complete even for in the specific case of determining voter satisfaction using Borda scoring:[1]

[1]The NP-hardness of a *variant* of the Chamberlin and Courant [4] proportional scheme is shown in [21], but the variant allows for arbitrary misrepresentation scores. The added flexibility in the reduction used means that it does *not* imply the

Theorem 6. *Given preference profile V, integer $K \geq 1$, and $t \geq 0$, deciding whether there exists a K-recommendation set Φ with (Borda) score $S_\beta(\Phi, V) \geq t$ is NP-complete.*

Proof Sketch. Membership in NP is easily verified. For hardness, we reduce an arbitrary *hitting set* instance to our problem: given $E = \{e_1, \ldots, e_p\}$, a set $\{B_1, \ldots, B_q\}$ of subsets of E, and integer $h \geq 1$, is there a $C \subseteq E$ of size at most K such that $\forall i \in \{1, \ldots, q\}, C \cap B \neq \emptyset$? We reduce this to our decision problem, with voters $N = \{1, \ldots, q\}$, alternatives $A = E \cup \{z_{ij} : i \in [q], j \in [\sum_{\ell=1}^{q} |B_\ell|]\}$, $m = |A|$, and $t = qm - \sum_{\ell=1}^{q} |B_\ell|$. Each voter ℓ has a preference ordering with elements in B_ℓ at the top (in arbitrary order), followed by $z_{\ell 1} z_{\ell 2} \cdots z_{\ell t}$, and with remaining alternatives $A \backslash B_\ell$ (in arbitrary order) at the bottom.

Any positive hitting set instance (say, with certificate C) corresponds to positive instance for in our problem. We simply take $\Phi = C$, and have $S_\beta(\Phi, V) \geq \sum_{\ell=1}^{q} m - |B_\ell|$ since, for each voter ℓ, there is an $e \in C$ that is in B_ℓ by definition of a hitting set. Summing the scores of the most preferred alternatives, $\max_{a \in \Phi} m - v_\ell(a) \geq m - |B_\ell|$, over all voters, gives $S_\beta(\Phi, V) \geq t$.

Suppose we have a negative hitting set instance. Consider any Φ that maximizes $S_\beta(\cdot, V)$. If Φ does not hit some B_ℓ then let $a' = \operatorname{argmin}_{a \in \Phi} v_\ell(a)$. If $a' \neq z_{\ell j}$ for any j then $m - v_\ell(a') < m - \sum_{\ell=1}^{q} |B_\ell|$ and $S_\beta(\Phi, V) < t$. Otherwise $a' = z_{\ell 1}$; but this implies that we can replace each such $z_{\ell 1} \in \Phi$ by some $b \in B_\ell$, which further implies that Φ hits *every* such B_ℓ and is thus a hitting set solution (contradiction). Hence, $S_\beta(\Phi, V) < t$. $\square$

We can formulate this NP-hard problem as an integer program (IP) with $m(n+1)$ variables and $1 + mn + n$ constraints. We note that [20] provide a similar IP for the Chamberlin and Courant proportional scheme. Let $x_i \in \{0, 1\}$, $i \leq m$ denote whether alternative a_i appears in the recommendation set Φ, and let $y_{\ell i} \in \{0, 1\}$, $\ell \leq n, i \leq m$ denote whether a_i is the most preferred element in Φ for voter ℓ. We then have:

$$\max_{x_i, y_{\ell i}} \quad \sum_{\ell \in N} \sum_{i=1}^{m} \alpha_\ell(a_i) \cdot y_{\ell i} \tag{4}$$

$$\text{subject to} \quad \sum_{i=1}^{m} x_i \leq K, \tag{5}$$

$$y_{\ell i} \leq x_i, \qquad \forall \ell \leq n, i \leq m \tag{6}$$

$$\sum_{i=1}^{m} y_{\ell i} = 1, \qquad \forall \ell \leq n. \tag{7}$$

Constraint (5) limits the slate to at most K alternatives (a optimal set of size less than K can be expanded arbitrarily to size K, since score is nondecreasing in size). Constraints (6) and (7) ensure voters benefit only from alternatives in Φ, and benefit from exactly *one* such element. The objective is simply $S_\alpha(\Phi, V)$. An optimal solution will always have $y_{\ell i} = 1$ where a_i is ℓ's most preferred alternative in the set defined by the x_i.

The IP may not scale to large problems. Fortunately, this is a constrained submodular maximization, which admits a simple greedy algorithm with approximation guarantees [18].

Algorithm Greedy**.** We receive inputs α, V and integer $K > 0$. Initially $\Phi_0 \leftarrow \emptyset$. We then update Φ iteratively K times, each time updating the recommendation set by adding the item that increases score the most, i.e., $\Phi_i \leftarrow \Phi_{i-1} \cup \{\operatorname{argmax}_{a \in A} S(\Phi_{i-1} \cup \{a\}, V)\}$. We output Φ_K.

Theorem 7. *For any given preference profile V, the function $S(\cdot, V)$ defined over 2^A, with $S(\emptyset, V) = 0$, is submodular and non-decreasing. Consequently, the constrained maximization of Eq. (2) can be approximated within a factor of $1 - \frac{1}{e}$ by* Greedy*. That is, $\frac{S(\mathsf{Greedy}, V)}{S(\Phi^*, V)} \geq 1 - \frac{1}{e}$.*

NP-hardness of our limited choice model.

Proof. Let $\Phi \subseteq \Phi' \subseteq A$, $a \in A$ and $v \in V$. It is clear that $S(\Phi, v) \leq S(\Phi', v)$. Since $S(\cdot, v)$ is non-decreasing for any vote v v, it is non-decreasing over profiles V, i.e., $S(\Phi', V) \geq S(\Phi, V)$.

If a is v's strictly most preferred alternative among those in Φ', then $S(\Phi \cup \{a\}, v) = S(\Phi' \cup \{a\}, v) = \alpha(v(a))$. Since $S(\Phi, v) \leq S(\Phi', v)$, this implies $S(\Phi \cup \{a\}, v) - S(\Phi, v) \geq S(\Phi' \cup \{a\}, v) - S(\Phi', v)$. If a is not strictly most preferred by v within the set Φ', then $S(\Phi' \cup \{a\}, v) = S(\Phi', v)$, hence $S(\Phi' \cup \{a\}, v) - S(\Phi', v) = 0$. Since $S(\Phi \cup \{a\}, v) \geq S(\Phi, v)$, again we have $S(\Phi \cup \{a\}, v) - S(\Phi, v) \geq S(\Phi' \cup \{a\}, v) - S(\Phi', v)$. This implies, by definition, the submodularity of $S(\cdot, v)$ for any vote v. Since the sum of submodular functions is also submodular, $S(\cdot, V)$ is submodular for profiles V. The $1 - \frac{1}{e}$ approximation ratio follows from [18]. □

Constructing a slate of K alternatives maximizing total positional score is similar to the K-*medians problem*, where at most K facilities (alternatives) need to be located to serve their nearest customers (voters) while minimizing the total distance between customers and their nearest facility. Distance corresponds to voter dissatisfaction with alternatives in the slate (i.e., negated α-score). Most work on K-medians focuses on metric settings—our problem does not have such an interpretation—and little work has been done on non-metric settings (see, e.g., [1]) especially w.r.t. ordinal preferences. Facility location is another related problem, though the aim is usually to minimize the total cost of opening facilities and serving the nearest customers, with no constraints on the number of facilities. In our setting, the tradeoff between a positional score and the cost of alternatives is not well-defined unless the score is a surrogate for profit/cost.

Experiments on APA Dataset The American Psychological Association (APA) held a presidential election in 1980, where roughly 15,000 members expressed preferences for 5 candidates—5738 votes were full rankings. Members roughly divide into "academics" and "clinicians," who are on "uneasy terms," with classes of voters tending to favor one group of candidates over another (candidate groups $\{1, 3\}$ and $\{4, 5\}$ appeal to different voters, with candidate 2 somewhere in the middle) [9]. We apply our model to the full-ranking dataset with $K = 2$ and Borda scoring. We expect our model to favour "diverse" pairings (with academic-clinician pairings scoring highest). Indeed, this is what we obtain—the optimal recommendation set is $\{3, 4\}$ with $S_\beta = 18182$. In fact, the for highest scoring pairings are all diverse in this sense. Greedy outputs the diverse set $\{1, 5\}$ with score 17668, whereas selecting the top two candidates from the Borda or Kemeny rankings gives $\{1, 3\}$ with score 17352, an inferior (and non-diverse) pairing. The quality of the Borda/Kemeny approximations is even worse with more "dramatic" positional scoring (i.e., with scoring functions that exaggerate the score difference between different positions as discussed below).

Experiments on Sushi Dataset We experiment with a *sushi dataset* consisting of 10 varieties of sushi, and 5000 full preference orderings elicited across Japan [14]. In our budgeted (limited choice) setting, we might imagine a banquet in which only a small selection of sushi types can be provided to a large number of guests. Table 1 shows the approximation ratios of various algorithms for different slate sizes K, using an exponentially decreasing PSF $\alpha_{\text{exp}}(i) = 2^{m-i}$. CPLEX was used to solve IP (4) to determine optimal slates (computation times are shown in the table). We evaluate our greedy algorithm, random sets of size K (avg. over 20 instances for each K), and Borda and Kemeny (where we use the top K candidates as the recommendation set). We see that the Greedy algorithm always finds the optimal slate (and, in fact, does so for all $K \leq 9$), yet does so very quickly (under 1s.) relative to CPLEX optimization. Borda and Kemeny provide decent approximations, but are not generally optimal. Unsurprisingly, for large K (relative to $|A|$) random subsets do well, but perform poorly for small K. Results using Borda scoring are similar except that, unsurprisingly, random sets yield better approximations, since Borda count penalizes less for recommending lower-ranked alternatives than the exponential PSF.

In both the APA and sushi dataset, Borda and Kemeny rankings offer good approximations, though this is likely due, in part, to correlation effects: items that are highly preferred by an agent of one type are also reasonably preferred by agents of other types. This is in contrast to a situation (cf. Fig. 1) where one group's highly ranked candidate is strongly dispreferred by other groups.

K	Greedy	Borda	Kemeny	Random	CPLEX (sec.)
2	1.0	1.0	0.932	0.531	49.1
3	1.0	0.986	0.949	0.729	90.38
5	1.0	0.989	0.970	0.813	20.32
7	1.0	1.0	1.0	0.856	13.16

Table 1: Results on the sushi dataset with 10 alternatives and 5000 full rankings. Four algorithms are shown in the columns along with their approximation ratio for each K. CPLEX solution times are shown in the last column.

4 General Budgeted Social Choice

In the limited choice model, we assume the main bottleneck is the size of the recommendation set Φ. Once Φ is determined, voters are free to choose their favourite alternative. We can generalize the problem slightly by assigning costs to the alternatives and limiting the total cost of Φ (rather than its size). A more significant generalization involves also assuming some cost associated with each voter that benefits from an element in Φ. For example, a company that decides to manufacture different product configurations must pay certain fixed production costs for each configuration (e.g., capital expenditures); in addition, there are per-unit costs associated with producing each unit of the product (e.g., labour/material/transporation costs).[2]

For each alternative $a \in A$, let t_a be its *fixed cost* and u_a its *unit cost*. We assume a total budget B that cannot be exceeded by Φ. However, since unit costs vary across $a \in \Phi$, a decision maker cannot simply propose a recommendation set Φ: allowing agents to choose their most preferred alternative freely may result in exceeding the budget (e.g., if voters all choose expensive alternatives). Instead, the decision maker produces an *assignment* of alternatives to agents that maximizes social welfare.

Definition 8. *A* recommendation function $\Phi : N \to A$ *assigns agents to alternatives. Given PSF α and profile V, the* α-score *of Φ is:*

$$S_\alpha(\Phi, V) = \sum_{\ell \in N} \alpha_\ell(\Phi(\ell)) \; . \tag{8}$$

Let $\Phi(N) = \{a : \Phi^{-1}(a) \neq \emptyset\}$ be the set recommended alternatives. The cost *of Φ is:*

$$C(\Phi) = \sum_{a \in A} \mathbf{1}[a \in \Phi(N)] \cdot t_a + \sum_{\ell \in N} u_{\Phi(\ell)} \; . \tag{9}$$

The first component in the cost of Φ corresponds to the fixed costs of the recommended alternatives, and second reflects the total unit costs. We now define the general budgeted problem:

Definition 9. *Given alternatives A, profile V, PSF α and budget $B > 0$, the* budgeted social choice problem *is:*

$$\max_{\Phi} S_\alpha(\Phi, V) \quad \textit{subject to} \quad C(\Phi) \leq B. \tag{10}$$

We say that the problem is *infeasible* if every Φ has total cost exceeding B. As in the limited choice model, we define the problem using PSFs to measure utility and total social welfare as our optimization criterion; but other variants are possible. We mention a few interesting special cases:

- If we wish to leave some voters unassigned an alternative, we can model this using a *dummy item* d with $t_d = u_d = 0$. Voter preference for d can default to the bottom of each ordering or can reflect genuine preference for being unassigned. All such problems are feasible.

[2]The possibility of extending proportional representation schemes to making tradeoffs between representativeness and committee size is mentioned as an interesting possibility by Chamberlin and Courant [4].

- When $t_a = t$ (i.e., fixed charges are constant) and $u_a = 0$ for all $a \in A$, this corresponds to the limited choice model for $K = \lfloor B/t \rfloor$. Since unit costs are zero, the optimal Φ will always assign a voter to its preferred alternative, and a recommendation set of size B/t can be used. If unit costs are constant as well, $u_a = u$, similarly we have $K = \left\lfloor \frac{B-nu}{t} \right\rfloor$.
- When fixed costs vary, but unit costs $u = u_a$ are constant, we generalize the limited choice model slightly: because unit costs are identical, agents can still select their preferred alternative from a slate (of varying size) whose total fixed cost does not exceed $B - nu$.
- If every recommendation function Φ satisfies $C(\Phi) \leq B$ (e.g., if all charges are zero), we are in a *fully personalizable setting*, and each agent is assigned their their most preferred alternative.

Input: α, V, B, fixed costs t and unit costs u.

$\Phi \leftarrow \emptyset$ and $A^* \leftarrow \emptyset$
Let N_Φ denote $\{\ell : \Phi(\ell)$ is undefined$\}$
{PHASE 1 : ADD ITEMS WITH BEST SWEET SPOT}
loop
    for $a \in A \backslash A^*$ **do**
        $J \leftarrow \{\ell : a \succ_\ell r(\ell)$ and $u_a \geq u_{r(\ell)}\}$
        $N_a = \overline{N_\Phi} \cup J$
        $R_a = \left[\frac{\alpha_\ell(a)}{u_a}\right]_{\ell \in \overline{N_\Phi}} \cup \left[\frac{\alpha_\ell(a) - \alpha_\ell(\Phi(\ell))}{u_a - u_{\Phi(\ell)}}\right]_{\ell \in J}$
        $SR_a \leftarrow$ sort R_a to get $(\beta_1/\gamma_1, \ldots, \beta_{|R_a|}/\gamma_{|R_a|})$ {If $\gamma_i = 0$ then the "ratio" gets put in front of sorted list. For another denominator $\gamma_j = 0$ we then compare whether $\beta_i > \beta_j$.}
        reorder N_a to $[\ell_1^a, \ldots, \ell_{|N_a|}^a]$ so ℓ_i^a corresponds to β_i/γ_i
        Let r_a^* and i_a^* be the max and argmax over i of $\{\frac{\sum_{j=1}^i \beta_j}{t_a + \sum_{j=1}^i \gamma_j} : i \in |SR_a|$ and $t_a + \sum_{j=1}^i \gamma_j \leq B - C(\Phi)\}$ if $\emptyset$ then set to undefined.
    end for
    if $a^* \leftarrow \text{argmax}_{a \in A \backslash A^*} r_a^*$ is undefined **then**
        break {*all* r_a^* is undefined—over budget}
    else
        append a^* to A^*
        update Φ with $\{(\ell_i^{a^*}, a^*) : 1 \leq i \leq i_a^*\} \cup \{(\ell, a^*) : \ell \in N, a^* \succ_\ell \Phi(\ell)$ and $u_{a^*} \leq u_{\Phi(\ell)}\}$
    end if
end loop
{PHASE 2: BACKTRACKING}
while Φ incomplete **do**
    $a^* \leftarrow$ pop A^*
    remove $\{(\ell, a^*) : \ell \in N, \Phi(\ell) = a^*\}$ from Φ
    $\tilde{A} \leftarrow \{a \in A : t_a + \sum_{\ell \in \overline{N_\Phi}} u_a \leq B - C(\Phi)\}$
    if $\tilde{A} \neq \emptyset$ **then**
        $a^* \leftarrow \text{argmax}_{a \in \tilde{A}} \sum_{\ell \in \overline{N_\Phi}} \alpha_\ell(a)$
        update Φ with $\{(\ell, a^*) : \ell \in \overline{N_\Phi}\}$ and **break**
    end if
end while
return INFEASIBLE if $\Phi = \emptyset$, otherwise Φ

Fig. 2: The SweetSpotGreedy (SSG) algorithm.

We note that the general problem can be modified in other ways. For instance, we may ignore budget, and instead allow an explicit tradeoff between social welfare (voter happiness) and costs, and simply maximize total score less total cost of Φ. In this way, unit cost would not prevent assignment of some more preferred alternative to a voter if the voter's satisfaction outweighed the unit cost (once a fixed charge is incurred) or if it maximized surplus. This would better reflect a profit maximization motive in some settings (treating user satisfaction as a measure of willingness to pay). Our model as defined above is more appropriate in settings where users of a recommended alternative cannot be (directly) charged for its use (e.g., as in the case of certain public goods, corporate promotions or incentive programs, etc.).

Our general budgeted social choice problem is related to several problems arising in operations research. When fixed costs vary but unit costs are constant, the problem is similar to *budgeted maximum coverage* [16], given by a set E of weighted elements and a family of subsets of E with costs, with the goal of finding a covering with total cost under a budget that maximizes total weight of the covered elements. Our problem is slightly different: viewing voters as elements and alternatives as the cover set, we have a score for each element-alternative pair. Our problem is more closely related to the recently defined *generalized maximum coverage* problem [5], with a weight and cost for each cover set-element pair (in our model, the unit costs would be constant), and a cost for each cover set (i.e., fixed costs). Unlike budgeted social choice, coverage of all elements is not required. As discussed earlier, facility location is also similar to our budgeted setting, though it typically places no restrictions on budget (it is instead absorbed into the objective). Akin to unit costs in our model, [12] studies facility location when facility costs include the cost of customers being served (cost is assumed concave in number of customers).

We can formulate the general budgeted problem as an IP similar to IP (4) (with the same number of variables and constraints):

$$\max_{x_i, y_{\ell i}} \quad (4)$$

$$\text{subject to} \quad \left[\sum_{i=1}^{m} t_{a_i} x_i\right] + \left[\sum_{\ell \in N} \sum_{i=1}^{m} u_{a_i} y_{\ell i}\right] \leq B, \tag{11}$$

$$\text{and } (6), (7).$$

An approximation algorithm for the general problem is complicated by the existence of unit costs. We may need to limit the assignment of expensive alternatives, despite "demand" from many voters. When unit costs are zero (or very low compared to fixed costs), the problem reduces to selecting a subset of alternatives as discussed above.

Still we develop a greedy heuristic algorithm called SweetSpotGreedy (or SSG). The main intuition behind our greedy heuristic is to successively "cover" or "satisfy" agents of a certain type by selecting their most preferred alternative. For a given $a \in A$, we sort voters based on their ranking of a and then compute the *bang-per-buck* ratio of assigning a to the first i voters—i.e., total score divided by total cost of assigning a to these i voters. We pick the index i^*_a that maximizes the bang-per-buck ratio r^*_a. This is the *sweet spot* since the marginal score improvement of assigning more a to additional voters doesn't justify the incremental cost of producing more of a. We then add to the recommendation function Φ that a^* with the greater ratio $r^*_{a^*}$ and assign it to the $i^*_{a^*}$ agents who prefer it most. We repeat this procedure after removing the previously assigned a, each time selecting a new a^* and recommending it to the voters that maximize its bang per buck. See Fig. 2 for further details. The first phase of the algorithm as described may not produce a feasible assignment Φ: the budget may be exhausted before all agents are assigned an alternative. A second backtracking phase produces a feasible solution by rolling back the most recent updates to Φ from Phase 1. Each time an alternative is rolled back, we try to find an $a \in A$ that can be assigned to all unassigned agents without depleting the budget. If after full backtracking this can't be achieved, the instance is infeasible (see Proposition 10).

SSG has running time $O(m^2 n \log n)$. The intuition behind our algorithm is similar in spirit to the $1 - \frac{1}{e} - o(1)$ approximation algorithm for generalized maximum coverage [5]. However, that algorithm is theoretical, requiring $O(m^2 n)$ calls to a fully polytime approximation scheme for the maximum density knapsack problem.

Proposition 10. SSG *returns INFEASIBLE iff the instance is infeasible.*

Proof. The if direction is obvious, since SSG always maintains feasibility of any solution Φ returned. If it returns INFEASIBLE, the backtracking phase must be entered and exited with $\Phi = \emptyset$. This implies $A^* = \emptyset$ since we have tried to roll back all additions to A^* only to discover there is no $a \in A$ with $t_a + n \cdot u_a \leq B$; that is, there is no single item assignable to all agents that doesn't exceed budget. This obviously implies infeasibility of the instance, since assigning the a minimizing $t_a + n \cdot u_a$ to all agents is the lowest cost Φ regardless of score. $\square$

As discussed above, when unit costs are zero our problem reduces to selecting a subset $\Phi \subseteq A$ with total fixed cost less than B. When fixed costs are constant, this essentially reduces to the limited choice problem. In fact, SSG outputs the same recommendation function as that outputted by Greedy (converting the set to a function in the obvious way).

Proposition 11. *If $u_a = 0$ and $t_a = 1$ for all $a \in A$ then* SSG *outputs the same recommendation as* Greedy. *Hence, it has an approximation ratio $1 - \frac{1}{e}$.*

Proof. To see that SweetSpotGreedy reduces to Greedy notice that in the first iteration of Phase 1, Φ is empty, and because unit costs are zero, the sweet spot for any $a \in A$ is to recommend a to all $\ell \in N$. So Φ gets updated by assigning the alternative a_1^*, which maximizes the gain in total score, to all agents. On the next iteration, again because unit costs are zero, the sweet spot for any $a \in A - \{a_1^*\}$ is to recommend a to all agents ℓ that prefer it over a_1^*. Hence, Φ is updated by including the best alternative a_2^*. This observation holds in all subsequent iterations: the sweet spot for any unused alternative a is to recommend it to all agents who prefer a over existing elements of Φ. This is exactly what Greedy does, picking an alternative in each iteration (which is implicitly recommended to all agents that prefer it over existing alternatives in Φ) that greedily maximizes the gain in score. The $1 - \frac{1}{e}$ approximation ratio follows from Theorem 7. $\square$

Experiments on Sushi Data We experimented with SweetSpotGreedy on the sushi dataset. In our first experiment, we randomly generate fixed costs while holding unit costs at zero. This corresponds to the special case discussed above that only slightly generalized the limited choice model. Integer fixed costs for the sushi varieties are chosen uniformly at random from $[20, 50)$, while the budget is set to 100. This means the recommendation set typically contains 2 to 5 items. We compared the performance of SSG against the optimal solution (computed using the IP above, solved using CPLEX) on 20 random instances (note that the preference profile is held fixed, corresponding to the data set). Both Borda scoring and the exponential PSF α_{exp} (see above) were tested and give similar results. With Borda, SSG is within 99% of the optimal recommendation function on average (it often attains the optimum, and is never worse than 94% of optimal). Its running times lie in the range $[1.91, 2.34]$ seconds (with a very simple Python implementation). Meanwhile, CPLEX has an average solution time of 114 seconds (the range is $[69s, 176s]$).

In a second experiment, we varied both fixed and unit costs with fixed costs substantially larger than unit costs. Specifically, integer unit costs were chosen uniformly at random from $[1, 4]$ and integer fixed costs from $[5000, 10000]$. We fixed the budget at 35000, which allows roughly 3 unique alternatives to be recommended. We again compare SSG to the optimal recommendation function on 20 random instances. Using Borda counts, the greedy algorithm gives recommendation functions that are, on average, within 98% of optimal, while taking 2–5s. to run. In contrast, CPLEX takes 458s. on average (range $[130s, 1058s]$) to produce an optimal solution. We achieve similar results using the exponential PSF, with greedy attaining average performance of 97% of optimal, and taking 3–6s. while CPLEX averages 321s. (range $[131s, 614s]$). These experiments show that SweetSpotGreedy has extremely strong performance, quickly finding excellent approximations to the optimal recommendation sets, when fixed costs are much larger than unit costs.

5 Conclusion

We have introduced a new class of *budgeted* social choice problems that spans the spectrum from genuine consensus (or "one-size-fits-all") recommendation typically studied in social choice to fully personalized decision-making. The key feature of our model—the fact that some customization to the preferences of distinct groups of users may be feasible where complete individuation is not—is characteristic of many real-world scenarios. Given a diverse array of user preferences, a decision maker must offer/produce/recommend a limited number of alternatives for the user population. This naturally leads to social welfare maximization goals whose solutions, crudely speaking, involve grouping/clustering agents with similar preferences and selecting one alternative for each group. Our model includes certain schemes for proportional representation as special cases, and indeed motivates the possible application for proportional schemes to ranking and recommendation. Such an objective often favours *diversity*, as opposed to *popularity*, of the chosen alternatives. This work can be viewed, for example, as justifying from social choice and decision-theoretic principles, that the top few web search results should be diversified so as to appeal to a wide range of user interests. We showed that the optimization induced by budgeted social choice is NP-hard; but we developed fast, intuitive greedy algorithms that have, in the case of the special case of limited choice, theoretical approximation guarantees. Critically, our greedy algorithms empirically provide excellent approximations on some real-world ordinal preference datasets.

Extensions of this work include the exploration of several variations of the budgeted model. For example, one might impose separate budgets for fixed and unit costs. If social welfare acts as a surrogate for the decision-maker's revenue/profit or return on investment, and the decision-maker has other investment options (e.g. a government considering public projects) one may wish to relax the budget constraints and instead maximize the return on investment per unit cost. Deeper connections to the proportional voting schemes is also being explored.

References

[1] M. R. Ackermann, J. Blömer, and C. Sohler. Clustering for metric and non-metric distance measures. *19th Symposium on Discrete Algorithms (SODA-08)*, pp.799–808, Philadelphia, 2008.

[2] K. J. Arrow. A difficulty in the concept of social welfare. *J. of Political Economy* 58:328–346, 1950.

[3] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare* 6:157–165, 1989.

[4] J. R. Chamberlin and P. N. Courant. Representative deliberations and representative decisions: proportional representation and the Borda rule. *The American Political Science Review* 77(3):718–733, 1983.

[5] R. Cohen, L. Katzir. The generalized maximum coverage problem. *Inf. Process. Lett.*, 108:15–22, 2008.

[6] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *J. Artif. Intel. Research*, 10:243–270, 1998.

[7] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. *UAI-05*, pp.145–152, Edinburgh, 2005.

[8] J.-C. de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781.

[9] P. Diaconis. A generalization of spectral analysis with application to ranked data. *Annals of Statistics* 17(3):949–979, 1989.

[10] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. *World Wide Web Conference*, pp.613–622, New York, 2001.

[11] W. Gaertner. *A Primer in Social Choice Theory*. Oxford University Press, Oxford, 2006.

[12] M. Taghi Hajiaghayi, M. Mahdian, and V. S. Mirrokni. The facility location problem with general cost functions. *Networks*, 42(1):42–47, 2003.

[13] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897 – 1916, 2008.

[14] T. Kamishima, H. Kazawa, and S. Akaho. Supervised ordering: An empirical survey. *IEEE International Conference on Data Mining*, 673–676, 2005.

[15] J. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[16] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Proc. Letters*, 70(1):39–45, 1999.

[17] T. Lu and C. Boutilier. The unavailable candidate model: A decision-theoretic view of social choice. *ACM Conference on Electronic Commerce 2010*, pp.263–274, Cambridge, MA.

[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14(1):265–294, 1978.

[19] C. H. Papadimitriou, M. Schapira, and Y. Singer On the hardness of being truthful. *FOCS 2008* pp. 250–259.

[20] R. F. Potthoff and S. J. Brams. Proportional representation: broadening the options. *Journal of Theoretical Politics* 10:147–178, 1998.

[21] A. D. Procaccia, J. S. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare* 30(3):353–362, 2008.

[22] P. Young. Optimal voting rules. *Journal of Economic Perspectives* 9:51–64, 1995.

Tyler Lu and Craig Boutilier

Dept. of Computer Science, University of Toronto, Toronto, CANADA

Email: `{tl,cebly}@cs.toronto.edu`

An Algorithm for the Coalitional Manipulation Problem under Maximin

Michael Zuckerman, Omer Lev, and Jeffrey S. Rosenschein

Abstract

We introduce a new algorithm for the Unweighted Coalitional Manipulation problem under the Maximin voting rule. We prove that the algorithm gives an approximation ratio of $1\frac{1}{2}$ to the corresponding optimization problem. This is an improvement over a previously known algorithm that gave a 2-approximation. We also prove that our analysis is tight, i.e., there are instances on which a $1\frac{1}{2}$-approximation is the best the algorithm can achieve.

1 Introduction

Exploring the computational complexity of, and algorithms for, the *manipulation problem* is one of the most important research areas in computational social choice.

In an election, voters submit linear orders (rankings, or profiles) of the candidates (alternatives); a *voting rule* is then applied to the rankings in order to choose the winning candidate. In the prominent impossibility result proven by Gibbard and Satterthwaite [4, 5], it was shown that for any voting rule, a) which is not a dictatorship, b) which is onto the set of alternatives, and c) where there are at least three alternatives, then there exist profiles where a voter can benefit by voting insincerely. Submitting insincere rankings in an attempt to benefit is called *manipulation*.

There are several ways to circumvent this result, one of which is by using computational complexity as a barrier against manipulation. The idea behind this technique is as follows: although there may exist a successful manipulation, the voter must *discover* it before it can be used—but for certain voting rules, discovering a successful manipulation might be computationally hard. This argument was used already in 1989 by Bartholdi et al. [2], and in 1991 by Bartholdi and Orlin [1], where they proved, respectively, that second-order Copeland and Single Transferable Vote are both $\mathcal{NP}$-hard to manipulate.

Later, the complexity of coalitional manipulation was studied by Conitzer et al. [3]. In the coalitional manipulation problem, a coalition of potentially untruthful voters try to coordinate their ballots so as to make some preferred candidate win the election. Conitzer et al. studied the problem where the manipulators are weighted: a voter with weight l counts as l voters, each of weight 1. This problem was shown to be $\mathcal{NP}$-hard, for many voting rules, even for a constant number of candidates. However, it has been argued that a more natural setting is the unweighted coalitional manipulation (UCM) problem, where all voters have equal power. In a recent paper [6], Xia et al. established as one of their main results that UCM is $\mathcal{NP}$-hard under the Maximin voting rule, even for 2 untruthful voters.

In 2009, Zuckerman et al. [7] defined a natural optimization problem for the unweighted setting (i.e., Unweighted Coalitional Optimization, UCO): finding the minimal number of manipulators that is sufficient to make some predefined candidate win. It is proven, as a corollary of their results, that the heuristic greedy algorithm proposed in the paper gives a 2-approximation to the UCO problem under Maximin. Here, we further study the UCO problem under Maximin, proposing a new greedy algorithm that gives a $1\frac{1}{2}$-approximation to the problem.[1] Then we provide an example showing that the approximation ratio of the

[1] Strictly speaking, our algorithm is for the *decision* problem, but since the conversion of our algorithm to one for the optimization problem is straightforward, we consider it an approximation algorithm for the

algorithm is not better than $1\frac{1}{2}$.

2 The Maximin Voting Rule, Manipulation and Condorcet winner

An election consists of a set $C = \{c_1, \ldots, c_m\}$ of candidates, and a set $S = \{v_1, \ldots, v_{|S|}\}$ of voters. Each voter provides a total order on the candidates (i.e., each voter submits a linear ranking of all the candidates). The setting also includes a *voting rule*, which is a function from the set of all possible combinations of votes to C.

The maximin voting rule is defined as follows. For any two distinct candidates x and y, let $N(x, y)$ be the number of voters who prefer x over y. The *maximin score* of x is $S(x) = \min_{y \neq x} N(x, y)$. The candidate with the highest maximin score is the winner.

Definition 2.1. In the Constructive Coalitional Unweighted Manipulation (CCUM) problem, we are given a set C of candidates, with a distinguished candidate $p \in C$, a set of (unweighted) voters S that have already cast their votes (these are the non-manipulators), and a set T of (unweighted) voters that have not yet cast their votes (these are the manipulators). We are asked whether there is a way to cast the votes in T so that p wins the election.

Definition 2.2. In the Unweighted Coalitional Optimization (UCO) problem we are given a set C of candidates, with a distinguished candidate $p \in C$, and a set of (unweighted) voters S that have already cast their votes (the non-manipulators). We are asked for the *minimal* n such that a set T of size n of (unweighted) manipulators can cast their votes in order to make p win the election.

Remark 2.3. We implicitly assume here that the manipulators have full knowledge about the non-manipulators' votes. Unless explicitly stated otherwise, we also assume that ties are broken adversarially to the manipulators, so that if p ties with another candidate, p loses. The latter assumption is equivalent to formulating the manipulation problems in their *unique winner* version, when one assumes that all candidates with maximal score win, but asks that p be the only winner.

Throughout this paper we will use the convention, unless explicitly stated otherwise, that $|C| = m$, $|S| = N$ and $|T| = n$. We will denote $N_i(x, y) = |\{j \mid x \succ_j y, \succ_j \in S \cup \{1, \ldots, i\}\}|$. That is, $N_i(x, y)$ will denote the number of voters from S and from the first i voters of T that prefer x over y (assuming S is fixed, and fixing some order on the voters of T). Furthermore, we will denote by $S_i(c)$ the accumulated score of candidate c from the voters of S and the first i voters of T. By definition, for each $c \in C$, $S_i(x) = \min_{y \neq x} N_i(x, y)$. Also, we denote for $x \in C$, $\mathrm{MIN}_i(x) = \{y \in C \setminus \{x\} \mid S_i(x) = N_i(x, y)\}$. We denote for $0 \leq i \leq n$, $\mathrm{ms}(i) = \max_{c \in C \setminus \{p\}} S_i(c)$. That is, $\mathrm{ms}(i)$ is the maximum score of the opponents of p after i manipulators have voted.

Definition 2.4. The *Condorcet winner* of an election is the candidate who, when compared with every other candidate, is preferred by more voters.

3 The Algorithm

Our algorithm for the CCUM problem under the maximin voting rule is given as Algorithm 1 (see the final page of the paper). It works as follows: fix some order on the manipulators;

optimization problem.

the current manipulator i ranks p first. He then builds a digraph $G^{i-1} = (V, E^{i-1})$, where $V = C \setminus \{p\}$, $(x, y) \in E^{i-1}$ iff ($y \in \mathrm{MIN}_{i-1}(x)$ and $p \notin \mathrm{MIN}_{i-1}(x)$). He iterates over the candidates that have not yet been ranked in his preference list. If there are candidates with an out-degree 0, then the manipulator adds such a candidate who has the lowest score (among the candidates with an out-degree 0) to his preference list. Note that the candidates with out-degree 0 are kept in stacks in order to guarantee a DFS-like order among candidates with the same score. This is needed for Lemma 5.5 to work. Otherwise, if there are no candidates with out-degree 0, then the algorithm tries to find a cycle with two adjacent vertices having the lowest score. If it finds such a cycle, then it picks the front vertex of these two. Otherwise, any candidate with the lowest score is chosen. After a candidate b is added to the manipulator's preference list, for each candidate y who has an outgoing edge (y, b), the algorithm removes all the outgoing edges of y, puts it into the appropriate stack, and assigns b to be y's "father" (this assignment is used to analyze the algorithm).

Note the subtle difference between calculating the scores in Algorithm 1 in this paper, as compared to in Algorithm 1 in [7]. In the latter, the manipulator i calculates what the score would be of the current candidate x if he put x at the current place in his preference list; in the algorithm we are now presenting, manipulator i just calculates $S_{i-1}(x)$. This difference is due to the fact that here, when we calculate the score of x, we know whether $d_{out}(x) > 0$, i.e., we know whether the score of x will grow by 1 if we put it at the current available place. So we separately compare the scores of candidates with out-degree > 0, and the scores of candidates with out-degree 0.

Definition 3.1. We refer to an iteration of the main **for** loop in lines 3–37 of Algorithm 1 as a *stage* of the algorithm. That is, a stage of the algorithm is a vote of any manipulator.

The intuition behind Algorithm 1 is as follows. The algorithm tries in a greedy manner to maximize the score of p, and to minimize the scores of p's opponents. To achieve this, it always puts p first in the preference lists, making the score of p grow by 1 with each manipulator. Regarding p's opponents, it tries first to rank candidates without any outgoing edges from them, since their score will not grow this way (because their score is achieved vs. candidates who were already ranked before them). When there are no candidates without outgoing edges, the algorithm finds the candidate with the minimal score, and ranks it in the next place in the preference list. After ranking each candidate, the edges in the graph are updated, so that all candidates whose minimal candidate has already been ranked, will be with outgoing degree 0. For an edge (x, y), if y has already been ranked, we remove all the edges going out from x, since if we rank x now, its score won't go up, and so it does not depend on other candidates in $\mathrm{MIN}_{i-1}(x)$. There is no need of an edge (x, y) if $p \in \mathrm{MIN}_{i-1}(x)$, since for all $x \in C \setminus \{p\}$, p is always ranked above x, and so whether y is ranked above x or not, the score of x will not grow.

Definition 3.2. In the digraph G^i built by the algorithm, if there exists an edge (x, y), we refer to $N_i(x, y) = S_i(x)$ as the *weight* of the edge (x, y).

4 2-approximation

We first prove that Algorithm 1 has an approximation ratio of 2. We then use this result in the proof of the $1\frac{1}{2}$ approximation ratio. The proof of Theorem 4.1 via Lemma 4.2 and Lemma 4.3 is quite similar to the proof of Theorem 3.16 in [7].

Theorem 4.1. *Algorithm 1 has a* 2*-approximation ratio for the UCO problem under the maximin voting rule.*

To prove the above theorem, we first need the following two lemmas. In the first one we prove that a certain sub-graph of the graph built by the algorithm contains a cycle passing through some distinguished vertex. We first introduce some more notation.

Let $G^i = (V, E^i)$ be the directed graph built by Algorithm 1 in stage $i+1$. For a candidate $x \in C \setminus \{p\}$, let $G^i_x = (V^i_x, E^i_x)$ be the graph G^i reduced to the vertices that were ranked below x in stage $i+1$, including x. Let $V^i(x) = \{y \in V^i_x \mid \text{there is a path in } G^i_x \text{ from } x \text{ to } y\}$. Also, let $G^i(x)$ be the sub-graph of G^i_x induced by $V^i(x)$.

Lemma 4.2. *Let i be an integer, $0 \le i \le n-1$. Let $x \in C \setminus \{p\}$ be a candidate. Denote $t = ms(i)$. Suppose that $S_{i+1}(x) = t+1$. Then $G^i(x)$ contains a cycle passing through x.*

Proof. First of all note that for all $c \in V^i(x)$, $S_i(c) = t$. It follows from the fact that by definition $S_i(c) \le t$. On the other hand, $S_i(x) = t$, and all the other vertices in $V^i(x)$ were ranked below x. Together with the fact that the out-degree of x was greater than 0 when x was picked, it gives us that for all $c \in V^i(x)$, $S_i(c) \ge t$, and so for all $c \in V^i(x)$, $S_i(c) = t$. We claim that for all $c \in V^i(x)$, $\mathrm{MIN}_i(c) \subseteq V^i(x)$. If, by way of contradiction, there exists $c \in V^i(x)$ s.t. there is $b \in \mathrm{MIN}_i(c)$ where $b \notin V^i(x)$, then $b \notin V^i_x$, since otherwise, if $b \in V^i_x$, then from $c \in V^i(x)$ and $(c, b) \in E^i_x$ we get that $b \in V^i(x)$. So $b \notin V^i_x$, which means that b was ranked by $i+1$ above x. After we ranked b we removed all the outgoing edges from c, and so we chose c before x since $d_{out}(c) = 0$ and $d_{out}(x) > 0$ (since the score of x went up in stage $i+1$). This contradicts the fact that $c \in V^i(x) \subseteq V^i_x$. Therefore, for every vertex $c \in V^i(x)$ there is at least one edge in $G^i(x)$ going out from c. Hence, there is at least one cycle in $G^i(x)$. Since at the time of picking x by voter $i+1$, for all $c \in V^i(x)$, $d_{out}(c) > 0$, and by the observation that for all $c \in V^i(x)$, $S_i(c) = t$, we have that the algorithm picked the vertex x from a cycle (lines 21–22 of the pseudocode). ◻

In the next lemma we put forward an upper bound on the growth rate of the scores of p's opponents.

Lemma 4.3. *For all $0 \le i \le n-2$, $ms(i+2) \le ms(i) + 1$*

Proof. Let $0 \le i \le n-2$. Let $x \in C \setminus \{p\}$ be a candidate. Denote $t = \mathrm{ms}(i)$. By definition, $S_i(x) \le t$. We would like to show that $S_{i+2}(x) \le t+1$. If $S_{i+1}(x) \le t$, then $S_{i+2}(x) \le S_{i+1}(x) + 1 \le t+1$, and we are done. So let us assume now that $S_{i+1}(x) = t+1$.

Let $V^i(x)$ and $G^i(x)$ as before. By Lemma 4.2, $G^i(x)$ contains at least one cycle. Let U be one such cycle. Let $a \in U$ be the vertex that was ranked highest among the vertices of U in stage $i+1$. Let b be the vertex before a in the cycle: $(b, a) \in U$. Since b was ranked below a at stage $i+1$, it follows that $S_{i+1}(b) = S_i(b) \le t$.

Suppose, for contradiction, that $S_{i+2}(x) > t+1$. Then the score of x went up in stage $i+2$, and so when x was picked by $i+2$, its out-degree in the graph was not 0. x was ranked by $i+2$ at place s^*. Then b was ranked by $i+2$ above s^*, since otherwise, when we had reached the place s^*, we would not pick x since b would be available (with out-degree 0, or otherwise—with score $S_{i+1}(b) \le t < t+1 = S_{i+1}(x)$)—a contradiction.

Denote by Z_1 all the vertices in $V^i(x)$ that have an outgoing edge to b in $G^i(x)$. For all $z \in Z_1$, $b \in \mathrm{MIN}_i(z)$, i.e., $S_i(z) = N_i(z, b)$. We claim that all $z \in Z_1$ were ranked by $i+2$ above x. If, by way of contradiction, there is $z \in Z_1$, s.t. until the place s^* it still was not added to the preference list, then two cases are possible:

1. If $(z, b) \in E^{i+1}$, then after b was added to $i+2$'s preference list, we removed all the outgoing edges of z, and we would put in z (with out-degree 0) instead of x, a contradiction.

2. $(z, b) \notin E^{i+1}$. Since $(z, b) \in E^i$, we have $S_i(z) = N_i(z, b)$. Also since z was ranked by $i+1$ below x, it follows that $S_i(z) = t$. So from $(z, b) \notin E^{i+1}$, we have that $S_{i+1}(z) = t$

and $N_{i+1}(z, b) = t + 1$. Therefore, when reaching the place s^* in the $i + 2$'s preference list, whether $d_{out}(z) = 0$ or not, we would not pick x (with the score $S_{i+1}(x) = t + 1$) since z (with the score $S_{i+1}(z) = t$) would be available, a contradiction.

Denote by Z_2 all the vertices in $V^i(x)$ that have an outgoing edge in $G^i(x)$ to some vertex $z \in Z_1$. In the same manner we can show that all the vertices in Z_2 were ranked in stage $i + 2$ above x. We continue in this manner, by defining sets $Z_3, \dots,$ where the set Z_l contains all vertices in $V^i(x)$ that have an outgoing edge to some vertex in Z_{l-1}; the argument above shows that all elements of these sets are ranked above x in stage $i + 2$. As there is a path from x to b in $G^i(x)$, we will eventually reach x in this way, i.e., there is some l such that Z_l contains a vertex y, s.t. $(x, y) \in E^i(x)$.

Now, if $(x, y) \in E^{i+1}(x)$, then since y was ranked by $i + 2$ above x, we have $S_{i+2}(x) = S_{i+1}(x) = t + 1$, a contradiction. And if $(x, y) \notin E^{i+1}(x)$, then since $(x, y) \in E^i(x)$ we get that $N_{i+1}(x, y) = t + 1$ and $S_{i+1}(x) = t$, a contradiction. □

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Let *opt* denote the minimum size of coalition needed to make p win. It is easy to see that $opt \geq \mathrm{ms}(0) - S_0(p) + 1$. We set $n = 2\mathrm{ms}(0) - 2S_0(p) + 2 \leq 2opt$. Then, by Lemma 4.3:

$$\mathrm{ms}(n) \leq \mathrm{ms}(0) + \left\lceil \frac{n}{2} \right\rceil = 2\mathrm{ms}(0) - S_0(p) + 1.$$

Whereas:

$$S_n(p) = S_0(p) + n = 2\mathrm{ms}(0) - S_0(p) + 2 > \mathrm{ms}(n).$$

So p will win when the coalition of manipulators is of size n. □

5 $1\frac{1}{2}$-approximation

Our next goal is to prove that Algorithm 1 has an approximation ratio of $1\frac{1}{2}$ when there are no 2-cycles in the graphs built by the algorithm.

Theorem 5.1. *For instances where there are no 2-cycles in the graphs G^i built by Algorithm 1, it gives a $1\frac{1}{2}$-approximation to the optimum.*

We first prove the following lemma regarding the length of the cycles in the digraphs built by the algorithm.

Lemma 5.2. *If for all $c \in C \setminus \{p\}$ it holds that $S_0(c) < \lfloor \frac{N}{2} \rfloor$, then during the run of the entire algorithm, in the graph built by the algorithm, there will be no cycles of length 2.*

Proof. Suppose that for all $c \in C \setminus \{p\}$ it holds that $S_0(c) < \lfloor \frac{N}{2} \rfloor$. By Lemma 4.3, it holds for all $c \in C \setminus \{p\}$ and all $0 \leq i \leq n - 2$, that $S_{i+2}(c) \leq \mathrm{ms}(i) + 1$. Then for all $0 \leq i \leq n$:

$$S_i(c) \leq \mathrm{ms}(0) + \left\lceil \frac{i}{2} \right\rceil < \left\lfloor \frac{N}{2} \right\rfloor + \left\lceil \frac{i}{2} \right\rceil \leq \left\lceil \frac{N+i}{2} \right\rceil.$$

Now if, by way of contradiction, there is a cycle of length 2 between vertices x and y after stage i, then $S_i(x) = N_i(x, y) < \lceil \frac{N+i}{2} \rceil$ and $S_i(y) = N_i(y, x) < \lceil \frac{N+i}{2} \rceil$, and then $S_i(y) = N_i(y, x) \leq \lfloor \frac{N+i}{2} \rfloor$. Hence, $N + i = N_i(x, y) + N_i(y, x) < \lceil \frac{N+i}{2} \rceil + \lfloor \frac{N+i}{2} \rfloor = N + i$, a contradiction. □

Lemma 5.3. *Suppose that there are no 2-cycles in the graphs built by the algorithm. Let $x \in C \setminus \{p\}$ be a candidate such that $S_{i+1}(x) = t + 1$ (where $t = ms(i)$), and let $G^i(x)$ be as described before Lemma 4.2. For each cycle U in $G^i(x)$, if U exists in G^{i+1}, i.e., after stage $i + 1$, then there are 3 distinct vertices a, b, c, s.t. $(c, b) \in U$, $(b, a) \in U$ and $S_{i+1}(b) = N_{i+1}(b, a) = S_{i+1}(c) = N_{i+1}(c, b) = t$.*

Proof. Let $U \subseteq E^i(x)$ be a cycle which stays also after $i + 1$ stages. Let a be the vertex which in stage $i + 1$ was chosen first among the vertices of U. Let b be the vertex before a in U, i.e., $(b, a) \in U$, and let c be the vertex before b in U, i.e., $(c, b) \in U$. Since there are no 2-cycles, a, b, c are all distinct vertices. Recall that for each $y \in V^i(x)$, $S_i(y) = t$. Since b was ranked below a in stage $i + 1$, we have $S_{i+1}(b) = N_{i+1}(b, a) = N_i(b, a) = S_i(b) = t$. If c was chosen after b in stage $i + 1$, then $S_{i+1}(c) = N_{i+1}(c, b) = N_i(c, b) = t$ and we are done. We now show that c cannot be chosen before b in stage $i + 1$. If, by way of contradiction, c were chosen before b, since after ranking a, $d_{out}(b) = 0$, it follows that when c was picked, its out-degree was also 0. Hence, there exists $d \in \mathrm{MIN}_i(c)$ which was picked by $i+1$ before c. And so, $S_{i+1}(c) = t$. On the other hand, since c was picked before b, we have $N_{i+1}(c, b) = t + 1 > S_{i+1}(c)$, and so the edge (c, b) does not exist in G^{i+1}, a contradiction to the fact that the cycle U stayed after stage $i + 1$. ☐

Lemma 5.4. *Let $x \in C \setminus \{p\}$ be a candidate such that $S_{i+1}(x) = t + 1$ (where $t = ms(i)$). Let $G^i(x)$ be as before. Then at least one cycle in $G^i(x)$ that passes through x, will stay after the stage $i + 1$, i.e., in G^{i+1}.*

Proof. In Lemma 4.2 we have proved that, in $G^i(x)$ at least one cycle passes through x. Since x appears in the preference list of $i + 1$ above all the $\mathrm{MIN}_i(x)$, it follows that each edge going out of x in $G^i(x)$, stays also in G^{i+1}. After we added x to the preference list of $i + 1$, all the vertices in all the cycles passing through x were added in some order to the preference list of $i + 1$, while they were with out-degree 0 at the time they were picked (it can be proved by induction on the length of the path from the vertex to x). Therefore, their "father" field was not null when they were picked. We have to prove that there is at least one cycle whose vertices were added in the reverse order (and then all the edges of the cycle stayed in G^{i+1}). Let $z_1 \in C \setminus \{p, x\}$ be some vertex such that $(x, z_1) \in G^i(x)$ and there is a path in $G^i(x)$ from z_1 to x. Let $z_2 = z_1.father$. As observed earlier, $z_2 \neq null$. We first show that when z_2 was picked by $i + 1$, it was with out-degree 0. Indeed, if, by contradiction, we suppose otherwise, then z_2 would have been picked after z_1 (the proof is by induction on the length of the shortest path from vertex to x, that each vertex such that there is a path from it to x was picked before z_2), and this is a contradiction to the fact that $z_2 = z_1.father$. Therefore, the "father" field of z_2 after stage $i + 1$ is not null. Let $z_3 = z_2.father$. If $z_3 = x$ then we are done because we have found a cycle $x \to z_1 \to z_2 \to z_3 = x$ which was ranked in stage $i + 1$ in the reverse order. Otherwise, by the same argument as before, we can show that when z_3 was picked, its out-degree was 0. This way we can pass from a vertex to its father until we reach p or null. We now show that we cannot reach p this way. Indeed, if, by contradiction, we reach p, then there is a path from x to p in G^i, and so all the vertices in this path, including x, were picked when their out-degree was 0, and this is a contradiction to the fact that the score of x went up in stage $i + 1$. Therefore, we cannot reach p when we go from a vertex to its father starting with z_1. Now, let z_j be the last vertex before null in this path. We would like to show that $z_j = x$. If, by contradiction, z_j was picked before x by voter $i + 1$, then all the vertices $z_{j-1}, \ldots, z_2, z_1$ would have been picked before x, when their out-degree is 0, and then x would have been picked when its out-degree is 0. This is a contradiction to the fact that x's score went up in stage $i+1$. Now suppose by contradiction that z_j was picked after x in stage $i + 1$. Then all the vertices that have a path from them to x, including z_1, would have been picked before z_j in stage $i + 1$, since the out-degree of

z_j was greater than 0 when it was picked. This is a contradiction to the fact that z_j was picked before z_1. So, $z_j = x$. This way we got a cycle $x \to z_1 \to \ldots \to z_{j-1} \to x$ which was ranked in the reverse order in stage $i+1$. $\square$

Lemma 5.5. *Suppose that there are no 2-cycles in the graphs built by the algorithm. Let $x \in C \setminus \{p\}$ be a candidate such that $S_{i+1}(x) = t+1$ (where $t = ms(i)$). Then after stage $i+2$ at least one of the following will hold:*

1. *There will be a vertex w in G^{i+2} s.t. $p \in MIN_{i+2}(w)$ and there will be a path from x to w.*
2. *There will be a vertex w in G^{i+2} with $S_{i+2}(w) \le t$, s.t. there will be a path from x to w.*

The proof of this lemma uses the same ideas as the proof of Lemma 5.4, and is omitted due to space limitations.

The next lemma is central in the proof of Theorem 5.1. It states that the maximum score of p's opponents grows rather slowly.

Lemma 5.6. *If there are no 2-cycles in the graphs built by the algorithm, then for all i, $0 \le i \le n-3$ it holds that $ms(i+3) \le ms(i)+1$.*

Proof. Let i, $0 \le i \le n-3$. Let $x \in C \setminus \{p\}$ be a candidate. Denote $\mathrm{ms}(i) = t$. We need to prove that $S_{i+3}(x) \le t+1$. If $S_{i+1}(x) \le t$, then similarly to Lemma 4.3 we can prove that $S_{i+3}(x) \le t+1$. So now we assume that $S_{i+1}(x) = t+1$. By Lemma 4.3, we have that $S_{i+2}(x) = t+1$. Suppose by contradiction that $S_{i+3}(x) = t+2$. x was ranked in stage $i+3$ at the place s^*. By Lemma 5.5 there exists a vertex w s.t. there is a path in G^{i+2} from x to w, and $p \in \mathrm{MIN}_{i+2}(w)$ or $S_{i+2}(w) \le t$. Then w was ranked in stage $i+3$ above the place s^*, because the score of x went up in stage $i+3$, and if, by contradiction, w was not ranked above the place s^*, then when we got to the place s^* we would prefer w over x. It is easy to see that all the vertices that have a path in G^{i+2} from them to w, and which were ranked below w in stage $i+3$, did not have their scores go up in that stage (since we took them one after another in the reverse order on their path to w when they were with out-degree 0). And as x was ranked below w, its score did not go up as well, and so $S_{i+3}(x) = S_{i+2} = t+1$, a contradiction. $\square$

We are now ready to prove the main theorem.

Proof of Theorem 5.1. Let *opt* denote the minimal size of the coalition of manipulators that can make p win the election. It is easy to see that $opt \ge \mathrm{ms}(0) - S_0(p) + 1$. We shall prove that Algorithm 1 will find a manipulation for $n = \left\lceil \frac{3\mathrm{ms}(0) - 3S_0(p) + 3}{2} \right\rceil \le \left\lceil \frac{3}{2} opt \right\rceil$. And indeed, by Lemma 5.6,

$$\mathrm{ms}(n) \le \mathrm{ms}(0) + \left\lceil \frac{n}{3} \right\rceil = \mathrm{ms}(0) + \left\lceil \frac{\mathrm{ms}(0) - S_0(p) + 1}{2} \right\rceil.$$

Whereas,

$$\begin{aligned} S_n(p) &= S_0(p) + n \\ &= S_0(p) + (\mathrm{ms}(0) - S_0(p) + 1) + \left\lceil \frac{\mathrm{ms}(0) - S_0(p) + 1}{2} \right\rceil \\ &= \mathrm{ms}(0) + 1 + \left\lceil \frac{\mathrm{ms}(0) - S_0(p) + 1}{2} \right\rceil \\ &> \mathrm{ms}(0) + \left\lceil \frac{\mathrm{ms}(0) - S_0(p) + 1}{2} \right\rceil \\ &\ge \mathrm{ms}(n). \end{aligned}$$

∎

Theorem 5.7. *The $1\frac{1}{2}$-approximation ratio of Algorithm 1 is valid also when there are 2-cycles in the graphs built by the algorithm.*

Proof. Due to space constraints, we will only provide a sketch of the proof, and we will omit the proofs of the lemmas (except of Lemma 5.9).

This following proof will show, in a way, that our algorithm is optimal in dismantling 2-cycles—if there are 2-cycles in G^i, then for every algorithm $ms^*(i) \geq ms(i)$. Once 2-cycles have been dismantled (and they cannot return), our algorithm performs a $1\frac{1}{2}$-approximation on the number of steps left, and thus, generally a $1\frac{1}{2}$-approximation on the optimal solution.

Lemma 5.8. *If there are no cycles of length 2 in a certain stage of the algorithm run (G^i), then no 2-cycles will be created in any further iteration—G^j ($j > i$) will have no 2-cycles.*

From now on, we shall assume G^0 contains at least one 2-cycle, with a Condorcet winner a.

Lemma 5.9. *If there is more than one 2-cycle at any stage i, there are no 2-cycles at stage $i+3$.*

Proof. Suppose $b_1, b_2, \ldots, b_t$ are the 2-cycle partners of a. Suppose $S_i(a) = N_i(a, b_k) = x$ and $S_i(b_k) = N_i(b_k, a) = y$.

If $x = y$, then each $N_i(b_k, b_r) = x$. In stage $i+1$, one vertex (w.l.o.g., a) will have a score of $x+1$, and the rest have a score of x. This is the same situation as before (multiple 2-cycles with a), but with $x+1 \neq x$. Now we will show that if $x \neq y$ we can eliminate 2-cycles in 2 stages (so if $x = y$, we need a total of 3 stages).

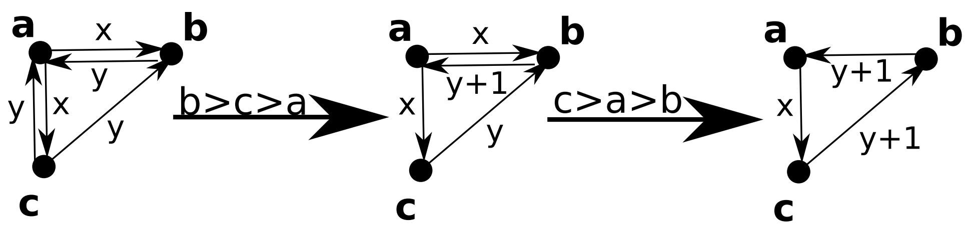


Figure 1: Dismantling multiple 2-cycles

Between each b_r, b_k there is at least one edge with value y. At stage $i+1$, $S_{i+1}(a) = x$ (as it is the Condorcet winner), some b's will have a value of $y+1$, while some will have a value of y. Of those with the value y, there will be at least one, b_ℓ, for which $N_{i+1}(a, b_\ell) = x$. This is because some b_k will be selected before a, either because $d_{out}(b_k) = 0$ or because it was according to lines 21–22 of the algorithm, which ensure that at least one b_k score will not change, and it will be selected before a according to line 17. In stage $i+2$, $S_{i+2}(a) = x$, since either b_ℓ will be selected before a, or if not, this means a was selected when $d_{out}(a) = 0$, which occurs when a's value doesn't change. Furthermore, if $S_{i+1}(b_r) = y+1$, and b_r was selected before b_ℓ, this means it happened due to $d_{out}(b_r) = 0$, and thus $S_{i+2}(b_r) = y+1$, and if it was selected after b_ℓ, since a's d_{out} after b_ℓ's selection is 0, if b_r is selected it is either before a, when its $d_{out} = 0$, or after a, and thus $S_{i+2}(b_r) = y+1$. Since $S_{i+2}(a) = x$, and there is no $c \in C$ such that $S_{i+2} = y+2$, this means there are no 2-cycles in G^{i+2}. ∎

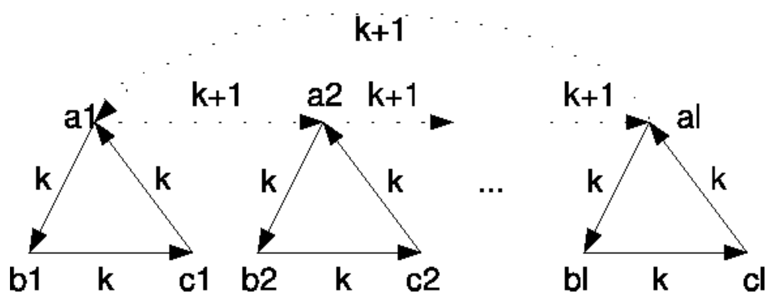


Figure 2: Example

Thus, if there is more than one 2-cycle in G^0, it will be eliminated in 3 steps at the most. Now, we wish to show that the algorithm eliminates 2-cycles as fast as possible, and then maintains its $1\frac{1}{2}$-approximation. Once the 2-cycles are eliminated the ms is the same (or less) than the optimal algorithm, and from that step onward our algorithm gives a $1\frac{1}{2}$-approximation.

Lemma 5.10. *Let b be a's partner in the 2-cycle in stage G^i. If in G^i there is $c \in C \neq a, b$ such that $N_i(a,c) = S_i(a)$, there will be no 2-cycle in G^{i+3}.*

Lemma 5.11. *If $c \in C$ was not a part of a 2-cycle in G^i, and is a part of a 2-cycle in G^{i+1}, there will be no 2-cycles in G^{i+4}.*

Therefore, we can assume that a and b were part of a 2-cycle in G^0, and they will be the only participants of a 2-cycle during the algorithm's run.

To continue, we need a few definitions. We will define $c \in C$ as $c \in \{x \in C | x \in \min_{y \in C \setminus a,b} N_0(b,y)\}$. We will define $d \in C$ as $d \in \{x \in C | x \in \min_{y \in C \setminus a,b} N_0(a,y)\}$.

Lemma 5.12. *If $N_0(b,c) < S_0(a)$, then after $N_0(b,c) - S_0(b) + 1$ steps there are no 2-cycles, and $ms(N_0(b,c) - S_0(b) + 1) = ms(0)$.*

Lemma 5.13. *Let $h = min(N_0(b,c), N_0(a,d))$. For any algorithm, if $h \geq S_0(a)$, $ms^*(S_0(a) - S_0(b) + 2(h - S_0(a))) \geq h$.*

Lemma 5.14. *Let $h = min(N_0(b,c), N_0(a,d))$. Using Algorithm 1, if $h \geq S_0(a)$, $ms(S_0(a) - S_0(b) + 2(h - S_0(a))) = h$, and there are no 2-cycles in $G^{S_0(a)+2(h-S_0(a))+1}$.*

We have shown that if there are multiple 2-cycles in G^0, we end up with no 2-cycles in G^3. If there is one, it is abolished, and the ms at the state in which it is abolished is the smallest possible. From that point on, our algorithm provides a $1\frac{1}{2}$-approximation (according to Theorem 5.1). $\square$

Now we show that our analysis of Algorithm 1 is accurate.

Theorem 5.15. *The $1\frac{1}{2}$ approximation ratio of Algorithm 1 is asymptotically tight.*

Proof. Consider the following example (see Figure 2). $C = \{p, a_1, b_1, c_1, a_2, b_2, c_2, \ldots, a_l, b_l, c_l\}$. Let k be an integer, $\frac{N}{3} \leq k < \frac{N}{2}$. $S_0(p) = 0$; for all $j, 1 \leq j \leq l$: $S_0(a_j) = N_0(a_j, b_j) = S_0(b_j) = N_0(b_j, c_j) = S_0(c_j) = N_0(c_j, a_j) = k$. In addition, for each $j, 1 \leq j \leq l-1$: $N_0(a_j, a_{j+1}) = k+1$, and $N_0(a_l, a_1) = k+1$. When showing the preferences of the manipulators, we denote by A_j the fragment $a_j \succ c_j \succ b_j$

of the preference, by B_j the fragment $b_j \succ a_j \succ c_j$, and by C_j the fragment $c_j \succ b_j \succ a_j$. Consider the following preference list of the manipulators:

$$\begin{aligned}
&p \succ A_l \succ A_{l-1} \succ \ldots \succ A_1 \\
&p \succ A_{l-1} \succ A_{l-2} \succ \ldots \succ A_1 \succ A_l \\
&p \succ A_{l-2} \succ A_{l-3} \succ \ldots \succ A_1 \succ A_l \succ A_{l-1} \\
&\ldots
\end{aligned}$$

It can be verified that in the above preference list, the maximum score of p's opponents ($\mathrm{ms}(i)$) grows by 1 every $\frac{m-1}{3}$ stages (starting with $k+1$). In addition, p's score grows by 1 every stage. Therefore, when we apply the voting above, the minimum number of stages (manipulators) n^* needed to make p win the election should satisfy $n^* > k + 1 + \lceil \frac{3n^*}{m-1} \rceil$. Since $\lceil \frac{3n^*}{m-1} \rceil < \frac{3n^*}{m-1} + 1$, the sufficient condition for making p win is:

$$n^* > k + 1 + \frac{3n^*}{m-1} + 1.$$

So, we have,

$$\begin{aligned}
(m-1)n^* &> (m-1)(k+2) + 3n^* \\
(m-4)n^* &> (m-1)(k+2) \\
n^* &> \frac{(m-1)(k+2)}{m-4}.
\end{aligned}$$

For large-enough m, $\frac{(m-1)(k+2)}{m-4} < k+3$, and so $n^* = k+3$ would be enough to make p win the election.

Now let us examine what Algorithm 1 will do when it gets this example as input. One of the possible outputs of the algorithm looks like this:

$$\begin{aligned}
&p \succ C_1 \succ C_2 \succ \ldots \succ C_l \\
&p \succ B_2 \succ B_3 \succ \ldots \succ B_l \succ B_1 \\
&p \succ A_3 \succ A_4 \succ \ldots \succ A_l \succ A_1 \succ A_2 \\
&p \succ C_4 \succ C_5 \succ \ldots \succ C_l \succ C_1 \succ C_2 \succ C_3 \\
&\ldots
\end{aligned}$$

It can be verified that in the above preference list, $\mathrm{ms}(i)$ grows by 1 every 3 stages, and p's score grows by 1 every stage. Therefore, the number of stages n returned by Algorithm 1 that are needed to make p win the election satisfies $n > k + \lceil \frac{n}{3} \rceil$. Since $\lceil \frac{n}{3} \rceil \geq \frac{n}{3}$, the necessary condition for making p win the election is:

$$n > k + \frac{n}{3}.$$

We then have,

$$\begin{aligned}
3n &> 3k + n \\
2n &> 3k \\
n &> \frac{3}{2}k.
\end{aligned}$$

So we find that the ratio $\frac{n}{n^*}$ tends to $1\frac{1}{2}$ as m and N (and k) tend to infinity. $\square$

6 Conclusions and Future Work

We introduced a new algorithm for approximating the UCO problem under the maximin voting rule, and investigated its approximation guarantees. In future work, it would be interesting to prove or disprove that Algorithm 1 presented in [7] has an approximation ratio of $1\frac{1}{2}$, for those instances where there is no Condorcet winner.[2] Another issue is to implement both algorithms, to empirically measure their performance and compare them in practice.

Acknowledgments

We would like to thank Reshef Meir, Aviv Zohar, and Jerome Lang for helpful discussions on the topics of this work. This work was partially supported by Israel Science Foundation grant #898/05, and Israel Ministry of Science and Technology grant #3-6797.

References

[1] J. Bartholdi and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8:341–354, 1991.

[2] J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.

[3] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007.

[4] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[5] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[6] Lirong Xia, Michael Zuckerman, Ariel D. Procaccia, Vincent Conitzer, and Jeffrey S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *The Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 348–353, Pasadena, California, July 2009.

[7] Michael Zuckerman, Ariel D. Procaccia, and Jeffrey S. Rosenschein. Algorithms for the coalitional manipulation problem. *Journal of Artificial Intelligence*, 173(2):392–412, February 2009.

Michael Zuckerman, Omer Lev, Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem 91904, Israel
Email: `michez@cs.huji.ac.il`
Email: `omerl@cs.huji.ac.il`
Email: `jeff@cs.huji.ac.il`

[2]We have an example showing that that algorithm is no better than a 2-approximation when there is a Condorcet winner.

Algorithm 1 Decides CCUM for maximin voting rule

procedure MAXIMIN(C, p, X_S, n) ▷ X_S is the set of preferences of voters in S, n is the number of voters in T
  $X \leftarrow \emptyset$ ▷ Will contain the preferences of T
  for $i = 1, \dots, n$ **do** ▷ Iterate over voters
    $P_i \leftarrow (p)$ ▷ Put p at the first place of the i-th preference list
    Build a digraph $G^{i-1} = (V, E^{i-1})$ ▷ $V = C \setminus \{p\}$, $(x, y) \in E^{i-1}$ iff ($y \in \mathrm{MIN}_{i-1}(x)$ and $p \notin \mathrm{MIN}_{i-1}(x)$)
    for $c \in C \setminus \{p\}$ **do** ▷ This for loop is used in the analysis
      if $d_{out}(c) = 0$ **then**
        $c.father \leftarrow p$
      else
        $c.father \leftarrow null$
      end if
    end for
    while $C \setminus P_i \neq \emptyset$ **do** ▷ while there are candidates to be added to i-th preference list
      Evaluate the score of each candidate based on the votes of S and $i-1$ first votes of T
      if there exists a set $A \subseteq C \setminus P_i$ with $d_{out}(a) = 0$ for each $a \in A$ **then** ▷ if there exist vertices in the digraph G^{i-1} with out-degree 0
        Add the candidates of A to the stacks Q_j, where to the same stack go candidates with the same score
        $b \leftarrow Q_1.popfront()$ ▷ Retrieve the top-most candidate from the first stack—with the lowest scores so far
        $P_i \leftarrow P_i + \{b\}$ ▷ Add b to i's preference list
      else
        Let $s = \min_{c \in C \setminus P_i}\{S_{i-1}(c)\}$
        if there is a cycle U in G^{i-1} s.t. there are 3 vertices a, b, c, s.t. $(c, b), (b, a) \in U$, and $S_{i-1}(c) = S_{i-1}(b) = s$ **then**
          $P_i \leftarrow P_i + \{b\}$ ▷ Add b to i's preference list
        else
          Pick $b \in C \setminus P_i$ s.t. $S_{i-1}(b) = s$ ▷ Pick any candidate with the lowest score so far
          $P_i \leftarrow P_i + \{b\}$ ▷ Add b to i's preference list
        end if
      end if
      for $y \in C \setminus P_i$ **do**
        if $(y, b) \in E^{i-1}$ **then** ▷ If there is a directed edge from y to b in the digraph
          Remove all the edges of E^{i-1} originating in y
          $y.father \leftarrow b$ ▷ This statement is used in algorithm analysis
          Add y to the front of the appropriate stack Q_j—according to $S_{i-1}(y)$
        end if
      end for
    end while
    $X \leftarrow X \cup \{P_i\}$
  end for
  $X_T \leftarrow X$
  if $\mathrm{argmax}_{c \in C}\{\text{Score of } c \text{ based on } X_S \cup X_T\} = \{p\}$ **then**
    return true ▷ p wins
  else
    return false
  end if
end procedure

Complexity of Safe Strategic Voting[1]

Noam Hazon and Edith Elkind

Abstract

We investigate the computational aspects of *safe manipulation*, a new model of coalitional manipulation that was recently put forward by Slinko and White [11]. In this model, a potential manipulator v announces how he intends to vote, and some of the other voters whose preferences coincide with those of v may follow suit. Depending on the number of followers, the outcome could be better or worse for v than the outcome of truthful voting. A manipulative vote is called *safe* if for some number of followers it improves the outcome from v's perspective, and can never lead to a worse outcome. In this paper, we study the complexity of finding a safe manipulative vote for a number of common voting rules, including Plurality, Borda, k-approval, and Bucklin, providing algorithms and hardness results for both weighted and unweighted voters. We also propose two ways to extend the notion of safe manipulation to the setting where the followers' preferences may differ from those of the leader, and study the computational properties of the resulting extensions.

1 Introduction

Computational aspects of voting, and, in particular, voting manipulation, is an active topic of current research. While the complexity of the manipulation problem for a single voter is quite well understood (specifically, this problem is known to be efficiently solvable for most common voting rules with the notable exception of STV [1, 2]), the more recent work has mostly focused on coalitional manipulation, i.e., manipulation by multiple, possibly weighted voters. In contrast to the single-voter case, coalitional manipulation tends to be hard. Indeed, it has been shown to be NP-hard for weighted voters even when the number of candidates is bounded by a small constant [4]. For unweighted voters, nailing the complexity of coalitional manipulation proved to be more challenging. However, Faliszewski et al. [5] have recently established that this problem is hard for most variants of Copeland, and Zuckerman *et al* [13] showed that it is easy for Veto and Plurality with Runoff. Further, a very recent paper [12] makes substantial progress in this direction, showing, for example, that unweighted coalitional manipulation is hard for Maximin and Ranked Pairs, but easy for Bucklin (see Section 2 for the definitions of these rules).

All of these papers (as well as the classic work of Bartholdi et al. [1]) assume that the set of manipulators is given exogenously, and the manipulators are not endowed with preferences over the entire set of candidates; rather, they simply want to get a particular candidate elected, and select their votes based on the non-manipulators' preferences that are publicly known. That is, this model abstracts away the question of how the manipulating coalition forms. However, to develop a better understanding of coalitional manipulation, it is desirable to have a plausible model of the coalition formation process. In such a model the manipulators would start out by having the same type of preferences as sincere voters, and then some agents—those who are not satisfied with the current outcome and are willing to submit an insincere ballot—would get together and decide to coordinate their efforts.

However, it is quite difficult to formalize this intuition so as to obtain a realistic model of how the manipulating coalition forms. In particular, it is not clear how the voters who are interested in manipulation should identify each other, and then reach an agreement which candidate to promote. Indeed, the latter decision seems to call for a voting procedure, and therefore is itself vulnerable to strategic behavior. Further, even assuming that suitable coalition formation and decision-making

[1] To appear in the proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT'10)

procedures exist, their practical implementation may be hindered by the absence of reliable two-way communication among the manipulators.

In a recent paper [11], Slinko and White put forward a model that provides a partial answer to these questions. They consider a setting where a single voter v announces his manipulative vote L (the truthful preferences of all agents are, as usual, common knowledge) to his set of associates F, i.e., the voters whose true preferences coincide with those of v. As a result, some of the voters in F switch to voting L, while others (as well as all voters not in F) vote truthfully. This can happen if, e.g., v's instructions are broadcast via an unreliable channel, i.e., some of the voters in F simply do not receive the announcement, or if some voters in F consider it unethical to vote non-truthfully. Such a situation is not unusual in politics, where a public figure may announce her decision to vote in a particular manner, and may be followed by a subset of like-minded people. That is, in this model, the manipulating coalition always consists of voters with identical preferences (and thus the problem of which candidate to promote is trivially resolved), and, moreover, the manipulators always vote in the same way. Further, it relies on minimal communication, i.e., a single broadcast message. However, due to lack of two-way communication, v does not know how many voters will support him in his decision to vote L. Thus, he faces a dilemma: it might be the case that if x voters from F follow him, then the outcome improves, while if some $y \neq x$ voters from F switch to voting L, the outcome becomes even less desirable to v than the current alternative (we provide an example in Section 2). If v is conservatively-minded, in such situations he would choose not to manipulate at all. In other words, he would view L as a successful manipulation only if (1) there exists a subset $U \subseteq F$ such that if the voters in U switch to voting L, the outcome improves; (2) for any $W \subseteq F$, if the voters in W switch to voting L the outcome does not get worse. Paper [11] calls any manipulation that satisfies (1) and (2) *safe*. The main result of [11] is a generalization of the Gibbard–Satterthwaite theorem [7, 10] to safe manipulation: the authors prove that any onto, non-dictatorial voting rule with at least 3 alternatives is safely manipulable, i.e., there exists a profile in which at least one voter has a safe manipulation. However, paper [11] does not explore the computational complexity of the related problems.

In the first part of this paper, we focus on algorithmic complexity of safe manipulation, as defined in [11]. We first formalize the relevant computational questions and discuss some basic relationships between them. We then study the complexity of these questions for several classic voting rules, such as Plurality, Veto, k-approval, Bucklin, and Borda, for both weighted and unweighted voters. For instance, we show that finding a safe manipulation is easy for k-approval and for Bucklin, even if the voters are weighted. In contrast, for Borda, finding a safe manipulation—or even checking that a given vote is safe—turns out to be hard for weighted voters even if the number of candidates is bounded by a small constant.

We then explore whether it is possible to extend the model of safe manipulation to settings where the manipulator may be joined by voters whose preferences differ from his own. Indeed, in real life a voter may follow advice to vote in a certain way if it comes from a person whose preferences are similar (rather than identical) to hers, or simply because she thinks that voting in this manner can be beneficial to her. For instance, in politics, a popular personality may influence many different voters at once by announcing his decision to vote in a particular manner. We propose two ways of formalizing this idea, which differ in their approach to defining the set of a voter's potential followers, and provide initial results on the complexity of safe manipulation in these models.

In our first extension, a manipulator v may be followed by all voters who rank the same candidates above the current winner as v does. That is, in this model a voter u may follow v if any change of outcome that is beneficial to v is also beneficial to u. We show that some of the positive algorithmic results for the standard model also hold in this more general setting. In our second model, a voter u may follow a manipulator v that proposes to vote L, if, roughly, there are circumstances when voting L is beneficial to u. This model tends to be computationally more challenging: we show that finding a safe strategic vote in this setting is hard even for very simple voting rules.

We conclude the paper by summarizing our results and proposing several directions for future research.

2 Preliminaries and Notation

An *election* is given by a set of *candidates* (also referred to as *alternatives*) $C = \{c_1, \ldots, c_m\}$ and a set of *voters* $V = \{1, \ldots, n\}$. Each voter i is represented by his *preference* R_i, which is a total order over C; we will also refer to total orders over C as *votes*. For readability, we will sometimes denote the order R_i by $\succ_i$. The vector $\mathcal{R} = (R_1, \ldots, R_n)$ is called a *preference profile*. We say that two voters i and j are of the same *type* if $R_i = R_j$; we write $V_i = \{j \mid R_j = R_i\}$.

A *voting rule* $\mathcal{F}$ is a mapping from the set of all preference profiles to the set of candidates; if $\mathcal{F}(\mathcal{R}) = c$, we say that c *wins* under $\mathcal{F}$ in $\mathcal{R}$. A voting rule is said to be *anonymous* if $\mathcal{F}(\mathcal{R}) = \mathcal{F}(\mathcal{R}')$, where $\mathcal{R}'$ is a preference profile obtained by permuting the entries of $\mathcal{R}$. To simplify the presentation, in this paper we consider anonymous voting rules only. In addition, we restrict ourselves to voting rules that are polynomial-time computable.

During the election, each voter i submits a vote L_i; the outcome of the election is then given by $\mathcal{F}(L_1, \ldots, L_n)$. We say that a voter i is *truthful* if $L_i = R_i$. For any $U \subseteq V$ and a vote L, we denote by $\mathcal{R}_{-U}(L)$ the profile obtained from $\mathcal{R}$ by replacing R_i with L for all $i \in U$.

Voting rules We will now define the voting rules considered in this paper. All of these rules assign scores to all candidates; the winner is then selected among the candidates with the highest score using a *tie-breaking rule*, i.e., a mapping $T : 2^C \to C$ that satisfies $T(S) \in S$. Unless specified otherwise, we assume that the tie-breaking rule is *lexicographic*, i.e., given a set of tied alternatives, it selects one that is maximal with respect to a fixed ordering $\succ$.

Given a vector $\alpha = (\alpha_1, \ldots, \alpha_m)$ with $\alpha_1 \geq \cdots \geq \alpha_m$, the *score* $s_\alpha(c)$ of a candidate $c \in C$ under a *positional scoring rule* F_α is given by $\sum_{i \in V} \alpha_{j(i,c)}$, where $j(i, c)$ is the position in which voter i ranks candidate c. Many classic voting rules can be represented using this framework. Indeed, *Plurality* is the scoring rule with $\alpha = (1, 0, \ldots, 0)$, *Veto* (also known as *Antiplurality*) is the scoring rule with $\alpha = (1, \ldots, 1, 0)$, and *Borda* is the scoring rule with $\alpha = (m-1, m-2, \ldots, 1, 0)$. Further, k-*approval* is the scoring rule with α given by $\alpha_1 = \cdots = \alpha_k = 1$, $\alpha_{k+1} = \cdots = \alpha_m = 0$; we will also refer to $(m-k)$-approval as k-*veto*.

Bucklin rule can be viewed as an adaptive version of k-approval. We say that k, $1 \leq k \leq m$, is the *Bucklin winning round* if for any $j < k$ no candidate is ranked in top j positions by at least $\lceil n/2 \rceil$ voters, and there exists some candidate that is ranked in top k positions by at least $\lceil n/2 \rceil$ voters. We say that the candidate c's *score in round* j is his j-approval score, and his *Bucklin score* $s_B(c)$ is his k-approval score, where k is the Bucklin winning round. The *Bucklin winner* is the candidate with the highest Bucklin score. Observe that the Bucklin score of the Bucklin winner is at least $\lceil n/2 \rceil$.

Weighted voters Our model can be extended to the situation where not all voters are equally important by assigning an integer *weight* w_i to each voter i. To compute the winner on a profile $(R_1, \ldots, R_n)$ under a voting rule $\mathcal{F}$ given voters' weights $\mathbf{w} = (w_1, \ldots, w_n)$, we apply $\mathcal{F}$ on a modified profile which for each $i = 1, \ldots, n$ contains w_i copies of R_i. As an input to our problems we usually get a *voting domain*, i.e., a tuple $S = \langle C, V, \mathbf{w}, \mathcal{R} \rangle$, together with a specific voting rule. When $\mathbf{w} = (1, \ldots, 1)$, we say that the voters are *unweighted*. For each $U \subseteq V$, let $|U|$ be the number of voters in U and let $w(U)$ be the total weight of the voters in U.

Safe manipulation We will now formally define the notion of safe manipulation. For the purposes of our presentation, we can simplify the definitions in [11] considerably.

As before, we assume that the voters' true preferences are given by a preference profile $\mathcal{R} = (R_1, \ldots, R_n)$.

Definition 1. We say that a vote L is an *incentive to vote strategically*, or a *strategic vote* for i at $\mathcal{R}$ under $\mathcal{F}$, if $L \neq R_i$ and for some $U \subseteq V_i$ we have $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_i \mathcal{F}(\mathcal{R})$. Further, we say that L

is a *safe strategic vote* for a voter i at $\mathcal{R}$ under $\mathcal{F}$ if L is a strategic vote at $\mathcal{R}$, and for any $U \subseteq V_i$ either $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_i \mathcal{F}(\mathcal{R})$ or $\mathcal{F}(\mathcal{R}_{-U}(L)) = \mathcal{F}(\mathcal{R})$.

To build intuition for the notions defined above, consider the following example.

Example 1. Suppose $C = \{a, b, c, d\}$, $V = \{1, 2, 3, 4\}$, the first three voters have preference $b \succ a \succ c \succ d$, and the last voter has preference $c \succ d \succ a \succ b$. Suppose also that the voting rule is 2-approval. Under truthful voting, a and b get 3 points, and c and d get 1 point each. Since ties are broken lexicographically, a wins. Now, if voter 1 changes his vote to $L = b \succ c \succ a \succ d$, b gets 3 points, a gets 2 points, and c gets 2 points, so b wins. As $b \succ_1 a$, L is a strategic vote for 1. However, it is not a safe strategic vote: if players in $V_1 = \{1, 2, 3\}$ all switch to voting L, then c gets 4 points, while b still gets 3 points, so in this case c wins and $a \succ_1 c$.

A *maximal* manipulation is one where all the voters from V_i choose to vote L. We will call the winner of such profile the *maximal manipulation winner* for L.

3 Computational Problems: First Observations

The definition of safe strategic voting gives rise to two natural algorithmic questions. In the definitions below, $\mathcal{F}$ is a given voting rule and the voters are assumed to be unweighted.

- ISSAFE($\mathcal{F}$): Given a voting domain, a voter i and a linear order L, is L a safe strategic vote for i under $\mathcal{F}$?
- EXISTSAFE($\mathcal{F}$): Given a voting domain and a voter i, can voter i make a safe strategic vote under $\mathcal{F}$?

The variants of these problems for weighted voters will be denoted, respectively, by WISSAFE($\mathcal{F}$) and WEXISTSAFE($\mathcal{F}$). Note that, in general, it is not clear if an efficient algorithm for (W)EXISTSAFE($\mathcal{F}$) can be used to solve (W)ISSAFE($\mathcal{F}$), or vice versa. However, if the number of candidates is constant, (W)EXISTSAFE($\mathcal{F}$) reduces to (W)ISSAFE($\mathcal{F}$). We formulate the following two results for weighted voters; clearly, they also apply to unweighted voters.

Proposition 1. *Consider any voting rule $\mathcal{F}$. For any constant k, if $|C| \leq k$, then a polynomial-time algorithm for* WISSAFE($\mathcal{F}$) *can be used to solve* WEXISTSAFE($\mathcal{F}$) *in polynomial time.*

Proof. In this case i has at most $k! = O(1)$ different votes, so he can try all of them. ☐

A similar reduction exists when each voter only has polynomially many "essentially different" votes.

Proposition 2. *Consider any scoring rule $\mathcal{F}_\alpha$ that satisfies either (i) $\alpha_j = 0$ for all $j > k$ or (ii) $\alpha_j = 1$ for all $j \leq m - k$, where k is a given constant. For any such rule, a polynomial-time algorithm for* WISSAFE($\mathcal{F}_\alpha$) *can be used to solve* WEXISTSAFE($\mathcal{F}_\alpha$) *in polynomial time.*

Proof. We consider case (i); case (ii) is similar. There are at most $n^k = \mathrm{poly}(n)$ different ways to fill the top k positions in a vote. Further, if two votes only differ in positions $k + 1, \ldots, m$, they result in the same outcome. Thus, to solve WEXISTSAFE($\mathcal{F}_\alpha$), it suffices to run WISSAFE($\mathcal{F}_\alpha$) on $\mathrm{poly}(n)$ instances. ☐

Observe that the class of rules considered in Proposition 2 includes Plurality and Veto, as well as k-approval and k-veto when k is bounded by a constant.

Further, we note that for unweighted voters it is easy to check if a given manipulation is safe.

Proposition 3. *The problem* ISSAFE($\mathcal{F}$) *is in* P *for any (anonymous) voting rule $\mathcal{F}$.*

Proof. Set $V_i = \{i_1, \ldots, i_s\}$. Since our voting rule is anonymous, it suffices to check the conditions of Definition 1 for $U \in \{\{i_1\}, \{i_1, i_2\}, \ldots, \{i_1, \ldots, i_s\}\}$, i.e., for $s \leq n$ sets of voters. $\square$

Together with Propositions 1 and 2, Proposition 3 implies that the problem EXISTSAFE($\mathcal{F}$) is in P for Plurality, Veto, k-veto and k-approval for constant k, as well as for any voting rule with a constant number of candidates.

Note that when voters are weighted, the conclusion of Proposition 3 no longer holds. Indeed, in this case the number of subsets of V_i that have different weights (and thus may have a different effect on the outcome) may be exponential in n. However, it is not hard to show that the problem remains easy when all weights are small (polynomially bounded).

4 Plurality, veto, and k-approval

We will now show that the easiness results for k-approval and k-veto extend to arbitrary $k \leq m$ and weighted voters (note that the distinction between k-veto and $(m-k)$-approval only matters for constant k).

Theorem 4. *For k-approval, the problems* wISSAFE *and* wEXISTSAFE *are in* P.

Proof. Fix a voter $v \in V$. To simplify notation, we renumber the candidates so that v's preference order is given by $c_1 \succ_v \ldots \succ_v c_m$. Denote v's truthful vote by R. Recall that V_v denotes the set of voters who have the same preferences as v. Suppose that under truthful voting the winner is c_j. For $i = 1, \ldots, m$, let $s_i(\mathcal{R}')$ denote the k-approval score of c_i given a profile $\mathcal{R}'$, and set $s_i = s_i(\mathcal{R})$.

We start by proving a useful characterization of safe strategic votes for k-approval.

Lemma 1. *A vote L is a safe strategic vote for v if and only if the winner in $\mathcal{R}_{-V_v}(L)$ is a candidate c_i with $i < j$.*

Proof. Suppose that L is a safe strategic vote for v. Then there exists an $i < j$ and a $U \subseteq V_v$ such that the winner in $\mathcal{R}_{-U}(L)$ is c_i. It must be the case that each switch from R to L increases c_i's score or decreases c_j's score: otherwise c_i cannot beat c_j after the voters in U change their vote from R to L. Therefore, if c_i beats c_j when the preference profile is $\mathcal{R}_{-U}(L)$, it continues to beat c_j after the remaining voters in V_v switch, i.e., when the preference profile is $\mathcal{R}_{-V_v}(L)$. Hence, the winner in $\mathcal{R}_{-V_v}(L)$ is not c_j; since L is safe, this means that the winner in $\mathcal{R}_{-V_v}(L)$ is c_ℓ for some $\ell < j$.

For the opposite direction, suppose that the winner in $\mathcal{R}_{-V_v}(L)$ is c_i for some $i < j$. Note that if two candidates gain points when some subset of voters switches from R to L, they both gain the same number of points; the same holds if both of them lose points.

Now, if $j > k$, a switch from R to L does not lower the score of c_j, so it must increase the score of c_i for it to be the maximal manipulation winner. Further, if a switch from R to L grants points to some $c_\ell \neq c_i$, then either $s_\ell < s_i$ or $s_\ell = s_i$ and the tie-breaking rule favors c_i over c_ℓ: otherwise, c_i would not be the maximal manipulation winner.

Similarly, if $j \leq k$, a switch from R to L does not increase the score of c_i, so it must lower the score of c_j. Further, if some $c_\ell \neq c_i$ does not lose points from a switch from R to L, then either $s_\ell < s_i$ or $s_\ell = s_i$ and the tie-breaking rule favors c_i over c_ℓ: otherwise, c_i would not be the maximal manipulation winner.

Now, consider any $U \subseteq V_v$. If $s_j(\mathcal{R}_{-U}(L)) > s_i(\mathcal{R}_{-U}(L))$, then c_j is the winner. If $s_i(\mathcal{R}_{-U}(L)) > s_j(\mathcal{R}_{-U}(L))$, then c_i is the winner. Finally, suppose $s_i(\mathcal{R}_{-U}(L)) = s_j(\mathcal{R}_{-U}(L))$. By the argument above, no other candidate can have a higher score. So, suppose that $s_\ell(\mathcal{R}_{-U}(L)) = s_i(\mathcal{R}_{-U}(L))$, and the tie-breaking rule favors c_ℓ over c_i and c_j. Then this would imply that c_ℓ wins in $\mathcal{R}$ or $\mathcal{R}_{-V_v}(L)$ (depending on whether a switch from R to L causes c_ℓ to lose points), a contradiction. Thus, in this case, too, either c_i or c_j wins. $\square$

Lemma 1 immediately implies an algorithm for wISSAFE: we simply need to check that the input vote satisfies the conditions of the lemma. We now show how to use it to construct an algorithm for wEXISTSAFE. We need to consider two cases.

$\mathbf{j > k}$**:**
In this case, the voters in V_v already do not approve of c_j and approve of all c_i, $i \leq k$. Thus, no matter how they vote, they cannot ensure that some c_i, $i \leq k$, gets more points than c_j. Hence, the only way they can change the outcome is by approving of some candidate c_i, $k < i < j$. Further, they can only succeed if there exists an $i = k+1, \dots, j-1$ such that either $s_i + w(V_v) > s_j$ or $s_i + w(V_v) = s_j$ and the tie-breaking rule favors c_i over c_j. If such an i exists, v has an incentive to manipulate by swapping c_1 and c_i in his vote. Furthermore, it is easy to see that any such manipulation is safe, as it only affects the scores of c_1 and c_i.

$\mathbf{j \leq k}$**:**
In this case, the voters in V_v already approve of all candidates they prefer to c_j, and therefore they cannot increase the scores of the first $j-1$ candidates. Thus, their only option is to try to lower the scores of c_j as well as those of all other candidates whose score currently matches or exceeds the best score among $s_1, \dots, s_{j-1}$.

Set $C_g = \{c_1, \dots, c_{j-1}\}$, $C_b = \{c_j, \dots, c_m\}$. Let C_0 be the set of all candidates in C_g whose k-approval score is maximal, and let $s_{\max}$ be the k-approval score of the candidates in C_0. For any $c_\ell \in C_b$, let s'_ℓ denote the number of points that c_ℓ gets from all voters in $V \setminus V_v$; we have $s'_\ell = s_\ell$ for $k < \ell \leq m$ and $s'_\ell = s_\ell - w(V_v)$ for $\ell = j, \dots, k$. Now, it is easy to see that v has a safe manipulation if and only if the following conditions hold:

- For all $c_\ell \in C_b$ either $s'_\ell < s_{\max}$, or $s'_\ell = s_{\max}$ and there exists a candidate $c \in C_0$ such that the tie-breaking rule favors c over c_ℓ;
- There exist a set $C_{\text{safe}} \subseteq C_b$, $|C_{\text{safe}}| = k - j + 1$, such that for all $c_\ell \in C_{\text{safe}}$ either $s'_\ell + w(V_v) < s_{\max}$ or $s'_\ell + w(V_v) = s_{\max}$ and there exists a candidate $c \in C_0$ such that the tie-breaking rule favors c over c_ℓ.

Note that these conditions can be easily checked in polynomial time by computing s_ℓ and s'_ℓ for all $\ell = 1, \dots, m$.

Indeed, if such a set C_{safe} exists, voter v can place the candidates in C_{safe} in positions $j, \dots, k$ in his vote; denote the resulting vote by L. Clearly, if all voters in V_v vote according to L, they succeed to elect some $c \in C_0$. Thus, by Lemma 1, L is safe. Conversely, if a set C_{safe} with these properties does not exist, then for any vote $L \neq R$ the winner in $\mathcal{R}_{-V_v}(L)$ is a candidate in C_b, and thus by Lemma 1 L is not safe. □

We remark that Theorem 4 crucially relies on the fact that we break ties based on a fixed priority ordering over the candidates. Indeed, it can be shown that there exists a (non-lexicographic) tie-breaking rule such that finding a safe vote with respect to k-approval combined with this tie-breaking rule is computationally hard (assuming k is viewed as a part of the input). As the focus of this paper is on lexicographic tie-breaking, we omit the formal statement and the proof of this fact.

In contrast, we can show that any scoring rule with 3 candidates is easy to manipulate safely, even if the voters are weighted and arbitrary tie-breaking rules are allowed.

Theorem 5. wISSAFE($\mathcal{F}$) *is in* P *for any voting rule* $\mathcal{F}$ *obtained by combining a positional scoring rule with at most three candidates with an arbitrary tie-breaking rule.*

Proof. For one candidate, the statement is trivial. With two candidates, every positional scoring rule is equivalent to Plurality, and under Plurality with two candidates no voter has an incentive to vote strategically.

Now, suppose that $|C| = 3$. Consider a voter i and assume without loss of generality that $R_i = (c_1, c_2, c_3)$. If $\mathcal{F}(\mathcal{R}) = c_1$, then i has no incentive to vote strategically. We will now consider the cases $\mathcal{F}(\mathcal{R}) = c_2$ and $\mathcal{F}(\mathcal{R}) = c_3$ separately.

1. $\mathcal{F}(\mathcal{R}) = c_2$. Suppose that L is a strategic vote for i. Then L cannot rank c_2 in top two positions. Indeed, any such manipulation does not decrease c_2's score and does not increase c_1's score. Thus, if c_2 had a higher score than c_1, this would still be the case no matter how many voters in V_i switch to voting L. Further, if both c_2 and c_1 had top scores, then L could succeed only if it does not change the scores of either of them. But in this case the score of c_3 does not change either, so the outcome remains the same. Thus, it remains to consider two cases: $L = (c_1, c_3, c_2)$ and $L = (c_3, c_1, c_2)$. Now, let $c = \mathcal{F}(\mathcal{R}_{-V_i}(L))$. If $c = c_3$, L is not safe. Further, if $c = c_2$, then we have $c_2 = \mathcal{F}(\mathcal{R}_{-U}(L))$ for any $U \subseteq V_i$, i.e., L is not a strategic vote for i. Finally, if $c = c_1$, then L is a safe strategic vote. Indeed, suppose that L is not safe, i.e., $\mathcal{F}(\mathcal{R}_{-U}(L)) = c_3$ for some $U \subset V_i$. Each switch from R_i to L does not decrease c_3's score, so in that case c_3 would be a winner in $\mathcal{R}_{-V_i}(L)$, a contradiction.

2. $\mathcal{F}(\mathcal{R}) = c_3$. It can be checked that if L is a strategic vote for i, then L has to rank c_2 first, i.e., $L \in \{(c_2, c_1, c_3), (c_2, c_3, c_1)\}$. If $\mathcal{F}(\mathcal{R}_{-V_i}(L)) = c_3$, by the same argument as above, there is no incentive for i to vote for L. Otherwise, L is a safe strategic vote, since c_3 is the least preferred candidate.

□

5 Bucklin and Borda

Bucklin rule is quite similar to k-approval, so we can use the ideas in the proof of Theorem 4 to design a polynomial-time algorithm for finding a safe manipulation with respect to Bucklin. However, the proof becomes significantly more complicated.

Theorem 6. *For the Bucklin rule,* wEXISTSAFE *is in* P.

Interestingly, despite the intuition that wISSAFE should be easier than wEXISTSAFE, it turns out that wISSAFE for Bucklin is coNP-hard.

Theorem 7. *For the Bucklin rule,* wISSAFE *is* coNP*-hard, even for a constant number of candidates.*

Proof. We give a reduction from SUBSET SUM. Recall that an instance of SUBSET SUM is given by a set of positive integers $A = \{a_1, \dots, a_s\}$ and a positive integer K. It is a "yes"-instance if there is a subset of indices $I \subseteq \{1, \dots, s\}$ such that $\sum_{i \in I} a_i = K$ and a "no"-instance otherwise. We assume without loss of generality that $K < \sum_{a_i \in A} a_i$.

Given an instance (A, K) of SUBSET SUM with $|A| = s$ and $\sum_{i=1}^{s} a_i = S$, we construct an instance of wISSAFE as follows. Set $C = \{a, b, c, x, y, z, x', y', z'\}$, and let $V = \{v_1, \dots, v_s, u_1, u_2, u_3, u_4\}$. Table 1 shows the preferences and weights of each voter; observe that the total weight of all voters is $4S$. We ask if the vote $L = (a, c, b, x, y, z, x', y', z')$ is a safe strate-

Table 1:

Voter	Preference order	Weight
v_i	$(x, y, z, a, b, c, x', y', z')$	a_i
u_1	$(a, c, b, x, y, z, x', y', z')$	$2S - K - 1$
u_2	$(x, c, b, a, y, z, x', y', z')$	1
u_3	$(y, z, b, a, c, x, x', y', z')$	K
u_4	$(x', y', z', a, b, c, x, y, z)$	S

gic vote for v_1 under Bucklin; as we will see, the answer to this question does not depend on the tie-breaking rule.

If all voters vote sincerely, then b wins in round 3 with $2S$ points, and all other voters get less that $2S$ points in the first three rounds. Note also that the total weight of voters in $C \setminus V_{v_1}$ that rank a first is $2S - K - 1$, and the total weight of voters in $C \setminus V_{v_1}$ that rank c second is $2S - K$.

Suppose that a group of voters $U \subseteq V_{v_1}$ votes L. If $w(U) < K$, then b remains the winner, while if $w(U) > K$ then a becomes the winner, as it gets the majority of votes in the first round. Therefore, L is a strategic vote for v_1. However, if $w(U) = K$, a only gets $2S - 1$ points in any of the first three rounds, while c gets $2S$ points in the second round. Therefore, in this case c wins, i.e., L is not safe for v_1. Hence, L is a safe strategic vote for v_1 if and only if no subset of A sums to K. □

For Borda, unlike k-approval and Bucklin, both of our problems are hard when the voters are weighted. The proof of the following theorem is similar to that of Theorem 7.

Theorem 8. *For the Borda rule,* wISSAFE *and* wEXISTSAFE *are* coNP*-hard. The hardness result holds even if there are only* 5 *candidates.*

6 Extensions of the Safe Strategic Voting Model

So far, we followed the model of [11] and assumed that the only voters who may change their votes are the ones whose preferences exactly coincide with those of the manipulator. Clearly, in real life this assumption does not always hold. Indeed, a voter may follow a suggestion to vote in a certain way as long as it comes from someone he trusts (e.g., a well-respected public figure), and this does not require that this person's preferences are completely identical to those of the voter. For example, if both the original manipulator v and his would-be follower u rank the current winner last, it is easy to see that following v's recommendation that leads to displacing the current winner is in u's best interests.

In this section, we will consider two approaches to extending the notion of safe strategic voting to scenarios where not all manipulators have identical preferences. In both cases, we define the set of potential followers for each voter (in our second model, this set may depend on the vote suggested), and define a vote L to be safe if, whenever a subset of potential followers votes L, the outcome of the election does not get worse (and sometimes gets better) from the manipulator's perspective. However, our two models differ in the criteria they use to identify a voter's potential followers. Due to space constraints, all proofs in this section are omitted.

Preference-Based Extension Our first model identifies the followers of a given voter based on the similarities in voters' preferences.

Fix a preference profile $\mathcal{R}$ and a voting rule $\mathcal{F}$, and let c be the winner under truthful voting. For any $v \in V$, let $I(v, c)$ denote the set of candidates that v ranks strictly above c. We say that two voters u and v are *similar* if $I(u, c) = I(v, c)$. A *similar set* S_v of a voter v for a given preference profile $\mathcal{R}$ and a voting rule $\mathcal{F}$ is given by $S_v = \{u \mid I(u, c) = I(v, c)\}$. (The set S_v depends on $\mathcal{R}$ and $\mathcal{F}$; however, for readability we omit $\mathcal{R}$ and $\mathcal{F}$ from the notation).

Note that if u and v are similar, they rank c in the same position. Further, a change of outcome from c to another alternative is positive from u's perspective if and only if it is positive from v's perspective. Thus, intuitively, any manipulation that is profitable for u is also profitable for v. Observe also that similarity is an equivalence relation, and the sets S_v are the corresponding equivalence classes. In particular, this implies that for any $u, v \in V$ either $S_u = S_v$ or $S_u \cap S_v = \emptyset$.

We can now adapt Definition 1 to our setting by replacing V_v with S_v.

Definition 2. *A vote L is a* strategic vote in the preference-based extension *for v at $\mathcal{R}$ under $\mathcal{F}$ if for some $U \subseteq S_v$ we have $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_v \mathcal{F}(\mathcal{R})$. Further, we say that L is a* safe strategic vote in the preference-based extension *for a voter v at $\mathcal{R}$ under $\mathcal{F}$ if L is a strategic vote at $\mathcal{R}$ under $\mathcal{F}$, and for any $U \subseteq S_v$ either $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_v \mathcal{F}(\mathcal{R})$ or $\mathcal{F}(\mathcal{R}_{-U}(L)) = \mathcal{F}(\mathcal{R})$.*

Observe that if L is a (safe) strategic vote for v at $\mathcal{R}$ under $\mathcal{F}$, then it is also a (safe) strategic vote for any $u \in S_v$. Indeed, $u \in S_v$ implies $S_u = S_v$ and for any $a \in C$ we have $a \succ_u \mathcal{F}(\mathcal{R})$ if and only if $a \succ_v \mathcal{F}(\mathcal{R})$. Note also that we do not require $L \neq R_v$: indeed, in the preference-based extension $L = R_v$ may be a non-trivial manipulation, as it may induce voters in $S_v \setminus \{v\}$ to switch their preferences to R_v. That is, a voter may manipulate the election simply by asking other voters with similar preferences to vote like he does. Finally, it is easy to see that for any voter v, the set S_v of similar voters is easy to compute.

The two computational problems considered throughout this paper, i.e., the safety of a given manipulation and the existence of a safe manipulation remain relevant for the preference-based model. We will refer to these problems in this setting as $\textsc{IsSafe}^{pr}$ and $\textsc{ExistSafe}^{pr}$, respectively, and use prefix W to denote their weighted variants. The problems (W)$\textsc{IsSafe}^{pr}$ and (W)$\textsc{ExistSafe}^{pr}$ appear to be somewhat harder than their counterparts in the original model. Indeed, while voters in S_v have similar preferences, their truthful votes may be substantially different, so it now matters *which* of the voters in S_v decide to follow the manipulator (rather than just *how many* of them, as in the original model). In particular, it is not clear if $\textsc{IsSafe}^{pr}(\mathcal{F})$ is polynomial-time solvable for any voting rule $\mathcal{F}$. However, it turns out that both of our problems are easy for k-approval, even with weighted voters.

Theorem 9. *For k-approval, the problems* W$\textsc{IsSafe}^{pr}$ *and* W$\textsc{ExistSafe}^{pr}$ *are in* P.

In the preference-based model, a voter v follows a recommendation to vote in a particular way if it comes from a voter whose preferences are similar to those of v. However, this approach does not describe settings where a voter follows a recommendation not so much because he trusts the recommender, but for pragmatic purposes, i.e., because the proposed manipulation advances her own goals. Clearly, this may happen even if the overall preferences of the original manipulator and the follower are substantially different. We will now propose a model that aims to capture this type of scenarios.

Goal-Based Extension If the potential follower's preferences are different from those of the manipulator, his decision to join the manipulating coalition is likely to depend on the specific manipulation that is being proposed. Thus, in this subsection we will define the set of potential followers F in a way that depends both on the original manipulator's identity i and his proposed vote L, i.e., we have $F = F_i(L)$. Note, however, that it is not immediately obvious how to decide whether a voter j can benefit from following i's suggestion to vote L, and thus should be included in the set $F_i(L)$. Indeed, the benefit to j depends on which other voters are in the set $F_i(L)$, which indicates that the definition of the set $F_i(L)$ has to be self-referential.

In more detail, for a given voting rule $\mathcal{F}$, an election (C, V) with a preference profile $\mathcal{R}$, a voter $i \in V$ and a vote L, we say that a voter j is *pivotal for a set $U \subseteq V$ with respect to* (i, L) if $j \notin U$, $R_j \neq L$ and $\mathcal{F}(\mathcal{R}_{-(U\cup\{j\})}(L)) \succ_j \mathcal{F}(\mathcal{R}_{-U}(L))$. That is, a voter j is pivotal for a set U if when the voters in U vote according to L, it is profitable for j to join them. Now, it might appear natural to define the follower set for (i, L) as the set that consists of i and all voters $j \in V$ that are pivotal with respect to (i, L) for some set $U \subseteq V$. However, this definition is too broad: a voter is included as long as it is pivotal for some subset $U \subseteq V$, even if the voters in U cannot possibly benefit from voting L. To exclude such scenarios, we need to require that U itself is also drawn from the follower set. Formally, we say that $F_i(L)$ is a *follower set* for (i, L) if it is a maximal set F that satisfies the following condition:

$$\forall j \in F \; [\, (j = i) \;\vee\; (\exists\, U \subseteq F \text{ s. t. } j \text{ is pivotal for } U \text{ with respect to } (i, L))] \qquad (*)$$

Observe that this means that $F_i(L)$ is a fixed point of a mapping from 2^V to 2^V, i.e., this definition is indeed self-referential. To see that the follower set is uniquely defined for any $i \in V$ and any vote L, note that the union of any two sets that satisfy condition (*) also satisfies (*); note also that we always have $i \in F_i(L)$.

We can now define what it means for L to be a *strategic vote in the goal-based extension* and a *safe strategic vote in the goal-based extension* by replacing the condition $U \subseteq S_i$ with $U \subseteq F_i(L)$ in Definition 2. We will denote the computational problems of checking whether a given vote is a safe strategic vote for a given voter in the goal-based extension and whether a given voter has a safe strategic vote in the goal-based extension by IsSafe^{gl} and ExistSafe^{gl}, respectively, and use the prefix W to refer to weighted versions of these problems.

Two remarks are in order. First, it may be the case that even though i benefits from proposing to vote L, he is never pivotal with respect to (i, L) (this can happen, e.g., if i's weight is much smaller that that of the other voters). Thus, we need to explicitly include i in the set $F_i(L)$, to avoid the paradoxical situation where $i \notin F_i(L)$. Second, our definition of a safe vote only guarantees safety to the original manipulator, but not to her followers. In contrast, in the preference-based extension, any vote that is safe for the original manipulator is also safe for all similar voters.

The definition of a safe strategic vote in the goal-based extension captures a number of situations not accounted for by the definition of a safe strategic vote in the preference-based extension. To see this, consider the following example.

Example 2. Consider an election with the set of candidates $C = \{a, b, c, d, e\}$, and three voters 1, 2, and 3, whose preferences are given by $a \succ_1 b \succ_1 c \succ_1 d \succ_1 e$, $e \succ_2 b \succ_2 a \succ_2 d \succ_2 c$, and $d \succ_3 a \succ_3 b \succ_3 c \succ_3 e$. Suppose that the voting rule is Plurality, and the ties are broken according to the priority order $d \succ b \succ c \succ e \succ a$.

Under truthful voting, d is the winner, so we have $S_1 \neq S_2$. Thus, in the preference-based extension, a vote that ranks a first is a safe strategic vote for voter 2, but a vote that ranks b first is not. On the other hand, let L be any vote that ranks b first. Then $F_1(L) = F_2(L) = \{1, 2\}$. Indeed, if voter 1 switches to voting L, the winner is still d, but it becomes profitable for voter 2 to join her, and vice versa. On the other hand, it is easy to see that voter 3 cannot profit by voting L. It follows that in the goal-based extension L is a safe strategic vote for voter 1.

From a practical perspective, it is plausible that in Example 2 voters 1 and 2 would be able to reconcile their differences (even though they are substantial—voter 1 ranks voter 2's favorite candidate last) and jointly vote for b, as this is beneficial for both of them. Thus, at least in some situations the model provided by the goal-based extension is intuitively more appealing. However, computationally it is considerably harder to deal with than the preference-based extension.

Indeed, it is not immediately clear how to compute the set $F_i(L)$, as its definition is non-algorithmic in nature. While one can consider all subsets of V and check whether they satisfy condition (*), this approach is obviously inefficient. We can avoid full enumeration if have access to a procedure $\mathcal{A}(i, L, j, W)$ that for each pair (i, L), each voter $j \in V$ and each set $W \subseteq V$ can check if $j = i$ or there is a set $U \subseteq W$ such that j is pivotal for U with respect to (i, L). Indeed, if this is the case, we can compute $F_i(L)$ as follows. We start with $W = V$, run $\mathcal{A}(i, L, j, W)$ for all $j \in W$, and let W' to be the set of all voters for which $\mathcal{A}(i, L, j, W)$ outputs "yes". We then set $W = W'$, and iterate this step until $W = W'$. In the end, we set $F_i(L) = W$. The correctness of this procedure can be proven by induction on the number of iterations and follows from the fact that if a set W contains no subset U that is pivotal for j, then no smaller set $W' \subset W$ can contain such a subset. Moreover, since each iteration reduces the size of W, the process converges after at most n iterations. Thus, this algorithm runs in polynomial time if the procedure $\mathcal{A}(i, L, j, W)$ is efficiently implementable. We will now show that this is indeed the case for Plurality (with unweighted voters).

Theorem 10. *Given an election (C, V) with a preference profile $\mathcal{R}$ and unweighted voters, a manipulator i, and a vote L, we can compute the set $F_i(L)$ with respect to Plurality in time polynomial in the input size.*

We can use Theorem 10 to show that under Plurality one can determine in polynomial time whether a given vote L is safe for a player i, as well as find a safe strategic vote for i if one exists, as long as the voters are unweighted.

Theorem 11. *The problems* IsSafe^{gl} *and* ExistSafe^{gl} *are polynomial-time solvable for Plurality.*

For weighted voters, computing the follower set is computationally hard even for Plurality. While this result does not immediately imply that wIsSafe^{gl} and wExistSafe^{gl} are also hard for Plurality, it indicates that these problems are unlikely to be easily solvable.

Theorem 12. *Given an instance* $(C, V, \mathbf{w}, \mathcal{R})$ *of Plurality elections, voters* $i, j \in V$ *and a vote* L*, it is* NP*-hard to decide whether* $j \in F_i(L)$*.*

Just a little further afield, checking whether a given vote is safe with respect to 3-approval is computationally hard even for unweighted voters. This is in contrast with the standard model and the preference-based extension, where safely manipulating k-approval is easy for arbitrary k.

Theorem 13. IsSafe^{gl} *is* coNP*-hard for* 3*-approval.*

Thus, while the preference-based extension appears to be similar to the original model of [11] from the computational perspective, the goal-based extension is considerably more difficult to work with.

7 Conclusions

In this paper, we started the investigation of algorithmic complexity of safe manipulation, as defined by Slinko and White [11]. We showed that finding a safe manipulation is easy for k-approval for an arbitrary value of k and for Bucklin, even with weighted voters. Somewhat surprisingly, checking whether a given manipulation is safe appears to be a more difficult problem, at least for weighted voters: while this problem is polynomial-time solvable for k-approval, it is coNP-hard for Bucklin. For the Borda rule, both checking whether a given manipulation is safe and identifying a safe manipulation is hard when the voters are weighted.

We also proposed two ways of extending the notion of safe manipulation to heterogeneous groups of manipulators, and initiated the study of computational complexity of related questions. Our first extension of the model of [11] is very simple and natural, and seems to behave similarly to the original model from the algorithmic perspective. However, arguably, it does not capture some of the scenarios that may occur in practice. Our second model is considerably richer, but many of the associated computational problems become intractable.

A natural open question is determining the complexity of finding a safe strategic vote for voting rules not considered in this paper, such as Copeland, Ranked Pairs, or Maximin. Moreover, for some of the voting rules we have investigated, the picture given by this paper is incomplete. In particular, it would be interesting to understand the computational complexity of finding a safe manipulation for Borda (and, more generally, for all scoring rules) for unweighted voters. The problem for Borda is particularly intriguing as this is perhaps the only widely studied voting rule for which the complexity of unweighted coalitional manipulation in the standard model is not known.

Other exciting research directions include formalizing and investigating the problem of selecting the best safe manipulation (is it the one that succeeds more often, or one that achieves better results when it succeeds?), and extending our analysis to other types of tie-breaking rules, such as, e.g., randomized tie-breaking rules. However, the latter question may require modifying the notion of a safe manipulation, as the outcome of a strategic vote becomes a probability distribution over the alternatives.

References

[1] J. J. Bartholdi, III, C. A. Tovey, and M. Trick. The computational difficulty of manipulating an election, *Social Choice and Welfare*, 6:227–241, 1989.

[2] J. J. Bartholdi, III and J. B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[3] V. Conitzer, M. Rognlie, L. Xia. Preference Functions that Score Rankings and Maximum Likelihood Estimation, in *Proc. IJCAI-09*, pp. 109–115, 2009.

[4] V. Conitzer, T. Sandholm, and J. Lang, When ere elections with few candidates hard to manipulate?, *J. ACM*, 54:1–33.

[5] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: ties matter, in *Proc. of AAMAS-08*, 2008.

[6] M. R. Garey, D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[7] A. F. Gibbard. Manipulation of voting schemes: a general result, *Econometrica*, 41:597–601, 1973.

[8] H. Moulin. Choice functions over a finite set: a summary, *Social Choice and Welfare*, 2:147–160, 1985.

[9] C. Papadimitriou. Computational Complexity, Addison Wesley, 1994.

[10] M. A. Satterthwaite. Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions, *Journal of Economic Theory*, 10:187–217, 1975.

[11] A. Slinko, S. White. Non-dictatorial Social Choice Rules Are Safely Manipulable, in *Proc. of COMSOC-08*, 2008.

[12] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules, in *Proc. of IJCAI-09*, 2009.

[13] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem, *Artificial Intelligence*, 173(2):392–412, 2009.

Noam Hazon
Department of Computer Science
Bar Ilan University
Israel
Email: `hazonn@cs.biu.ac.il`

Edith Elkind
Division of Mathematical Science
Nanyang Technological University
Singapore
Email: `eelkind@ntu.edu.sg`

An Empirical Study of Borda Manipulation

Jessica Davies, George Katsirelos, Nina Narodytska, and Toby Walsh

Abstract

We study the problem of coalitional manipulation in elections using the unweighted Borda rule. We provide empirical evidence of the manipulability of Borda elections in the form of two new greedy manipulation algorithms based on intuitions from the bin-packing and multiprocessor scheduling domains. Although we have not been able to show that these algorithms beat existing methods in the worst-case, our empirical evaluation shows that they significantly outperform the existing method and are able to find optimal manipulations in the vast majority of the randomly generated elections that we tested. These empirical results provide further evidence that the Borda rule provides little defense against coalitional manipulation.

1 Introduction

Elections are a well established mechanism to aggregate the preferences of individuals to reach a consensus decision. New applications of voting and social choice have emerged in the field of multiagent systems and are used on a daily basis by many people in the form of polls and ratings systems on the internet. As an election is meant to be a fair way of reaching a decision, it is important to study the weaknesses of different voting systems with respect to their vulnerability to manipulation, bribery and control. In this paper we focus on the manipulation problem, where a coalition of agents votes to ensure a desired outcome rather than reporting their true preferences. It is assumed that the manipulators act with full knowledge of the votes of the remaining electorate, but even so, the structure of the voting system may make it difficult to ensure that the desired candidate wins. No practical voting system can prevent a coalition of enough manipulators from achieving their goal in all elections. However, some mechanisms may be easier to manipulate than others. For example, the required size of the coalition may be impractical, especially in real-world settings where obtaining the cooperation of and coordinating more than two or three people can be difficult. Even if the number of extra votes isn't a concern, calculating the required set of manipulator votes may be computationally infeasible.

In this work we study the voting system based on using the Borda rule to aggregate the votes. The Borda rule is a positional scoring rule proposed by the French scientist Jean-Charles de Borda in 1770. Like all positional scoring rules, each voter simply ranks the m candidates according to their preference. The votes are aggregated by adding a score of $m - k$ to a candidate for each time it appears k^{th} in a vote. The candidates with the highest aggregated score win the election. The simplicity of this rule may have contributed to its independent reinvention on at least one other occasion; political elections in two Pacific island states use slight modifications of the Borda rule [11]. It is also commonly used in competitions such as the Eurovision song contest, the election of the Most Valuable Player in major league baseball, and the Robocup competition.

The susceptibility of Borda elections to manipulation has been strongly suggested by recent theoretical work. Although the problem is NP-hard if the manipulators' votes are weighted [6], in the unweighted case the complexity class is still frustratingly unknown. Xia et al. observe that:

> *"The exact complexity of the problem [coalition manipulation with unweighted votes] is now known with respect to almost all of the prominent voting rules, with the glaring exception of Borda"* [17]

A number of recent theoretical results suggest that manipulation may often be computational easy [5, 10, 15, 16]. Brelsford et al. [3] showed that weighted (and unweighted) Borda manipulation has a FPTAS, which means that finding a very close to optimal manipulation can be done in polynomial time. Along these lines, Zuckerman et al. [19] gave a simple greedy algorithm to calculate a manipulation, that in the unweighted case uses at most one more manipulator than is optimal. In addition, even Borda himself appears to have recognised that his rule was susceptible to manipulation, having retorted that:

> *"My scheme is intended only for honest men"*, quoted on page 182 of [2]

More recently, strategic voting was identified in the 1991 presidential candidate elections in the Republic of Kiribati (where a variant of the Borda rule is used) [11]. This suggests that the manipulability of the Borda rule is not just a theoretical possibility but a practical reality.

The manipulability of voting rules has also been studied empirically [13, 14]. For example, Walsh studied the Single Transferable Vote rule, which is theoretically NP-hard to manipulate. However, he provided ample evidence that in practise, elections using this rule are easy to manipulate [14]. We provide further empirical evidence that the Borda rule provides little defense to manipulation, by showing that in many elections, an optimal manipulation can be found (and often verified) in polynomial time. Our starting point is the greedy algorithm of Zuckerman et al. [19], which decides the vote of each manipulator in turn by reversing the candidates ordered by current score. Although this algorithm provides a guarantee that in the worst case it only uses one more manipulator than is optimal, the theoretical analysis does not extend to answer the question of how frequently it uses this extra manipulator. Perhaps another greedy algorithm exists that finds the optimal manipulation much more frequently. If so, it could be used in conjunction with that of Zuckerman et al. to provide a verified optimal solution whenever it finds a solution using one fewer manipulator. We introduce two new greedy algorithms, based on intuitions from the bin-packing and multiprocessor scheduling domains, and provide theoretical and empirical comparison between their performance and that of Zuckerman et al.'s greedy algorithm. The new algorithms result in a significant improvement over Zuckerman et al.'s algorithm, allowing the optimal manipulation to be found and verified quickly on 99% of more than 60,000 randomly generated elections.

The paper continues with the definitions and background in Section 2, followed in Section 3 by our new greedy algorithms. Section 4 presents the experimental results and we conclude in the last section.

2 Background

In this section we introduce notation and definitions that will be used throughout the paper.

An *election* is a pair $E = (V, m)$ where m is the number of candidates. We refer to the distinguished candidate who the manipulators want to win the election as candidate $1 \leq d \leq m$; the other $m - 1$ candidates are then the *competing* candidates. V is a set of *votes*, where a vote is an ordering of the candidates $v = c_1 > c_2 > ... > c_m$ such that $\bigcup c_j = \{1, .., m\}$. Given a vote v, the *score* of a candidate i under the Borda rule, denoted $s(v, i)$, equals $m - k$ where $c_k = i$. If V is a set of votes, then the score of a candidate i given by these votes is $s(V, i) = \Sigma_{v \in V} s(v, i)$. Given an election $E = (V, m)$, the *winners* are defined as those candidates $1 \leq i \leq m$ such that $s(V, i)$ is maximal. A *manipulation* of an election $E = (V, m)$ is a set of manipulator votes M such that $s(V \cup M, d) \geq s(V \cup M, i)$ for all $i \neq d$. We assume that ties are broken in favour of the manipulators. The manipulation problem is to find a manipulation such that $|M| = n$ is minimized.

Sometimes we will refer to a manipulation using n votes as an *n-manipulation*.

We define some additional notation that will be helpful in describing our greedy algorithms.

Definition 1 *Given an election $E = (V, m)$, a number of manipulators n, the* gap *of candidate $1 \leq i \leq m$, is defined as $g_{E,n}(i) = s(V, d) + n(m-1) - s(V, i)$. If the context is clear, we call the gap of candidate i simply g_i.*

Intuitively, the gap of a candidate i is the difference between the score the distinguished candidate receives after the manipulators have voted, and the score of i before the manipulators vote. Without loss of generality, we assume that the manipulators always rank d first. Note that if g_i is negative for any i, then there is no n-manipulation.

Definition 2 *Given an election $E = (V, m)$, an n-manipulation matrix $A_{E,n}$ is an $n \times m$ matrix such that all elements of column d are equal to $m-1$, each row contains all numbers from 0 to $m-1$ and column i sums to at most $g_{E,n}(i)$ for all $1 \leq i \leq m$.*

It is easy to see that such a matrix represents an n-manipulation of the election, where each column represents a competing candidate, and each row corresponds to the vote of a distinct manipulator. We will drop the parameters E and n and refer to matrix A when the context is clear. We use the notation $A(i)$ to denote the i^{th} column of A, and $sum(A(i))$ is defined to be the sum of the elements in $A(i)$.

Observation 1 *Given an election $E = (V, m)$ and a number of manipulators n, if $\Sigma_{i=1}^{m-1} g_{E,n}(i) < (n/2)(m-1)(m-2)$ then there is no n-manipulation.*

This follows directly from Definition 2, since each of the n manipulator votes contributes a total of $\Sigma_{k=0}^{m-2} k = (1/2)(m-1)(m-2)$ score to the scores of the competing candidates. In other words, there must be enough difference between the original scores of the competing candidates and the achievable score of the distinguished candidate, otherwise an n-manipulation can not exist. We call the multiset containing n copies of each $0 \leq k \leq m-2$ S_n.

The greedy algorithm of Zuckerman et al. [19] is shown in Figure 1, and from now on will be referred to as `REVERSE`. The manipulation matrix A starts off empty, and is augmented row by row until enough manipulators have been added that the distinguished candidate wins. The `sort` procedure puts the distinguished candidate first, and then sorts the competing candidates in increasing order by their current score, in order to create the next manipulator's vote.

Example 1. Suppose $E = (V, 5)$ where V contains the votes $v_1 = 1 > 2 > 3 > 4 > 5$, $v_2 = 2 > 3 > 4 > 1 > 5$, $v_3 = 3 > 4 > 1 > 2 > 5$ and $v_4 = 4 > 1 > 2 > 3 > 5$, and $d = 5$. Then $s(V, 5) = 0$, and $s(V, i) = 10$ for all competing candidates $i < 5$. In order for candidate 5 to win the election, at least 4 manipulators are required since $\Sigma_i g_{E,3}(i) = 4 * (4 * 3 - 10) = 8$ but $(n/2)(m-1)(m-2) = 1.5 * 4 * 3 = 18$. `REVERSE` will make the first manipulator vote $w_1 = 5 > 1 > 2 > 3 > 4$ (ordering the competing candidates arbitrarily), at which point, e.g., $s(V \cup \{w_1\}, 1) = 10 + 3 = 13$. The candidates' scores are shown in Figure 2 after each iteration of the ***while*** loop. Since $s(V \cup \{w_1, w_2, w_3, w_4\}, 5) = 16$, `REVERSE` finds the optimal manipulation.

3 Greedy Algorithms for Borda Manipulation

The definition of manipulation matrix from Section 2 is a useful abstraction, that suggests a connection to bin-packing or multiprocessor scheduling [4]. Intuitively, the elements of the manipulators

```
REVERSE(V,m,d)
A[i] ← ∅ for all 1≤i≤m
n ← 0
while max_i{sum(A[i]) + s(V,i)} > sum(A[d]) + s(V,d)
    w ← sort{i < j ⟺ (sum(A[i])+s(V,i) < sum(A[j])+s(V,j) or i=d)}
    A[i].push(s(w,i)) for all i
    n ← n + 1
return A
```

Figure 1: The greedy algorithm of Zuckerman et al. [19].

Candidate i	1	2	3	4	5
$s(V,i)$	10	10	10	10	0
$s(V \cup \{w_1\}, i)$	13	12	11	10	4
$s(V \cup \{w_1, w_2\}, i)$	13	13	13	13	8
$s(V \cup \{w_1, w_2, w_3\}, i)$	16	15	14	13	12
$s(V \cup \{w_1, w_2, w_3, w_4\}, i)$	16	16	16	16	16

Figure 2: Scores given by `REVERSE`, for Example 1.

votes, S_n, must be assigned to the columns of A such that the sum of each column is at most g_i. In the bin-packing problem, a set of objects with sizes between zero and one must be grouped into a minimum number of bins such that the sum of the objects in each bin is at most one. So in our case, the set of objects would be S_n, representing the elements whose positions in manipulation matrix A are initially unknown. One of the main differences is that our matrix A has a constraint on each row, that it must contain all values from 0 to $m-1$, and it is not clear how this translates to other domains. Luckily, Theorem 3.1 tells us that we don't have to worry about this constraint. If a correctly sized matrix B containing n elements equal to j for each $0 \leq j \leq m-1$ can be found such that the column sums are at most the candidate's gaps and column d contains all the $m-1$'s, then it can always be converted to a manipulation matrix A.

Theorem 3.1 *Suppose there exists an $n \times m$ matrix B such that the total number of elements in B equal to k, for each $0 \leq k \leq m-1$ is n. Let the sum of the elements in the i^{th} column of B be g_i. Then there is another $n \times m$ matrix A with the same set of elements as B and the same column sums, such that each row contains exactly one element equal to k, for each $0 \leq k \leq m-1$.*

Proof By induction on n. When $n = 1$, we have $B = [b_{1,1}, ..., b_{1,m}]$ such that B contains exactly one element of value k for each $0 \leq k \leq m-1$. Therefore, just set $A = B$.

Assume that the theorem holds for all numbers of rows less than n. We prove that it also holds for n rows. Let B be an $n \times m$ matrix such that the total number of elements in B equal to k, for each $0 \leq k \leq m-1$ is n. Let the sum of the elements in the i^{th} column be g_i.

Define a bipartite graph $G = (S \cup T, E)$ such that the set of left-hand vertices is $S = \{0, ..., m-1\}$ (these will represent the set of values of the elements of row 1 in A), and the set of right-hand vertices is $T = \{1, ..., m\}$ representing the columns of B. E contains an edge $(i,j)_k$ for each $i \in S$, $j \in T$ and $1 \leq k \leq n$ such that $i = B(k,j)$.

Note that there can be up to n edges between two vertices i and j. Since every value appears n times in B, $|\{(k,j) : i = B(k,j)\}| = n$ and so the degree of each $i \in S$ is exactly n. For each $j \in T$, the degree will also be n: one edge to each $i = B(k,j)$, $1 \leq k \leq n$.

Therefore, if we take any $P \subseteq S$, $n|P|$ edges leave P. Since every vertex in T is also of degree n, each vertex in the neighbourhood of P, $nbhd(P)$, can accommodate at most n incoming edges.

Therefore, $|nbhd(P)|$ is not less than $|P|$. Since the Hall condition holds [8], there is a perfect matching in G that assigns each value from 0 to $m-1$ to a position in the first row of B, as follows.

Let $M = \{e_1, ..., e_m\} \subseteq E$ be the set of edges in the matching. For each $e = (i, j)_k \in M$, let $A(1, j) = i$. Since M is a matching, each i, $0 \leq i \leq m-1$ appears in exactly one column, and each column is assigned exactly one element. Therefore, the first row of A is well defined. Also note that for each column j, $A(1, j)$ appears in the j^{th} column of B.

Let B' be the matrix defined by taking B and removing one element equal to $A(1, j)$ from each column j. Then B' is an $n-1 \times m$ matrix containing exactly $n-1$ elements equal to i for each $0 \leq i \leq m-1$, since the elements removed were one of each value. The column sums for B' are $g_j - A(1, j)$ for all columns j. By the induction hypothesis, there exists an $n-1 \times m$ matrix A' such that A' contains the same elements as B' and the same column sums, but each row of A' contains exactly one element equal to i, for $0 \leq i \leq m-1$. Given that we've already defined the first row of A, let the remaining $n-1$ rows be A'. Then A contains the same set of values as B, with the same column sums $A(1, j) + (g_j - A(1, j)) = g_j$, and every row of A contains exactly one element equal to i, for each $0 \leq i \leq m-1$.

Therefore, by induction, the theorem holds for all n. □

If a matrix B exists whose column sums are at most the value of the candidates' gaps, and sum($B[d]$) = g_d, then matrix A gives a manipulation, where each row of A defines the vote of one of the manipulators. Therefore, we can devise algorithms to discover B and be assured that A exists.

However, the manipulation problem has two additional differences to bin-packing. First, the number of objects in each bin must be exactly n, while bin-packing has no such constraint. Secondly, each of our bins has a different maximum capacity g_i. The former constraint has been studied in the multiprocessor scheduling domain, where the problem is to schedule jobs on a set of n processors such that the memory resources are never exceeded and the time to complete all jobs is minimized [9]. Our problem corresponds to the case where each job takes a unit of processing time. For each element $a \in S_n$, there is a job with memory requirement equal to a. The number of processors is equal to the number of manipulators n, and the amount of available memory resource at time step i is equal to g_i. We wish to find a schedule that uses $m-1$ time steps, which will be possible if an n-manipulation exists. Krause et al. consider the case where the memory resource remains constant over time, and present theoretical analysis of a simple scheduling algorithm that assigns the jobs one at a time to particular time steps. Their scheduler takes the unassigned job with largest memory requirements and assigns it to a time step (with at least one processor free), that has the maximum remaining available memory. If no time step exists that can accommodate this job, a new time step is added.

Our first greedy algorithm is based on this same intuition, where it translates to giving the largest scores to the competing candidates that have the least score so far. In this it is similar to `REVERSE`, but we are now free to pursue this heuristic strictly, while `REVERSE` for example decides which candidate the second voter's $m-2$ should be assigned to *after* the smaller scores of the first manipulator are assigned. This can sometimes be an advantage, but it may also lead the algorithm to make more serious mistakes, as we will show.

3.1 Largest Score in Largest Gap

Our first greedy algorithm, `LSLG` is shown in Figure 3. `LSLG` takes the number of manipulators as an argument and returns the matrix B (from Theorem 3.1) if it is able to find an n-manipulation. On line 1, the matrix B (represented as an array of vectors) is initialized so that every column vector is empty. On line 2, the column corresponding to the distinguished candidate is filled with the maximum value, $m-1$. On line 3, the array S is initialized with the sorted elements of S_n defined

```
LSLG(V,n,d)
   // B[i] is the i^th column of B
1. B[i] ← ∅ for all 1≤i≤m
   // B[d] is filled with n m-1's
2. B[d] ← {m-1,...,m-1}
   // Each score is repeated n times in S
3. S ← {m-2,...,m-2,m-3,...,m-3,...,1,...,1,0,...,0}
4. while S ≠ {}
      // The column of B that contains fewer than n elements,
      // with the lowest sum
5.    c ← argmin_i{sum(B[i]) + s(V,i) : |B[i]| < n}
6.    B[c].push(S[0])
7.    S ← S - S[0]
8. if sum(B[d]) + s(V,d) ≥ max_i{sum(B[i]) + s(V,i)}
9.    return B
10.else
11.   return Failure
```

Figure 3: The greedy algorithm based on placing the largest remaining score in the column of A with the most room.

in Section 2. Each iteration of the ***while*** loop on lines 4-7 removes the first (largest) element of S and pushes it (on line 6) into the column of B that has the lowest sum so far. Note that we use the notation $|B(i)|$ to denote the current number of elements in the i^{th} column of B. Once all elements of S have been assigned, the loop terminates and line 8 checks if a valid manipulation has been produced. If so, B is returned, and if not, the algorithm reports Failure.

The following proposition shows that this algorithm can sometimes find an optimal manipulation when `REVERSE` fails, and this is true for an infinite family of instances.

Proposition 1 *Let $E = (V, m)$ be an election such that $m > 2$ is even, $d = m$, $s(V, d) = 0$ and $s(V, i) = \frac{m}{2} + i$ for all $i \neq d$. Then `LSLG` finds an optimal 2-manipulation, but `REVERSE` produces a 3-manipulation.*

Proof

First, note that two non-manipulator votes are always sufficient to create such an election. Let $\sigma =< 1, 2, ..., m-1 >$ and let

$$\sigma' =< \frac{m}{2}+1, \frac{m}{2}+2, ..., \frac{m}{2}+\frac{m}{2}-1, 1, 2, ..., \frac{m}{2} >$$

Then $\sigma + \sigma' =$

$$< \left(1+\frac{m}{2}+1\right), \left(2+\frac{m}{2}+2\right), ..., \left(\frac{m}{2}-1+\frac{m}{2}+\frac{m}{2}-1\right), \left(\frac{m}{2}+1\right), ..., \left(m-1+\frac{m}{2}\right) >$$

which gives us $\frac{m}{2} + 2x$ for $1 \leq x \leq \frac{m}{2} - 1$ and $\frac{m}{2} + 2x - 1$ for $1 \leq x \leq \frac{m}{2}$, or in other words, $\frac{m}{2} + i$ for all $1 \leq i \leq m-1$ (i.e. all $i \neq d$).

The first vote generated by `REVERSE` is $r_1 = m > 1 > 2 > ... > m-1$, after which $s(V \cup \{r_1\}, i) = \frac{m}{2} + m - 1$ for all competing candidates, which is larger than the score of the distinguished candidate $s(V \cup \{r_1\}, m) = m - 1$. Therefore another manipulator is added, without loss of generality its vote is $r_2 = m > 1 > 2 > ....m-1$. The resulting scores of the competing candidates are $s(V \cup \{r_1, r_2\}, i) = \frac{m}{2} + (m-1) + (m-i-1) = (5/2)m - 2 - i$. So candidate $i = 1$ still has larger score than $s(V \cup \{r_1, r_2\}, m) = 2m - 2$. Therefore, REVERSE does not find a 2-manipulation.

The first $m-1$ iterations of LSLG will place the k^{th} largest score from S_2 into the k^{th} column of matrix B for $1 \leq k \leq m-1$. Note that the k^{th} largest score is $m - 2 - \lfloor (k-1)/2 \rfloor$. Let B_{m-1} be the matrix at this point. Then $sum(B_{m-1}(i)) + s(V, i) = (m - 2 - \lfloor (i-1)/2 \rfloor) + \frac{m}{2} + i$ for all $i < m$. The next $m-1$ iterations of LSLG will place the k^{th} largest score from S_2 into the k^{th} column of matrix B for $m \leq k \leq 2(m-1)$. So column $i < m$ will receive the element $\frac{m}{2} - 1 - \lceil (i-1)/2 \rceil$. Let $B_{2(m-1)}$ be the matrix when the loop terminates. Then $sum(B_{2(m-1)}(i)) + s(V, i) = (m - 2 - \lfloor (i-1)/2 \rfloor) + (\frac{m}{2} + i) + (\frac{m}{2} - 1 - \lceil (i-1)/2 \rceil) = 2(m-1)$ for all $i < m$, while the achievable score of m is also $2(m-1)$. Therefore, LSLG does find a 2-manipulation. Figure 4 shows the matrix generated by LSLG (column $d = m$ is omitted), where the shaded areas represents the scores $s(V, i)$ for each $i < m$.

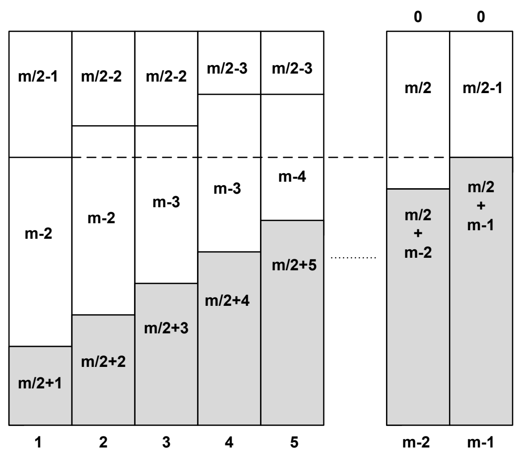


Figure 4: The 2-manipulation generated by LSLG for the election in Proposition 1

□

Unfortunately, LSLG does not share the guarantee of REVERSE that in the worst case it requires one extra manipulator than is optimal. In fact, Theorem 3.2 shows that the number of extra manipulators LSLG might require is unbounded.

Theorem 3.2 *Let k be positive integer greater than zero and divisible by 36. Let $s(V, 1) = 6k$, $s(V, 2) = 4k$, $s(V, 3) = 2k$, $s(V, 4) = 0$ be the scores of four candidates after some non-manipulators V vote, and let $d = 4$. Then REVERSE will find the optimal manipulation, using $2k$ manipulators. However, LSLG requires at least $2k + k/9 - 3$ manipulators.*

Proof First, we should mention that for any k there is a set of votes V_k that gives the specified scores to the four candidates: V_k is simply $2k$ votes, all equal to $1 > 2 > 3 > 4$. REVERSE will use $2k$ manipulators, all voting $4 > 3 > 2 > 1$, to achieve a score of $6k$ for all candidates (the only optimal manipulation). It remains to argue that LSLG requires more than $2k + k/9 - 4$ manipulators. Assume for contradiction that we find a manipulation using $n = 2k + k/9 - 4 = 19k/9 - 4$ manipulators. We will follow the execution of LSLG until a contradiction is obtained. Note that given our definition of n, since k is divisible by 4 and 9, $\frac{n-k}{2}$ is an integer.

First, the algorithm will place k 2's in B[3], at which point $sum(B[3]) = 2k + 2k = 4k = s(V_k, 2)$. Then it will begin to place 2's in columns B[2] and B[3] evenly, until all remaining $n - k$ 2's have been placed into B. At this point, B[2] contains $\frac{n-k}{2}$ 2's, and the number of 2's that B[3] contains is $k + \frac{n-k}{2} = k/2 + n/2 = k/2 + (19k/9 - 4)/2 = 14k/9 - 2 < 19k/9 - 4 = n$. So at this point, B[3] is not full yet and B[2] isn't either (it has fewer elements than B[3]). Both columns sum to $4k + 2(\frac{n-k}{2}) = 46k/9 - 4 = 5k + k/9 - 4 < 6k$. Therefore, the algorithm will start putting 1's in both B[2] and B[3] evenly, until either their column sums reach $6k$ or B[3] gets filled. In fact, B[3] will be filled before its sum reaches $6k$, since B[3] requires $\frac{n-k}{2}$ more elements to be filled, but at this point, $sum(B[2]) = sum(B[3]) = 46k/9 - 4 + \frac{n-k}{2} = 51k/9 - 6 = 5k + 2k/3 - 6 < 6k$.

Now, the algorithm will continue by putting $k/3 + 6$ 1's into B[2], at which point $sum(B[2]) = 51k/9 - 6 + k/3 + 6 = 6k$. Then the algorithm will start putting 1's evenly in both B[1] and B[2],

```
LSLA(V,n)
1. B[i] ← ∅ for all 1≤i≤m
   // B[d] is filled with n m-1's
2. B[d] ← {m-1,...,m-1}
   // Each score is repeated n times in S
3. S ← {m-2,...,m-2,m-3,...,m-3,...,1,...,1,0,...,0}
4. while S ≠ {}
     // The column of B with highest average desired score
5.   c ← argmax_i{ [g_i-sum(B[i])] / [n-|B[i]|]] : |B[i]| < n}
6.   s ← chooseScore(g_c-sum(B[c]), S)
7.   B[c].push(s)
8.   S ← S - {s}
9. if sum(B[d]) + s(V,d) ≥ max_i{sum(B[i]) + s(V,i)}
10.  return B
11.else
12.  return Failure

chooseScore(g,S)
1. s ← max{s ∈ S : s ≤ g}
2. if s = None
3.    s = S[0]
4. return s
```

Figure 5: The greedy algorithm based on average desired score, for n manipulators.

until either it runs out of 1's or B[2] is filled. In fact, the 1's will run out before B[2] is filled, since B[2] requires $n-(\frac{n-k}{2}+\frac{n-k}{2}+k/3+6) = 2k/3-6$ more elements, which is equal to the number of remaining 1's, but these are spread between B[1] and B[2]. So B[2] will get $(2k/3-6)/2 = k/3-3$ additional 1's, for a total of $sum(B[2]) = 4k + 2(\frac{n-k}{2}) + \frac{n-k}{2} + k/3 + 6 + k/3 - 3 = 19k/3 - 3 > 19k/3 - 12 = 3n$. Since $sum(B[2]) > 3n$ there is no manipulation using $n = 19k/9 - 4$ manipulators. Therefore, LSLG requires at least $n + 1 = 2k + k/9 - 3$ manipulators. □

This result shows the weakness of LSLG, that it only considers the relative sizes of the competing candidates' current scores. Therefore if two candidates' column sums ever become equal during LSLG, they will often be treated equivalently for the remainder of the iterations. In the example from Theorem 3.2, this is the fatal mistake, since at the point where $sum(B[3])$ becomes equal to $sum(B[2])$, column 3 requires fewer additional elements before it is filled (i.e. $|B[2]| < |B[3]|$). Therefore, it is important for column 3 to receive larger elements than column 2. In fact, all of the largest elements must be taken by column 3, and none given to column 2. However, LSLG will begin treating the two equal columns the same, distributing the remaining 2's evenly between $B[2]$ and $B[3]$. This observation motivates our second greedy algorihthm.

3.2 Average Desired Score

The second greedy algorithm is based on the idea that it is not enough to simply assign the largest scores to the columns of B that have the largest gap. Each column of B also requires exactly n elements in order to be filled, where n is the number of manipulators currently attempted. To balance these two requirements, we can look at the *remaining gap* $g_i - sum(B[i])$ and divide it by the remaining number of scores that must be added to column i, $n - |B[i]|$. Notice that if we had $n - |B[i]|$ scores of this average size available (for each i), we could fill every column of B perfectly. Since we don't, a sensible heuristic is to put the largest scores in the columns that have

j	1	2	3	4	5	6	7	8
$S(V_1, j)$	67	60	59	58	58	52	52	42
$S(V_2, j)$	41	34	30	27	27	26	25	14

Figure 6: Examples where `LSLG` beats `LSLA` by finding the optimal number of manipulators vs. using one extra.

largest average desired score. This algorithm, called `LSLA`, is shown in Figure 5.

The structure of `LSLA` is similar to `LSLG`, so it will not be explained line by line. Note that on line 5 of `LSLA` we need some way to break ties between candidates that have the same average desired score. We could break ties arbitrarily, but we also consider choosing the candidate i with minimum $|B[i]|$ since this column needs more additional scores. We found experimentally that the latter tie breaking policy works better overall, although there are some instances where only the arbitrary policy finds the optimal manipulation. The procedure `chooseScore` is used to avoid violating the maximum column sum g_i earlier than necessary. Given an array of unassigned scores and the size of a column's remaining gap, it returns the largest unassigned score that fits in the remaining gap. We found experimentally that this was vital to finding the optimal manipulation in the majority of cases.

We now compare `LSLA` to the other two greedy algorithms. `LSLA` behaves similarly to `REVERSE` on the instances from Theorem 3.2, and thus it performs better than `LSLG` on an infinite family of instances. In fact, in the next section we will see that we have never found an instance for which `REVERSE` can find an optimal manipulation but `LSLA` fails. However, cases do exist where the simpler greedy algorithm `LSLG` finds the optimal manipulation and `LSLA` fails. Two examples are shown in Figure 6, but analysis of these cases has failed to produce a generalizable pattern. In the next section we provide further experimental evidence of the superiority of `LSLA` compared to the other two algorithms.

4 Empirical Comparison

In this section we compare the performance of `REVERSE`, `LSLG` and `LSLA` from a practical perspective. Our experimental setup is similar to that of Walsh [14]. We consider two methods of generating non-manipulator votes, the uniform random votes model and the Polya Eggenberger urn model [1]. In the uniform random votes model, each vote is drawn uniformly at random from all $m!$ possible votes. In the urn model, votes are drawn from an urn at random, but we place them back into the urn along with a other votes of the same type. This model attempts to capture varying degrees of social homogeneity, or the similarity between voters' preferences. We set $a = m!$, which means that there is a 50% chance that the second vote is the same as the first. It would be interesting to consider varying the degree of vote similarity by experimenting with different values of a. In future work we also intend to study votes generated from real-world elections, e.g. [7]. We generated election instances for numbers of candidates m and numbers of non-manipulators p in $\{2^2, ..., 2^7\}$. We generated 1000 instances for each pair (m, p). Since the votes were generated randomly, for small numbers of candidates some duplicate instances were produced. The total number of distinct Uniform elections obtained was 32679, and the number of distinct Urn elections was 31530.

In order to determine the optimal number of manipulators exactly, we modeled the manipulation problem as a constraint satisfaction problem (CSP). The model we used comes directly from the definition of the manipulation matrix A, Definition 2. In this model, there are $n \times m - 1$ finite domain variables, with domains equal to $\{0, ..., m - 2\}$ that represent the unknown elements of A.

m	# Inst.	`REVERSE`	`LSLG`	`LSLA`	`LSLG` beat `LSLA`
4	2771	2611	2573	2771	0
8	5893	5040	5171	5852	2
16	5966	4579	4889	5883	3
32	5968	4243	4817	5879	1
64	5962	3980	4772	5864	3
128	5942	3897	4747	5821	2
Total	32502	24350	26969	32070	11
%		75	83	99	<1

Figure 7: Number of Uniform elections for which each algorithm found an optimal manipulation.

m	# Inst.	`REVERSE`	`LSLG`	`LSLA`	`LSLG` beat `LSLA`
4	3929	3666	2604	3929	0
8	5501	4709	2755	5496	0
16	5502	4357	2264	5477	1
32	5532	4004	2008	5504	0
64	5494	3712	1815	5475	0
128	5571	3593	1704	5565	0
Total	31529	24041	13150	31446	1
%		76	42	99.7	<1

Figure 8: Number of Urn elections for which each algorithm found an optimal manipulation.

There are n ALLDIFF constraints, each over the variables of a row, that ensure each vote is properly formed. $m-1$ constraints over the variables of each column i of A ensure that their sum is at most g_i. Finally, if $g_i = g_j$ for any two columns $i < j$, we added a constraint that $A[i][0] < A[j][0]$ over their row-1 elements. This breaks the symmetry between the two columns and reduces the number of equivalent solutions to the model. We used the solver Gecode [12] to find a solution to the CSP, using Domain Over Weighted Degree as the variable ordering heuristic. The timeout for Gecode was set to one hour, and all experiments were performed on processors of typical contemporary performance.

We will refer to the number of manipulators used by `REVERSE` as N_r. We ran the three competing greedy algorithms, and if this did not determine the optimal manipulation (i.e. none did better than `REVERSE`), we checked whether Observation 1 or the fact that $g_{E,N_r-1}(i)$ is negative for some candidate i allow us to conclude that a (N_r-1)-manipulation is impossible. If the optimal number of manipulators was still unknown, we attempted to find an (N_r-1)-manipulation using Gecode.

Uniform Elections Using the combined method described above, we were able to determine the optimal number of manipulators in 32502 out of the 32679 distinct Uniform elections. The results are shown in Figure 7, grouped by the number of candidates m. The first column shows the number of candidates, and the second column shows the number of instances for which we report results. The next three columns show the number of instances for which each of the greedy algorithms could find an optimal manipulation. The last column shows the number of instances on which `LSLG` found the optimal solution but `LSLA` did not. These results show that both `LSLG` and `LSLA` provide a significant improvement over `REVERSE`, solving 83% and 99% of instances to optimality overall. We also notice that `REVERSE` solves fewer problems to optimality as the number of candidates increases, while `LSLA` does not seem to suffer from this problem as much: `LSLA` solves 100% of the $m = 4$ instances and 98% of the 128 candidate elections. In addition to the results in the table, we mention that in every one of the 32502 instances, if REVERSE found an n-manipulation either LSLA did too, or LSLA found an $(n-1)$-manipulation.

Urn Elections We were able to determine the optimal number of manipulators for 31529 out of the 31530 unique Urn elections. Figure 8 presents the results, in the same format as Figure 7. REVERSE solves about the same proportion of the Urn instances as it did of the Uniform instances, 76%. However, LSLG performance drops significantly, and is in fact much worse than REVERSE at 42% of instances solved. This can be explained by the structure of the Urn elections, which contain many identical votes. This results in a similar pattern of non-manipulator scores to those in Theorem 3.2 on which LSLG has pathological behavior. Surprisingly, the good performance of LSLA is maintained. LSLA found the optimal manipulation on more than 99% of the instances, dominates REVERSE and only lost one instance to LSLG in this set of experiments.

5 Conclusion

We studied the coalitional manipulation problem in elections using the unweighted Borda rule. We provided insight into the structure of the solutions that allows us to build algorithms that construct a manipulation in a manner similar to bin-packing rather than constructing an entire vote at each step. Using this insight, we proposed two new algorithms, LSLG and LSLA. We have provided no optimality guarantees for these algorithms. In fact, we show that LSLG may require an unbounded number of additional manipulators relative to the optimal. However, there are infinite families of instances in which both algorithms can find the optimal but the algorithm proposed by Zuckerman et al. [19], which does have a worst-case guarantee, can not. In an empirical evaluation performed over more than 60000 randomly generated instances, LSLA finds the optimal manipulation in more than 99% of the cases, is never outperformed by REVERSE and in only 12 instances by LSLG. This result provides further empirical evidence that the unweighted Borda rule can be manipulated effectively using relatively simple algorithms.

In future work, we intend to determine whether we can provide theoretical optimality guarantees for LSLA similar to those that are known for REVERSE and theoretically verify the strict dominance that we observed empirically. Further, we intend to investigate whether we can extend our algorithms to always find the optimal number of manipulators for these elections. Another question that arises from this work is whether similar insights can be developed for other scoring rules.

Acknowledgments: Jessica Davies is supported by the National Research Council of Canada. George Katsirelos is supported by the ANR UNLOC project ANR 08-BLAN-0289-01. Nina Narodytska and Toby Walsh are funded by the Australian Government's Department of Broadband, Communications and the Digital Economy and the Australian Research Council.

References

[1] S. Berg. Paradox of Voting Under an Urn Model: The Effect of Homogeneity. *Public Choice*, 47:377-387, 1985.

[2] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[3] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor and I. Schnoor. Approximability of Manipulating Elections. In *Proc. 23rd Conference on Artificial Intelligence (AAAI-2008)*, p.44-49, 2008.

[4] E. Coffman, J. Csirik and J. Leung. Variants of Classical One-Dimensional Bin Packing. Chapter 33 of *Handbook of Approximation Algorithms and Meta-Heuristics*, Teofilo Gonzalez, ed., Francis and Taylor Books (CRC Press), 2006.

[5] V. Conitzer and T. Sandholm. Nonexistence of Voting Rules That Are Usually Hard to Manipulate. In *Proc. of the 21st National Conference on AI (AAAI-06)*,2006.

[6] V. Conitzer, T. Sandholm and J. Lang. When are Elections with Few Candidates Hard to Manipulate?. In *Journal of the Association for Computing Machinery*, 54, 2007.

[7] J. Dobra. An Approach to Empirical Studies of Voting Paradoxes: An Update and Extension. *Public Choice*, 41:241-250, 1983.

[8] P. Hall. On Representatives of Subsets. *Journal of the London Mathematical Society*, 10:26-30, 1935.

[9] K. Krause, V. Shen and H. Schwetman. Analysis of Several Task-Scheduling Algorithms for a Model of Multiprogramming Computer Systems. *Journal of the Association for Computing Machinery*, 22(4):522-550, 1975.

[10] A. Procaccia and J. Rosenschein. Junta Distributions and the Average-Case Complexity of Manipulating Elections. *Journal of Artificial Intelligence Research*, 28:157-181, 2007.

[11] B. Reilly. Social Choice in the South Seas: Electoral Innovation and the Borda Count in the Pacific Island Countries. *International Political Review*, 23(4):355-372, 2002.

[12] C. Schulte, M. Lagerkvist and G. Tack. Gecode. http://www.gecode.org/

[13] T. Walsh. Where are the Really Hard Manipulation Problems? The Phase Transition in Manipulating the Veto Rule. In *Proc. of IJCAI-2009*, 2009.

[14] T. Walsh. An Empirical Study of the Manipulability of Single Transferable Voting. *to appear in Proc. of ECAI 2010*.

[15] L. Xia and V. Conitzer. Generalized Scoring Rules and the Frequency of Coalitional Manipulability. In *Proc. of the 9th ACM Conference on Electronic Commerce (EC-08)*, p.109-118, 2008.

[16] L. Xia and V. Conitzer. A Sufficient Condition for Voting Rules to Be Frequently Manipulable. In *Proc. of the 9th ACM Conference on Electronic Commerce (EC-08)*, p.99-108, 2008.

[17] L. Xia, V. Conitzer and A. Procaccia. A Scheduling Approach to Coalitional Manipulation. In *Proc. 11th ACM Conference on Electronic Commerce*, 2010.

[18] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer and J. Rosenschein. Complexity of Unweighted Coalitional Manipulation Under Some Common Voting Rules. In *Proc. 21st International Joint Conference on Artificial Intelligence (IJCAI-2009)*, p.348-353, 2009.

[19] M. Zuckerman, A. Procaccia and J. Rosenschein. Algorithms for the Coalitional Manipulation Problem. In *Proc. 19th Symposium on Discrete Algorithms (SODA-2008)*, p.277-286, 2008.

Jessica Davies
University of Toronto
Toronto, Canada
Email: `jdavies@cs.toronto.edu`

George Katsirelos
Université Lille-Nord de France
CRIL/CNRS UMR8188
Lens, F-62307, France
Email: `gkatsi@gmail.com`

Nina Narodytska
NICTA and UNSW
Sydney, Australia
Email: `ninan@cse.unsw.edu.au`

Toby Walsh
NICTA and UNSW
Sydney, Australia
Email: `toby.walsh@nicta.com.au`

Fair Division under Ordinal Preferences: Computing Envy-Free Allocations of Indivisible Goods[1]

Sylvain Bouveret, Ulle Endriss, and Jérôme Lang

Abstract

We study the problem of fairly dividing a set of goods amongst a group of agents, when those agents have preferences that are ordinal relations over alternative bundles of goods (rather than utility functions) and when our knowledge of those preferences is incomplete. The incompleteness of the preferences stems from the fact that each agent reports their preferences by means of an expression of bounded size in a compact preference representation language. Specifically, we assume that each agent only provides a ranking of individual goods (rather than of bundles). In this context, we consider the algorithmic problem of deciding whether there exists an allocation that is possibly (or necessarily) envy-free, given the incomplete preference information available, if in addition some mild economic efficiency criteria need to be satisfied. We provide simple characterisations, giving rise to simple algorithms, for some instances of the problem, and computational complexity results, establishing the intractability of the problem, for others.

1 Introduction

The problem of fairly dividing a set of goods amongst a group of agents has recently started to receive increased attention in the AI literature [6, 10, 15, and others]. The study of the computational aspects of fair division, in particular, finds a natural home in AI; and fair division is immediately relevant to a range of applications in multiagent systems and electronic commerce.

To define an instance of a fair division problem, we need to specify the *type of goods* we want to divide, the nature of the *preferences* that individual agents hold, and the kind of *fairness criterion* we want to apply when searching for a solution. In this paper, we are concerned with *indivisible goods* that cannot be shared: each item needs to be allocated to (at most) one agent in its entirety. This choice renders fair division a combinatorial optimisation problem.

Regarding preferences, most work in fair division has made the assumption that the preferences of individual agents can be modelled as utility (or valuation) functions, mapping bundles of goods to a suitable numerical scale. This assumption is technically convenient, and it is clearly appropriate in the context of applications with a universal currency, rendering preferences interpersonally comparable. On the other hand, from a cognitive point of view, assuming such cardinal preferences may be questionable, as it requires an agent to be able to attach a number to every conceivable state of the world. In this paper, we make instead the (much weaker, and arguably more realistic) assumption that agents have *ordinal preferences*, and for the sake of simplicity we assume that these preferences are strict orders (which is a common assumption in fair division and voting). That is, each agent i is equipped with a preference relation $\succ_i$: $A \succ_i B$ expresses that agent i prefers the set of items A over the set of items B.

[1]This paper will also be presented at the 19th European Conference on Artificial Intelligence (ECAI-2010).

The third parameter is the criterion used to define what makes an allocation "fair". Restricting attention to ordinal preferences rules out some criteria. For instance, the Rawlsian (or egalitarian) approach to fairness ties social welfare to the welfare of the worst-off agent [16], which presupposes that interpersonal comparison of preferences is possible. Instead, we focus on the important criterion of *envy-freeness* [13]. An allocation is envy-free if each agent likes the bundle she received at least as much as any of the bundles received by others. Besides envy-freeness, a secondary criterion we shall be working with is *Pareto efficiency*, which also only requires ordinal preferences. An allocation is Pareto efficient if there is no other allocation making some agents better and no agent worse off.

A challenging aspect of devising methods for fair division with indivisible goods is its combinatorial nature [9]: the space of possible bundles grows exponentially in the number of goods. If there are 20 goods, each agent would, in principle, have to rank over one million bundles. This leads to the following dilemma: either we allow agents to express any possible preference relation on the set of all subsets of items, and end up with an exponentially large representation, as in the *descending demand procedure* of Herreiner and Puppe [14], which, while of great theoretical interest, is computationally infeasible as soon as the number of goods is more than a few units; or we restrict the range of preferences that agents may express. The latter is the path followed by Brams and King [8] and Brams et al. [7], who address the problem using the following approach: Elicit the preferences $\rhd_i$ of each agent i over *single* goods (the assumption is that this is a strict linear order) and induce an (incomplete) preference order $\succ_i$ over bundles as follows: for two bundles A and B, infer $A \succ_i B$ if there exists an injective mapping $f : (B \setminus A) \to (A \setminus B)$ such that $f(a) \rhd_i a$ for any $a \in B \setminus A$. That is, $\succ_i$ ranks A above B if a (not necessarily proper) subset of A *pairwise dominates* B, *i.e.*, if A is definitely preferred to B given the limited information (provided in the form of $\rhd_i$) available—under reasonable assumptions on how to "lift" preferences from single goods to bundles.[2] From a "computational" perspective, we might say that Brams and coauthors [7, 8] are using $\rhd_i$ as a *compact representation* of $\succ_i$. In fact, their approach coincides precisely with a simple fragment of the language of *conditional importance networks* (CI-nets), a compact graphical representation language for modelling ordinal preference relations that are monotonic [5]. The fragment in question are the so-called (exhaustive) *SCI-nets*, which we will define in Section 2.2.

We will model agent preferences using SCI-nets. Each SCI-net induces an incomplete preference order over bundles, with the intended interpretation that the agent's true preference order is some complete order that is consistent with the known incomplete order. This requires a nonstandard approach to defining fairness criteria. Here, again, we follow Brams and King [8] and Brams et al. [7] and define an allocation as being *possibly* envy-free if it is envy-free for *some* set of complete preferences that are consistent with the known incomplete preferences; and we say an allocation is *necessarily* envy-free if it is envy-free under *all* possible completions. We define possible and necessary Pareto efficiency accordingly.

The main question we study in this paper is then: *Given partially specified agent preferences, modelled in terms of SCI-nets, does there exist an allocation that is possibly (necessarily) envy-free?* As the allocation that simply disposes of all goods (*i.e.*, that does not assign any goods to the agents) is always both possibly and necessarily envy-free, to be interesting, this question needs to be asked under some efficiency requirements. In particular, we will ask whether there exists such allocations that are *complete* (*i.e.*, that allocate every item to some agent) or possibly (necessarily) Pareto efficient.

Some of our results are positive: we are able to provide simple characterisations of situations in which an allocation of the desired kind exists, and these characterisations

[2] The problem of lifting preferences over items to sets of items has been studied in depth in social choice theory [3]. Indeed, pairwise dominance is closely related to the axiom of "(weak) preference dominance" put forward by Sen in the context of work on formalising freedom of choice [17].

immediately suggest an algorithm for computing such an allocation. Other results are negative: deciding existence of an allocation of the desired kind (and thus also computing such an allocation) often turns out to be intractable.

The remainder of the paper is organised as follows. In Section 2 we define the model of fair division we shall be working with. In particular, this includes the language used to specify agent preferences and several fairness and efficiency criteria. In Section 3 we give the main results of this paper; namely, we show that while it is easy to compute possibly envy-free allocations that are also complete or possibly Pareto efficient, requiring necessary envy-freeness makes the problem NP-hard. The concluding Section 4 includes a short discussion of related work. (For lack of space, some proofs are only sketched.)

2 The model

Let $\mathcal{A} = \{1, \ldots, n\}$ be a finite set of *agents* and $\mathcal{G} = \{x_1, \ldots, x_m\}$ be a finite set of *goods* ($n \geq 2$ and $m \geq 1$). An *allocation* $\pi : \mathcal{A} \to 2^{\mathcal{G}}$ is a mapping from agents to sets of goods such that $\pi(i) \cap \pi(j) = \emptyset$ for any two distinct agents $i, j \in \mathcal{A}$; thus, goods are *indivisible*. An allocation π with $\pi(1) \cup \cdots \cup \pi(n) = \mathcal{G}$ is called *complete*.

In this section, we define criteria for identifying fair (or efficient) allocations of goods. These criteria will be defined in terms of the preferences of the individual agents over the bundles they receive.

2.1 Basic terminology and notation

A *strict partial order* is a binary relation that is irreflexive and transitive. A *linear order* is a strict partial order that is complete (*i.e.*, $X \succ Y$ or $Y \succ X$ whenever $X \neq Y$). A binary relation $\succ$ on $2^{\mathcal{G}}$ is *monotonic* if $X \supset Y$ implies $X \succ Y$. If $\succ$ (or $\rhd$) is a binary relation, then $\succeq$ (or $\trianglerighteq$) represents the reflexive closure of that relation (*i.e.*, $X \succeq Y$ if and only if $X \succ Y$ or $X = Y$). Given two binary relations R and R' on $2^{\mathcal{G}}$, we say that R' *refines* R if $R \subseteq R'$.

2.2 Preferences: SCI-nets

The preference relation of each agent $i \in \mathcal{A}$ is assumed to be a linear order $\succ_i^\star$ over the bundles (subsets of $\mathcal{G}$) she might receive. However, as argued above, eliciting $\succ_i^\star$ entirely would be infeasible; so we do not assume that $\succ_i^\star$ is fully known to us (or even to the agents themselves). Instead, for each agent i we are given a strict partial order $\succ_i$ representing our partial knowledge of $\succ_i^\star$, and the true preference of i is *some* complete refinement of $\succ_i$. The strict partial orders $\succ_i$ are generated from expressions of a suitable preference representation language. In this paper, we focus on the language of SCI-nets, *i.e.*, precondition-free CI-nets in which all compared sets are singletons [5]. We now introduce SCI-nets;[3] for full CI-nets see [5].

Definition 1 (SCI-nets) *An SCI-net $\mathcal{N}$ on $\mathcal{G}$ is a linear order on $\mathcal{G}$, denoted by $\rhd_{\mathcal{N}}$ (or simply $\rhd$, when the context is clear). A strict partial order $\succ$ on $2^{\mathcal{G}}$ complies with $\mathcal{N}$, if (i) $\succ$ is monotonic and (ii) $S \cup \{x\} \succ S \cup \{y\}$ for any x, y such that $x \rhd_{\mathcal{N}} y$ and any $S \subseteq \mathcal{G} \setminus \{x, y\}$. The preference relation $\succ_{\mathcal{N}}$ induced by $\mathcal{N}$ is the smallest strict partial order that complies with $\mathcal{N}$.*

As discussed earlier, $\succ_{\mathcal{N}}$ is the partial order we obtain when we lift the order $\rhd_{\mathcal{N}}$ on $\mathcal{G}$ to an order on $2^{\mathcal{G}}$ by invoking the principles of monotonicity and pairwise dominance, as

[3] What we call "SCI-nets" here were called "exhaustive SCI-nets" in [5].

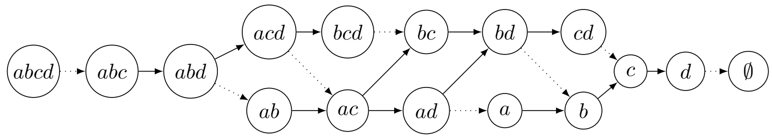


Figure 1: Preference relation induced by SCI-net $a \rhd b \rhd c \rhd d$. Dotted arcs are obtained by monotonicity; arcs obtained by transitivity are omitted.

proposed by Brams and coauthors [7, 8]. We can give yet another characterisation of $\succ_{\mathcal{N}}$, in terms of a utility function: Given SCI-net $\mathcal{N}$ and $A \subseteq \mathcal{G}$, for every $k \leq |A|$ we denote with $A^{\mathcal{N}}_{(k)}$ the k-most important element of A; *i.e.*, if $x \in A$ and $\#\{y \in A \mid y \succeq_{\mathcal{N}} x\} = k$ then $A^{\mathcal{N}}_{(k)} = x$. Given a vector $w = (w_1, ..., w_m) \in (\mathbb{R}^+)^m$ inducing the additive utility function $u_w : 2^{\mathcal{G}} \to \mathbb{R}$ with $u_w(A) = \sum_{x_i \in A} w_i$, and SCI-net $\mathcal{N} = x_{\theta(1)} \rhd \cdots \rhd x_{\theta(m)}$ (for some permutation θ of $\{1, \ldots, m\}$), we say that w and $\mathcal{N}$ are *compatible* if $w_{\theta(1)} > \cdots > w_{\theta(m)}$.

Proposition 1 (Dominance) *Given an SCI-net $\mathcal{N}$ and bundles $A, B \subseteq \mathcal{G}$, the following statements are equivalent:*

(1) $A \succ_{\mathcal{N}} B$

(2) *There exists an injective mapping $f : (B \setminus A) \to (A \setminus B)$ such that $f(a) \rhd_{\mathcal{N}} a$ for any $a \in B \setminus A$.*

(3) *There exists an injective mapping $g : B \to A$ such that $g(a) \trianglerighteq_{\mathcal{N}} a$ for all $a \in B$ and $g(a) \rhd_{\mathcal{N}} a$ for some $a \in B$.*

(4) *Either $A \supset B$, or the following three conditions are satisfied:*
- $|A| \geq |B|$*;*
- *for every $k \leq |B|$, $A^{\mathcal{N}}_{(k)} \trianglerighteq_{\mathcal{N}} B^{\mathcal{N}}_{(k)}$;*
- *there exists a $k \leq |B|$ such that $A^{\mathcal{N}}_{(k)} \rhd_{\mathcal{N}} B^{\mathcal{N}}_{(k)}$.*

(5) *For any w compatible with $\mathcal{N}$ we have $u_w(A) > u_w(B)$.*

The proof is simple; we omit it due to space constraints.

2.3 Criteria: envy-freeness and efficiency

For the fair division problems we study, each agent $i \in \mathcal{A}$ provides an SCI-net $\mathcal{N}_i$. This gives rise to a *profile* of strict partial orders $(\succ_{\mathcal{N}_1}, \ldots, \succ_{\mathcal{N}_n})$. For any such profile (whether it has been induced by SCI-nets or not), we can ask whether it admits a fair solution.

As our agents are only expressing *incomplete* preferences, the standard notions of envy-freeness and efficiency need to be adapted. For any solution concept, we may say that it is *possibly* satisfied (if *some* refinement of the preference profile to a profile of linear orders satisfies it) or that it is *necessarily* satisfied (if *all* such refinements do). The following definitions are a synthesis of those introduced by Brams and King [8] and Brams et al. [7].[4] While the results reported in the sequel apply to scenarios where each agent expresses her preferences in terms of an SCI-net, we state these definitions independently from the preference representation language in use.

Definition 2 (Modes of envy-freeness) *Given a profile of strict partial orders $(\succ_1, \ldots, \succ_n)$ on $2^{\mathcal{G}}$, an allocation π is called*

[4] Brams and coauthors [7, 8] use a different terminology: our necessarily (resp. possibly) envy-free allocations correspond to their allocations that are not envy-possible (resp. that are not envy-ensuring), and our necessarily (resp. possibly) Pareto efficient allocations correspond to their Pareto-ensuring (resp. Pareto-possible) allocations. We believe that applying the standard modalities of "necessary" and "possible" to basic fairness and efficiency criteria is the most systematic way of defining these notions.

(*i*) *possibly envy-free (PEF) if for every $i \in \mathcal{A}$ there exists a linear order $\succ_i^\star$ refining $\succ_i$ such that $\pi(i) \succ_i^\star \pi(j)$ for all $j \in \mathcal{A}$;*[5] *and*

(*ii*) *necessarily envy-free (NEF) if for every $i \in \mathcal{A}$ and every linear order $\succ_i^\star$ refining $\succ_i$ we have $\pi(i) \succ_i^\star \pi(j)$ for all $j \in \mathcal{A}$.*

Next we establish alternative characterisations of PEF and NEF allocations, which are more "computation-friendly".

Proposition 2 (PEF and NEF allocations) *Given $(\succ_1, \dots, \succ_n)$,*
- *π is NEF if and only if for all i, j, we have $\pi(i) \succ_i \pi(j)$;*
- *π is PEF if and only if for all i, j, we have $\pi(j) \not\succ_i \pi(i)$.*

Proof. The first point is obvious: π is NEF iff for every i and j, and every $\succ_i^\star$ refining $\succ_i$ we have $\pi(i) \succ_i^\star \pi(j)$, *i.e.*, iff $\pi(i) \succ_i \pi(j)$ holds for every i, j. For the second point, suppose $\pi(j) \succ_i \pi(i)$ for some i, j; then $\pi(j) \succ_i^\star \pi(i)$ holds for any refinement $\succ_i^\star$ of $\succ_i$, which implies that π is not PEF. The converse direction is less immediate, because the condition C_i: "for all j, $\pi(j) \not\succ_i \pi(i)$" only guarantees that for every i and every $j \neq i$ there exists an refinement $\succ_i^{\star j}$ of $\succ_i$ such that $\pi(i) \succ_i^{\star j} \pi(j)$. Assume that C_i holds and let the relation R_i be defined by $R_i = [\succ_i \cup \{(\pi(i), B) \mid B \neq \pi(i) \text{ and } B \not\succ_i \pi(i)\}]$. We show that R_i is acyclic. First, suppose there is an X such that $X R_i X$. Then by definition of R_i, $X \succ_i X$ ($X \neq \pi(i)$ by definition of R_i), which cannot be the case since $\succ_i$ is a well-defined strict order. Suppose now that there exists an irreducible cycle $X_1, \dots, X_q$ of length at least 2 such that $X_1 R_i X_2 \dots R_i X_q R_i X_{q+1} = X_1$, and $X_j \neq X_k$ for every $1 \leq j \neq k \leq q$. From the definition of R_i, for every $k \leq q$ we have either $X_k \succ_i X_{k+1}$ or ($X_k = \pi(i)$ and $X_{k+1} \not\succ_i \pi(i)$). Because $\succ_i$ is acyclic, there is at least one k such that $X_k = \pi(i)$. Because the cycle is irreducible, there is at most one k such that $X_k = \pi(i)$. Therefore, there is exactly one k such that $X_k = \pi(i)$; without loss of generality, let $k = 1$. We have (a) $X_2 \succ_i \pi(i)$ and (b) for every $j \neq 1$, $X_j \succ_i X_{j+1}$, that is, $X_1 = \pi(i) R_i X_2 \succ_i X_3 \succ_i \dots \succ_i X_q \succ_i X_1 = \pi(i)$. Because $\succ_i$ is transitive, $X_2 \succ_i X_3 \succ_i \dots \succ_i X_q \succ_i \pi(i)$ implies $X_2 \succ_i \pi(i)$, which contradicts (a). Therefore, R_i is acyclic, and its transitive closure $R_i^\star$ is a strict partial order. Take $\succ_i^\star$ to be any linear order refining $R_i^\star$. Because R_i contains $\succ_i$, $\succ_i^\star$ refines $\succ_i$; and for every j, because $\pi(j) \not\succ_i \pi(i)$, by construction of R_i we have that $\pi(i) R_i \pi(j)$, therefore also $\pi(i) \succ_i^\star \pi(j)$. □

Example 1 *Let $m = 5$, $n = 2$, $\mathcal{N}_1 = a \rhd b \rhd c \rhd d$ and $\mathcal{N}_2 = d \rhd c \rhd b \rhd a$. Consider the allocation π defined by $\pi(1) = \{a, d\}$ and $\pi(2) = \{b, c\}$. We have $\{b, c\} \not\succ_1 \{a, d\}$ and $\{a, d\} \not\succ_2 \{b, c\}$, therefore π is PEF. However, π is not NEF, but the allocation π' such that $\pi'(1) = \{a, b\}$ and $\pi'(2) = \{c, d\}$ is NEF (hence also PEF).*

Recall that for a profile of linear orders $(\succ_1^\star, \dots, \succ_n^\star)$ on $2^{\mathcal{G}}$, an allocation π' is said to *Pareto-dominate* another allocation π if $\pi'(i) \succeq_i^\star \pi(i)$ for all $i \in \mathcal{A}$ and $\pi'(j) \succ_j^\star \pi(j)$ for some $j \in \mathcal{A}$.

Definition 3 (Modes of dominance) *Given a profile of strict partial orders $(\succ_1, \dots, \succ_n)$ on $2^{\mathcal{G}}$ and two allocations π and π',*

(*i*) *π' possibly Pareto-dominates π if π' Pareto-dominates π for some profile of linear orders $(\succ_1^\star, \dots, \succ_n^\star)$ refining $(\succ_1, \dots, \succ_n)$.*

(*ii*) *π' necessarily Pareto-dominates π if π' Pareto-dominates π for all profiles of linear orders $(\succ_1^\star, \dots, \succ_n^\star)$ refining $(\succ_1, \dots, \succ_n)$.*

[5] The usual definition of envy-freeness only requires that each agent should be *at least as happy* with her share as with the share of anyone else, *i.e.*, that $\pi(i) \succeq_i^\star \pi(j)$ holds for all $i, j \in \mathcal{A}$. Here, $\pi(i) \succeq_i^\star \pi(j)$ and $\pi(i) \succ_i^\star \pi(j)$ are equivalent, because $\pi(i) \succeq_i^\star \pi(j)$ is equivalent to $\pi(i) \succ_i^\star \pi(j)$ or $\pi(i) = \pi(j)$, and of course we have $\pi(i) \neq \pi(j)$.

We get characterisations of possible and necessary Pareto dominance that are similar as those of Proposition 2.

Proposition 3 (Pareto dominance) *Given* $(\succ_1, \ldots, \succ_n)$,
• π' *necessarily Pareto-dominates* π *if and only if (a) for all* i*, we have* $\pi'(i) \succeq_i \pi(i)$ *and (b) for some* i*, we have* $\pi'(i) \succ_i \pi(i)$*;*
• π' *possibly Pareto-dominates* π *if and only if (c) for all* i*, we have* $\pi(i) \not\succ_i \pi'(i)$ *and (d) for some* i*, we have* $\pi(i) \not\succeq_i \pi'(i)$*.*

Proof. For the first point: (a) and (b) together clearly imply that π' necessarily dominates π. Conversely, assume π' necessarily dominates π. Then, by definition, π' Pareto-dominates π for all profiles of linear orders refining the partial orders. Exchanging the position of the two universal quantifiers immediately gives (a). Now, suppose that there is no i such that $\pi'(i) \succ \pi(i)$. Then for each i there is at least one refinement $\succ_i^\star$ such that $\pi(i) \succeq_i^\star \pi'(i)$. Let $P^\star = (\succ_1^\star, ..., \succ_n^\star)$. $P^\star$ refines $(\succ_1, ..., \succ_n)$, and for $P^\star$, π' does not Pareto dominate π, which contradicts the initial assumption, and we are done. The proof for the second point is similar. □

Definition 4 (Modes of efficiency) *Given a profile of strict partial orders* $(\succ_1, \ldots, \succ_n)$ *on* $2^{\mathcal{G}}$*, an allocation* π *is called*

(*i*) *possibly Pareto efficient (PPE) if there exists no allocation* π' *that necessarily Pareto-dominates* π*; and*

(*ii*) *necessarily Pareto efficient (NPE) if there exists no allocation* π' *that possibly Pareto-dominates* π*.*

Above concepts naturally extend to the case where preferences are modelled using a representation language, such as SCI-nets. For example, given a profile of SCI-nets $(\mathcal{N}_1, \ldots, \mathcal{N}_n)$, an allocation π is PEF if π is PEF for the profile $(\succ_{\mathcal{N}_1}, \ldots, \succ_{\mathcal{N}_n})$.

3 Computing envy-free allocations

In this section, we consider the problem of checking whether, for a given profile of SCI-nets, there exists an allocation that is (possibly or necessarily) envy-free, and that also satisfies a secondary efficiency requirement (in particular completeness).

3.1 Possible envy-freeness

We first ask whether a given profile of SCI-nets permits an allocation that is both PEF and complete. It turns out that there is a very simple characterisation of those profiles that do: all that matters is the number of distinct goods that are ranked at the top by one of the agents (in relation to the number of agents and goods). As will become clear in the proof of this result, the algorithm for computing a complete PEF allocation is also very simple.

Proposition 4 (PEF: general case) *If* n *agents express their preferences over* m *goods using SCI-nets and* k *distinct goods are top-ranked by some agent, then there exists a complete PEF allocation if and only if* $m \geq 2n - k$*.*

Proof. First, suppose there are $m \geq 2n - k$ goods. Executing the following protocol will result in a PEF allocation of $2n - k$ of those goods: (1) Go through the agents in ascending order, ask them to pick their top-ranked item if it is still available and ask them leave the room if they were able to pick it. (2) Go through the remaining $n - k$ agents in ascending order and ask them to claim their most preferred item from those still available. (3) Go through the remaining agents in descending order and ask them to claim their most preferred item from those still available. The resulting allocation is PEF, because for no agent the bundle of (one or two) goods(s) she obtained is pairwise dominated by any of the other bundles: she either is one of the k agents who received their top-ranked item or she was able to pick her second item before any of the agents preceding her in the first round were allowed to pick *their* second item. The remaining goods (if any) can be allocated to any of the agents; the resulting allocation remains PEF and is furthermore complete.

Second, suppose there are $m < 2n - k$ goods. Then, by the pigeon hole principle, there must be at least one agent i who receives an item that is not her top-ranked item $\widehat{x_i}$ and no further items beyond that. But then i will necessarily envy the agent who does receive $\widehat{x_i}$; thus, the allocation cannot be PEF. □

Example 2 *Let $m = 6$, $n = 4$, $\mathcal{N}_1 = a \rhd b \rhd c \rhd d \rhd e \rhd f$, $\mathcal{N}_2 = a \rhd d \rhd b \rhd c \rhd e \rhd f$, $\mathcal{N}_3 = b \rhd a \rhd c \rhd d \rhd f \rhd e$ and $\mathcal{N}_4 = b \rhd a \rhd c \rhd e \rhd f \rhd d$. We have $k = 2$ and $m \geq 2n - k$. Therefore, the algorithm returns a complete PEF allocation, namely, if we consider the agents in the order $1 > 2 > 3 > 4$: $\pi(1) = \{a\}$; $\pi(2) = \{d, f\}$; $\pi(3) = \{b\}$; $\pi(4) = \{c, e\}$. However, if f were unavailable, there would not be any complete PEF allocation.*

It is possible to show that Proposition 4 remains true if we require allocations to be PPE rather than just complete:

Proposition 5 (PPE-PEF: general case) *If n agents express their preferences over m goods using SCI-nets and k distinct goods are top-ranked by some agent, then there exists a PPE-PEF allocation if and only if $m \geq 2n - k$.*

Proof. First, any PPE allocation is complete; therefore, if there exists a PPE-PEF allocation, there also exists a complete PEF allocation. Conversely, if we refine the protocol given in the proof of Proposition 4 by allowing the last agent in round three to take all the remaining items at the end, then that protocol returns an allocation that is the product of *sincere choices* [8] by the agents for the sequence $1, 2, \ldots, n, n, \ldots, 1, \ldots, 1$. By Proposition 1 of Brams and King [8], any such allocation is PPE. □

The complexity of determining whether there exists an NPE-PEF allocation is still an open problem.

3.2 Necessary envy-freeness

Next, we turn attention to the problem of checking whether a NEF allocation exists, given a profile of SCI-nets. This is a considerably more demanding property than possible envy-freeness. For instance, it is easy to see that a necessary precondition for the existence of a complete NEF allocation is that all agents have distinct top-ranked goods (because any agent not receiving her top-ranked good *might* envy the agent receiving it, whatever other goods the two of them may obtain). Another necessary precondition is the following:

Lemma 6 (NEF: necessary condition) *If n agents express their preferences over m goods using SCI-nets and a complete NEF allocation does exist, then m must be a multiple of n.*

Proof. If m is not a multiple of n, then for an allocation to be complete, some agent i must receive fewer goods than another agent j. But *any* SCI-net (including that of i) is consistent with a linear order ranking any bundle of size k above any bundle of size less than k (for all k). Hence, such an allocation cannot be NEF. □

If there are as many goods as there are agents ($m = n$), then checking whether a complete NEF allocation exists is easy: it does if and only if all agents have distinct top-ranked goods. The next most simple case in which there is a chance that a complete NEF allocation might exist is when there are twice as many goods as agents ($m = 2n$). We now show that checking whether such an allocation exists (and computing it) is intractable:

Proposition 7 (NEF: general case) *If n agents express their preferences over m goods using SCI-nets, then deciding whether there exists a complete NEF allocation is* NP*-complete (even if $m = 2n$).*

Proof. Membership in NP is straightforward from Proposition 2. Hardness is proved by reduction from [X3C] (exact cover by 3-sets): given a set S of size $3q$, and a collection $C = \langle C_1, \ldots, C_n \rangle$ of subsets of S of size 3, does there exist a subcollection C' of C such that every element of S is present exactly once in C'?

Without loss of generality, we have $n \geq q$. To any instance $\langle S, C \rangle$ of [X3C] we associate the following allocation problem:

- $6n$ objects: $3n$ "dummy" objects $\{d_i^1, d_i^2, d_i^3 | i = 1, \ldots, n\}$, $3q$ "main" objects $\{m_i | i = 1, \ldots 3q\}$ and $3(n-q)$ "auxiliary" objects $\{o_i | i = 1, \ldots 3(n-q)\}$
- $3n$ agents $\{c_i, c_i', c_i'' | i = 1, \ldots, n\}$. c_i, c_i' and c_i'' are called agents of type i and if $C_i = \{j, k, l\}$, their preferences are expressed by the following SCI-nets:
  c_i: $d_i^1 \rhd d_i^2 \rhd d_i^3 \rhd m_j \rhd m_k \rhd m_l \rhd o_1 \rhd o_2 \rhd o_3 \rhd \ldots \rhd o_{3(n-q)-2} \rhd o_{3(n-q)-1} \rhd o_{3(n-q)} \rhd D \rhd M$;
  c_i': $d_i^2 \rhd d_i^3 \rhd d_i^1 \rhd m_k \rhd m_l \rhd m_j \rhd o_2 \rhd o_3 \rhd o_1 \rhd \ldots \rhd o_{3(n-q)-1} \rhd o_{3(n-q)} \rhd o_{3(n-q)-2} \rhd D \rhd M$;
  c_i'': $d_i^3 \rhd d_i^1 \rhd d_i^2 \rhd m_l \rhd m_j \rhd m_k \rhd o_3 \rhd o_1 \rhd o_2 \rhd \ldots \rhd o_{3(n-q)} \rhd o_{3(n-q)-2} \rhd o_{3(n-q)-1} \rhd D \rhd M$;
  where D (resp. M) means "all other dummy (resp. main) objects in any arbitrary order". m_j, m_k and m_l will be called "first-level objects" for c_i, c_i' and c_i''.

Suppose there exists an exact cover C' of C. C' contains exactly q subsets, therefore $C \setminus C'$ contains $n - q$ subsets. Let $f : C \setminus C' \to \{1, \ldots, n-q\}$ be an arbitrary bijective mapping. Define the allocation $\pi_{C'}$ as follows:

1. every agent gets her preferred dummy object d_i^j;
2. if $C_i \in C'$ then every agent of type i gets her preferred (first-level) main object (we will call these agents "lucky" ones);
3. if $C_i \notin C'$, every (unlucky) agent of type i gets an auxiliary object: c_i gets $o_{3f(i)-2}$, c_i' gets $o_{3f(i)-1}$, and c_i'' gets $o_{3f(i)}$.

Let us check that $\pi_{C'}$ is a complete allocation. Obviously, every dummy object is allocated (by point 1 above). Since C' is a cover, every main object is allocated as first-level object for some agent (by point 2 above). Since f is a bijective mapping, every auxiliary object is allocated (by point 3 above). Every agent gets exactly 2 objects, so no object can be allocated twice and the allocation is complete.

Now, we check that $\pi_{C'}$ is NEF. Since every agent receives her top-ranked object and another one, then by Proposition 1, checking that a does not necessarily envy b comes down to checking that $\pi(a)^a_{(2)} \rhd_a \pi(b)^a_{(2)}$ (hence comparing only the ranks of the worst objects in $\pi(a)$ and $\pi(b)$).

- For each lucky agent a, $rank(\pi(a)^a_{(2)}) = 4$. Each other agent gets either one main object or an auxiliary one. In both cases, the rank is obviously worse than 4, hence preventing a from possibly envying anyone else.
- The worst object received by any unlucky agent a of type i (say w.l.o.g. c_i) is her best one among the triple $\{o_{3f(i)-2}, o_{3f(i)-1}, o_{3f(i)}\}$. The worst object received by another agent of type i (say w.l.o.g. c_i') is another one from the same triple, that is

obviously worse for c_i. Hence no agent of type i can envy any other agent of the same type. Let b be an agent of type $j \neq i$ (lucky or not). b receives her top-ranked object d_j^k ($k \in \{1, 2, 3\}$), which is ranked worse than every auxiliary object for a, hence preventing a from possibly envying b.

Conversely, assume π is a complete NEF allocation. We first note that in π, every agent receives exactly two objects, among which her preferred object; therefore, in π the assignment of all dummy objects is completely determined.

Now, suppose there is an agent a that gets a main object $m(a)$ which is not among her first-level ones. Let m_j be one of her first-level objects. Then some agent b receives both m_j and a dummy object, both ranked higher than $m(a)$ in a's SCI-net. Hence a possibly envies b. From this we conclude that in π, the second object received by an agent is either a first-level object, or an auxiliary object.

Moreover, if an agent of type i (say, c_i) receives a first-level object, then the other two agents of type i must also receive a first-level object, for if it is not the case for one of them, she gets an auxiliary object and possibly envies c_i. Therefore, in π, for every i, either all agents of type i receive a first-level object, or none.

Finally, define C_π as the set of all C_i such that all the agents of type i receive a first-level object. π being complete, every main object must be given. Therefore, C_π is a cover of S. Because no main object can be given to two different agents, C_π is an exact cover of S.

The reduction being polynomial, this proves NP-hardness. □

Example 2, continued. *There is no complete NEF allocation, because m is not a multiple of n. If any one of the four agents is removed, again there is no complete NEF allocation, because there are two distinct agents with the same top object. If only agents 1 and 3 are left in, again it can be checked that there is no complete NEF allocation. If only agents 2 and 3 are left in, then there is a complete NEF allocation, namely $\pi(2) = \{a, d, e\}$, $\pi(3) = \{b, c, f\}$.*

Proposition 7 extends to the case of PPE allocations:

Proposition 8 (PPE-NEF: general case) *If n agents express their preferences over m goods using SCI-nets, then deciding whether there exists a PPE-NEF allocation is* NP*-complete (even if $m = 2n$).*

Proof. Given a sequence s of n agents, we can compute in polynomial time the allocation π_s that corresponds to the product of sincere choices according to s (which is PPE by Brams and King's characterisation [8]), and check in polynomial time that it is NEF. Thus s is a polynomial certificate for the problem, hence membership in NP.

For NP-hardness we can use the same reduction from [X3C]. Since every PPE allocation is complete, there is a PPE-NEF allocation only if there is a complete NEF allocation, hence only if there is an exact cover. Conversely, assume that there is an exact cover. Then the complete and NEF allocation obtained in the proof of Proposition 7 is also PPE by Brams and King's characterisation [8], since it is obtained by a sequence of sincere choices by agents (all the agents in sequence in the first round, then all the lucky agents, and finally all the unlucky agents). □

The hardness part of the proofs above extends to the case of NPE allocations (but we do not know whether the problem is still in NP).

Proposition 9 (NPE-NEF: general case) *If n agents express their preferences over m goods using SCI-nets, then deciding whether there exists an NPE-NEF allocation is* NP*-hard (even if $m = 2n$).*

Proof sketch. The idea of the proof (only sketched due to space constraints) is based on the same reduction from [X3C]: there is an NPE-NEF allocation only if there is a complete NEF allocation (since every NPE allocation is complete), hence only if there is an exact cover. Conversely, if there is an exact cover C', we can prove by contradiction that the allocation $\pi_{C'}$ is NPE. $\square$

In the special case of allocation problems with just two agents, a complete NEF allocation can be computed in polynomial time:

Proposition 10 (NEF: two agents) *If there are only two agents and both express their preferences using SCI-nets, then deciding whether there exists a complete NEF allocation is in* P.

We assume w.l.o.g. that the number of objects is even ($m = 2q$), for if not we know there cannot be any complete NEF allocation. We have an exact characterisation of NEF allocations:

Lemma 11 *Let $n = 2$ and π a complete allocation. π is NEF if and only if for every $i = 1, 2$ and every $k = 1, \ldots, q$, π gives agent i at least k of her $2k - 1$ most preferred objects.*

Proof. W.l.o.g., let the preference relation of agent 1 be given by $x_1 \rhd_1 x_2 \rhd_1 \ldots \rhd_1 x_{2q}$.

Assume that (1) for every $i = 1, 2$ and every $k = 1, \ldots, q$, π gives agent i at least k among $\{x_1, \ldots, x_{2k-1}\}$. Let $I = \{i, x_i \in \pi(1)\}$ and $J = \bar{I} = \{i, x_i \in \pi(2)\}$. Let $I = \{i_1, \ldots, i_q\}$ and $J = \{j_1, \ldots, j_q\}$ with $i_1 < \ldots < i_q$ and $j_1 < \ldots < j_q$. Let f be the following one-to-one mapping from I to J: for every $k = 1, \ldots, q$, $f(i_k) = j_k$. For every $k \leq q$, because of (1), we have that $i_k \leq 2k - 1$. Now, since $I \cap J = \emptyset$, $J \cap \{1, \ldots, 2k - 1\}$ contains at most $k - 1$ elements, therefore $j_k \geq 2k$, which implies $i_k < j_k$ and $x_{i_k} \rhd_1 x_{j_k}$. Thus f is a one-to-one mapping from I to J such that for every $i \in I$, agent 1 prefers x_i to $x_{f(i)}$. Symmetrically, we can build a one-to-one mapping g from J to I such that for every $j \in J$, agent 2 prefers x_j to $x_{g(j)}$. This implies that π is NEF.

Reciprocally, assume there exists a $k \leq q$ such that π gives agent 1 at most $k - 1$ objects among $\{x_1, \ldots, x_{2k-1}\}$. Then π gives agent 2 at least k objects among $\{x_1, \ldots, x_{2k-1}\}$. This implies that for any one-to-one mapping f from $\pi(1)$ to $\pi(2)$, there is some $i \leq k$ such that $x_{f(i)} \rhd_1 x_i$, therefore π is not NEF. Symmetrically, if there exists a $k \leq q$ such that π gives agent 2 at most $k - 1$ objects among her $2k - 1$ preferred objects, then π is not NEF. $\square$

Proof (Proposition 10). Let the preference relation of agent 1 be, w.l.o.g., $x_1 \rhd_1 x_2 \rhd_1 \ldots \rhd_1 x_{2q}$. From that SCI-net, we build the flow network shown in Figure 2 (edge labels x/y correspond to the edge lower bound x and capacity y).

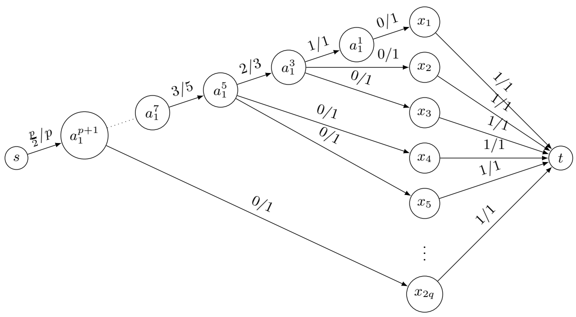


Figure 2: The flow network for one agent.

We build the same flow network for agent a_2 (nodes a_1^k are now called a_2^k) and identify, between the two networks, the nodes corresponding to the same objects, the source s, and the sink t.

We claim (but do not show due to lack of space) that there is an allocation π satisfying the condition stated in Lemma 11 if and only if there is a feasible flow of value p in the latter network.

The problem of finding a feasible flow in a network with lower bounds as well as capacities is known as the *circulation problem* and is known to be solvable in (deterministic) polynomial-time [12]. Hence the problem of deciding whether there exists a complete NEF allocation for a problem with two agents is in P. □

4 Conclusion and related work

We have studied the problem of computing envy-free allocations of indivisible goods, when agents have ordinal preferences over bundles of goods and when we only know their preferences over single items with certainty. Building on work from the ("non-computational") fair division literature, in particular the contributions by Brams et al. [7, 8], we have proposed a framework in which to study such questions, we have provided a number of alternative characterisations of the central concepts involved, and we have analysed the computational complexity of computing allocations of the desired kind.

We have been able to show that computing an allocation that is possibly envy-free is easy (whether paired with a requirement for completeness or possible Pareto efficiency). We have also been able to show that computing necessarily envy-free allocations is NP-hard (whatever the secondary efficiency requirement); only for problems with just two agents there is a polynomial (but non-trivial) algorithm. The complexity of finding envy-free allocation that are necessarily Pareto efficient is not fully understood at this stage. In particular, it is conceivable that deciding the existence of allocations that are both necessarily envy-free and necessarily Pareto efficient might not even be in NP; we leave the full analysis of this question to future work.

Future work should also seek to extend our results to *nonstrict* SCI-nets, where indifference between single goods is allowed. Problems that are still easy with strict SCI-nets, such as the existence of a complete PEF allocation, could conceivably become NP-complete. Intuitively, the more indifferences the agents express, the more complete the preference relations and the closer the notions of possible and necessary envy-freeness, which means that possible envy-freeness will be harder to guarantee.

Our work is part of a growing literature on computational aspects of fair division. In particular, complexity aspects of envy-freeness have been studied, for example in the works of Lipton et al. [15] and de Keijzer et al. [11], who address the problem of finding envy-free and complete (resp. Pareto efficient) allocations, when the agents have numerical additive preferences. Bouveret and Lang [6] also address the same problem, for various notions of efficiency, in a context where the agents have utilities expressed in compact form. However, none of these computational works concerns ordinal preferences, and none have considered possible or necessary satisfaction of fairness criteria. There is also a related stream of works on the *Santa Claus problem*, consisting in computing maxmin fair allocations (see *e.g.*, Bansal and Sviridenko [2], Bezáková and Dani [4], Asadpour and Saberi [1]). These works encode fairness by an egalitarian collective utility function and do not consider envy-freeness.

References

[1] A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *Proc. of STOC-2007*, 2007.

[2] N. Bansal and M. Sviridenko. The Santa Claus problem. In *Proc. of STOC-2006*, pages 31–40, 2006.

[3] S. Barberà, W. Bossert, and P. Pattanaik. Ranking sets of objects. In *Handbook of Utility Theory*, volume 2. Kluwer, 2004.

[4] I. Bezáková and V. Dani. Allocating indivisible goods. *SIGecom Exch.*, 5(3), 2005.

[5] S. Bouveret, U. Endriss, and J. Lang. Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods. In *Proc. of IJCAI-2009*, 2009.

[6] S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *JAIR*, 32:525–564, 2008.

[7] S. J. Brams, P. H. Edelman, and P. C. Fishburn. Fair division of indivisible items. *Theory and Decision*, 5(2):147–180, 2004.

[8] S. J. Brams and D. King. Efficient fair division—help the worst off or avoid envy? *Rationality and Society*, 17(4):387–421, 2005.

[9] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Preference handling in combinatorial domains: From AI to social choice. *AI Magazine*, 29(4):37–46, 2008.

[10] Y. Chevaleyre, U. Endriss, and N. Maudet. Allocating goods on a graph to eliminate envy. In *Proc. of AAAI-2007*, 2007.

[11] B. de Keijzer, S. Bouveret, T. Klos, and Y. Zhang. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In *Proc. of ADT-2009*, 2009.

[12] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[13] D. K. Foley. Resource allocation and the public sector. *Yale Economic Essays*, 7(1):45–98, 1967.

[14] D. Herreiner and C. Puppe. A simple procedure for finding equitable allocations of indivisible goods. *Social Choice and Welfare*, 19(2):415–430, 2002.

[15] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proc. EC-2004*. ACM Press, 2004.

[16] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[17] A. K. Sen. Welfare, preference and freedom. *J. of Econometrics*, 50(1–2):15–29, 1991.

Sylvain Bouveret
Onera-DTIM
2, avenue Édouard Belin. BP74025, 31055 Toulouse Cedex 4, France
Email: `sylvain.bouveret@onera.fr`

Ulle Endriss
Institute for Logic, Language and Computation, University of Amsterdam
Postbus 94242. 1090 GE Amsterdam, The Netherlands
Email: `ulle.endriss@uva.nl`

Jérôme Lang
LAMSADE, Université Paris-Dauphine
Place du Maréchal de Lattre de Tassigny. 75775 Paris Cedex 16, France
Email: `lang@lamsade.dauphine.fr`

A Maximin Approach to Finding Fair Spanning Trees

Andreas Darmann, Christian Klamler, and Ulrich Pferschy

Abstract

This paper analyzes the computational complexity involved in solving fairness issues on graphs, e.g., in the installation of networks such as water networks or oil pipelines. Based on individual rankings of the edges of a graph, we will show under which conditions solutions, i.e., spanning trees, can be determined efficiently given the goal of maximin voter satisfaction. In particular, we show that computing spanning trees for maximin voter satisfaction under voting rules such as approval voting or the Borda count is $\mathcal{NP}$-hard for a variable number of voters whereas it remains polynomially solvable for a constant number of voters.

1 Introduction

Spanning trees have first been used in connection with fair division problems in the 1970s for fairly assigning costs to individuals in a graph theoretical setting (Bird [3]). From this starting point, a huge body of literature has developed in recent years with a certain vicinity to Social Choice Theory, often axiomatically motivated (e.g., Bogomolnaia and Moulin [4], Dutta and Kar [11] and Kar [12]). In this paper we want to strengthen this link to Social Choice Theory by looking at the maximin voter satisfaction and analyzing the computational complexity of solution methods based on certain well-known social choice rules.

Many of the current papers use graphs to model certain networks, such as the installation of water or power networks, oil pipelines, road constructions, or links between different countries. Costs are assigned to the edges in such a graph and the goal is to connect all nodes (individuals, countries, etc.) at minimum total cost and fairly assign that cost to the nodes.

In this paper[1] we do not consider any monetary costs, be it because they are negligible or because they are covered by some external source (e.g., the state). Our approach is based on individuals' preferences over the edges of a graph and we analyze methods that - given those preferences - fairly, i.e., socially acceptably, install networks. The focus of our analysis, however, does not lie in the quality of the solution, i.e., in an axiomatic analysis of the solution methods, but in the computational complexity involved.

An example in that respect could be a village that has to install a sewage or water network or countries that need to agree on oil pipelines. Each homeowner or country needs to be connected but obviously there are many different ways to connect everyone. Mathematically the situation can be represented as a graph, i.e., the nodes are the homeowners and the edges are the connections between pairs of homeowners, and a solution is a spanning tree. The problem, however, is that homeowners might have different preferences over which connections (edges) should be used in the spanning tree. E.g. one homeowner might prefer a certain connection over another connection for environmental reasons, whereas another homeowner might just prefer any connection further away from his own garden to any connection that is closer to his garden. As we consider that costs are no issues here, the ordinal rankings over edges by those homeowners are the only inputs that can be used by any solution method.

[1] A major part of this work appeared in Darmann et al. [9, 10].

The quality of different solution methods based on social choice rules has been analyzed in a previous paper by Darmann et al. [8], extensive studies of social choice rules can be found in Brams and Fishburn [6], Nurmi [14] and Saari [19] among many others. The goal in this paper, however, is to look at the computational complexity involved in finding optimal spanning trees based on such solution methods, i.e., whether such solutions can be found in polynomial time or not.[2] Our main focus will be on methods using scores as in the Borda count or in approval voting and the basis for evaluating different solutions will be the maximin voter satisfaction (MMVS).
In a completely different setup, namely the consideration of different scenarios to represent uncertainty in Robust Optimization, closely related models of spanning tree problems were considered, e.g., in Aissi et al. [1] and Kouvelis and Yu [13]. While their works assign arbitrary numerical values as weights of the edges, we will consider the outcome of voting procedures to compare edges and trees.
An important differentiation arises from the number of voters considered in the problem, i.e., whether this number is fixed or not. Following the results of Aissi et al. [1], it is shown that for a fixed number of voters, solutions based on MMVS can be found in polynomial time. Things do change when the number of voters is variable, i.e., the number of voters is part of the input of the problem. This makes the problem significantly harder in the case of general edge weights as has been shown by Kouvelis and Yu [13]. However, as far as the $\mathcal{NP}$-hardness results are concerned, the simple structure of edge weights arising from the respective voting rules requires a completely different proof technique than their previously known results.
The contribution of this paper is to answer the questions of complexity posed by the application of voting rules from Social Choice Theory. We show that even under very simple voting structures such as approval voting, vote-against-t elections and choose-t elections for $t \geq 2$, MMVS is $\mathcal{NP}$-hard. Furthermore we show that MMVS is intractable for both dichotomous and multichotomous voter preferences. Moreover, irrespective of whether the voters' preferences are weak or strict orders on the edge set, MMVS under Borda voting is $\mathcal{NP}$-hard. Only for the two structurally most simple solution methods under consideration MMVS can be solved in polynomial time, namely for plurality voting and vote-against-1 election. In fact, our result settles the complexity status for any reasonable election process: If every voter is allowed to distinguish only one edge in a positive or negative sense the problem remains polynomially solvable. As soon as two or more edges receive an appraisal different from the remaining edges, the problem becomes $\mathcal{NP}$-hard.
The paper is structured as follows: We give the formal framework in Section 2 and then restate and discuss previous results for a fixed number of voters in Section 4. In Section 5 we keep the number of voters variable and prove our main results.

2 Preliminaries

In order to be able to express preferences, we give some basic definitions for relations; the terminology is adopted from Roberts [17].
A binary relation $\succsim \subseteq A \times A$ on a set A is called *complete* if $\forall a, b \in A,\ a \neq b,\ (a \succsim b$ or $b \succsim a)$. $\succsim$ is *reflexive* if $\forall a \in A,\ a \succsim a$. It is called *transitive* if $\forall a, b, c \in A,\ (a \succsim b$ and $b \succsim c) \Rightarrow a \succsim c$. Finally, $\succsim$ is called *asymmetric* if $\forall a, b \in A,\ a \succsim b \Rightarrow \neg(b \succsim a)$; and we call it *symmetric* if $\forall a, b \in A,\ a \succsim b \Rightarrow b \succsim a$. A relation is called *weak order* if it is complete, reflexive and transitive. A relation is called *strict order*, if it is complete, transitive and asymmetric.
Let $G = (V, E)$ be an undirected and connected graph. Let $n := |V|$ and τ be the set of spanning trees of G. For every voter i, $1 \leq i \leq k$, we are given a preference relation $\succsim_i$ on E. Unless otherwise stated, $\succsim_i$ is assumed to be a weak order on E, consisting of an asymmetric part $\succ_i$ and

[2] $\mathcal{P} \neq \mathcal{NP}$ is tacitly assumed throughout this paper.

a symmetric part $\sim_i$ respectively. The symmetric part $\sim_i$ of $\succsim_i$ induces a partition $E_1, E_2, ..., E_q$ of E, such that for all j, $1 \leq j \leq q$, we have $e \sim_i f$ for all $e, f \in E_j$. The sets E_j, $1 \leq j \leq q$, are called preference classes. In case $q = 2$ we call $\succsim_i$ dichotomous. If $q \geq 3$ the order $\succsim_i$ is called multichotomous. Furthermore, we refer to the k-tuple $\pi = (\succsim_1, \succsim_2, \ldots, \succsim_k)$ as a voter preference profile.

The basic concept used in this work is the one of voters' scoring functions, which can be understood as a generalization of the positional scoring procedures (for details concerning these procedures see Brams and Fishburn [6]).

Definition 2.1 *Let $1 \leq i \leq k$. We call a function $v_i : E \to \mathbb{N}_0$ voter i's scoring function, if*

1. *for all $e, f \in E$ $e \succsim_i f \;\Leftrightarrow\; v_i(e) \geq v_i(f)$, and*
2. $\max_{e \in E}\{v_i(e)\}$ *is bounded by a polynomial in n.*

Definition 2.2 *For $1 \leq i \leq k$ let v_i be voter i's scoring function. Voter i's score (or count) of tree $T \in \tau$ is $v_i(T) := \sum_{e \in T} v_i(e)$.*

Hence, voters' preferences on trees are assumed to be additively separable, i.e., there do not exist complementaries or synergies between the edges. Many scoring procedures can be embedded in the framework of voters' scoring functions. For example, approval voting (see Brams and Fishburn [5]), plurality voting (see Roberts [18]), vote-against-t elections (presented in Brams and Fishburn [6]) and Borda voting (see Brams and Fishburn [6] and Vorsatz [22]) can be formulated within this framework.[3]

Definition 2.3 *Let $1 \leq i \leq k$. For $e, f \in E$, $e \neq f$, let*

$$\delta_i(e,f) := \begin{cases} 2 & \text{if } e \succ_i f \\ 1 & \text{if } e \sim_i f \\ 0 & \text{otherwise.} \end{cases}$$

Then in Borda voting, voter i's scoring function is the Borda function $b_i : E \to \mathbb{N}_0$ defined by $b_i(e) := \sum_{f \in E \setminus \{e\}} \delta_i(e,f)$. For $e \in E$ we call $b_i(e)$ voter i's Borda[4] *count of edge e. Voter i's Borda count of tree $T \in \tau$ is $b_i(T) := \sum_{e \in T} b_i(e)$.*

In approval voting, for every voter i the set E is partitioned into a set $S_i \subseteq E$ of edges voter i approves of and a set $S_i^c := E \setminus S_i$ of edges voter i disapproves of.

Definition 2.4 *Let $1 \leq i \leq k$. In approval voting voter i's scoring function is the function $a_i : E \to \mathbb{N}_0$ with*

$$a_i(e) = \begin{cases} 1 & \text{if } e \in S_i \\ 0 & \text{if } e \in S_i^c. \end{cases}$$

[3] The use of scoring functions on edges to obtain scores for spanning trees has not received much attention yet in the literature. A general axiomatic analysis as surveyed by Barbera et al. [2] might help to provide support for such a use.

[4] If $\succ_i$ is a strict order on E, we have $b_i(e) = 2 \cdot |\{f \in E : e \succ_i f\}|$ for $e \in E$. Let $\hat{\delta}_i(e,f) := \frac{1}{2}\delta_i(e,f)$ for all $e, f \in E$, $e \neq f$, and let $\hat{b}_i(e) := \sum_{f \in E \setminus \{e\}} \hat{\delta}_i(e,f)$ for $e \in E$. Thus $\hat{b}_i(e) = |\{f \in E : e \succ_i f\}|$, and hence $\hat{b}_i(e)$ would define voter i's Borda count of edge e in the canonical way. Note that $b_i(e) > b_i(f) \Longleftrightarrow \hat{b}_i(e) > \hat{b}_i(f)$ for all $e, f \in E$, $e \neq f$, and $\sum_{e \in T_1} b_i(e) > \sum_{e \in T_2} b_i(e) \Longleftrightarrow \sum_{e \in T_1} \hat{b}_i(e) > \sum_{e \in T_2} \hat{b}_i(e)$ for all $T_1, T_2 \in \tau$. The function $\hat{b}_i$ however does not map from E into the set of non-negative integers but may take rational values as well. Since this causes some technical inconvenience (i.e., Theorem 4.1 cannot be applied directly), $\hat{b}_i$ is omitted in this work.

The function a_i is called voter i's approval function. Voter i's approval count of $T \in \tau$ is defined by $a_i(T) := \sum_{e \in T} a_i(e)$.

Choose-t elections and vote-against-t elections constitute two special cases of approval voting. A *choose-t election*[5] corresponds to approval voting subject to the requirement that for a fixed $t \in \mathbb{N}$ $|S_i| = t$ for $1 \leq i \leq k$. In this context, a choose-1 election is called *plurality voting*. Approval voting under the requirement that for a fixed $t \in \mathbb{N}$ $|S_i^c| = t$ for $1 \leq i \leq k$ is called *vote-against-t election*.

3 Problem formulation

With the above preliminaries we are now able to state the maximin voter satisfaction problem.

Definition 3.1 ***Maximin voter satisfaction problem (MMVS)***
Let $G = (V, E)$ be an undirected graph, let I be a set of voters and let π be a voter preference profile. For $i \in I$ let v_i be voter i's scoring function. The maximin voter satisfaction problem (MMVS) is the following problem:

$$\max_{T \in \tau} \min_{i \in I} v_i(T)$$

Maximizing the minimum of such concepts as utility, costs, time, etc. is a very common way to formalize the idea of fairness. Such a maximin approach to fairness can especially be found in the literature on networks, scheduling, etc. On the other hand, maximin fairness also has a certain link to fairness in Social Choice Theory, originally discussed decades ago by Rawls [16]. However, there are also many other approaches to formalize fairness based on proportionality, equitability, envy-freeness, etc. and used in areas such as mathematics and economics (Brams and Taylor [7], Thomson [21]).
From a completely different point of view the problem appears in the Operations Research literature in the context of Robust Optimization. One possibility to model an optimization problem under uncertainty is the consideration of different scenarios each of which induces different data for the problem. Maximizing the objective function for the worst-case scenario amounts to a maximin problem with voters corresponding to scenarios. In this context Aissi et al. [1] refer to an analogon of MMVS as *max-min spanning tree problem* while Kouvelis and Yu [13] use the terminology *absolute robust minimum spanning tree problem*. In this paper, however, the aim is to analyze the complexity of aggregating voters' opinions with the help of special types of voting procedures.

4 MMVS with a fixed number of voters

In this section the number k of voters is assumed to be a constant integer number. Likewise one could say that k is not regarded as a part of the input within this section. With this point of view MMVS is known to be solvable in polynomial time (see Aissi et al. [1]). We restate this result in the following theorem.

Theorem 4.1 *(Aissi et al. [1])*
MMVS can be solved in $\mathcal{O}(n^4 W^k \log W)$ time, where $W \in \mathbb{N}$ is an upper bound for the objective function value.

[5] In the literature, choose-t elections are also called t-approval voting (Peters et al. [15]) or vote-for-exactly-t procedures (Brams and Fishburn [6]).

Noting that for approval voting there is $W \leq n$ and for Borda voting $W \leq 2nm$, this theorem yields the following corollary.

Corollary 4.2 *MMVS under approval voting can be solved in* $\mathcal{O}(n^{4+k} \log n)$ *time. MMVS under Borda voting can be solved in* $\mathcal{O}(n^{4+k} m^k \log n)$ *time.*

However, for the special case of plurality voting MMVS can even be solved in linear time.

Proposition 4.3 *MMVS under plurality voting can be solved in* $\mathcal{O}(mk) = \mathcal{O}(m)$ *time.*

Proof. Given the graph $G = (V, E)$, let $E_1 := \{e \in E | v_i(e) = 1 \text{ for at least one } i,\ 1 \leq i \leq k\}$. If the subgraph $H = (V, E_1)$ is acyclic, then there obviously exists a spanning tree T of G such that $E_1 \subseteq T$ holds. In this case trivially $\max_{T \in \tau} \min_{i \in I} v_i(T) = 1$. If on the other hand H contains a cycle, then clearly there cannot exist a spanning tree T of G with $E_1 \subseteq T$. Thus for each spanning tree T of G there is an edge of E_1 that is not contained in T. Hence for each $T \in \tau$ we have $\min_{i \in I} v_i(T) = 0$ which yields $\max_{T \in \tau} \min_{i \in I} v_i(T) = 0$.
Calculating the set E_1 takes $\mathcal{O}(mk) = \mathcal{O}(m)$ time, the determination whether H is acyclic or not can be done in $\mathcal{O}(m)$ time. This proves the proposition. □

5 MMVS with a variable number of voters

In this section the number k of voters is not assumed to be constant but may vary instead, i.e., k is considered to be part of the input. This approach seems to make MMVS significantly harder. To be more precise, MMVS was shown to be strongly $\mathcal{NP}$-hard for arbitrary scoring functions by Kouvelis and Yu [13]. The question of the computational complexity of MMVS under the common voting rules such as approval voting, plurality voting, choose-t elections, vote-against-t elections and Borda voting is not answered by Kouvelis and Yu [13] though and to the authors' best knowledge has been open so far.
We improve upon the result of Kouvelis and Yu [13] and show that MMVS is $\mathcal{NP}$-hard even in case of very basic voting procedures. In particular, MMVS turns out to be $\mathcal{NP}$-hard even under the simple procedure of approval voting – that is, MMVS remains $\mathcal{NP}$-hard if the range of the voters' scoring functions is restricted to $\{0, 1\}$.[6] We also show that this result still holds if the number of approved or disapproved edges is some fixed $t \geq 2$ (choose-t elections and vote-against-t elections respectively for $t \geq 2$). Moreover, we can show that MMVS is $\mathcal{NP}$-hard under Borda voting. In contrast to these results, it can easily be shown that MMVS under plurality voting and vote-against-1 elections can be solved in polynomial time.
The key instrument used in the $\mathcal{NP}$-hardness proofs presented in this section is to reduce the $\mathcal{NP}$-complete monotone one-in-three 3SAT problem (Schaefer [20]) to the decision problem corresponding to MMVS.

Definition 5.1 ***Monotone one-in-three 3SAT problem (monotone 1-in-3SAT)***

GIVEN:	*A set X of variables and a collection C of clauses over X such that every clause is made up of exactly three positive literals.*
QUESTION:	*Is there a truth assignment for X such that every clause contains exactly one true literal?*

Remark. Note that in above definition every clause contains exactly three literals all of which must be positive. That is, in monotone 1-in-3SAT there are no negated literals. Therefore in monotone 1-in-3SAT the set X of variables corresponds to set of literals over X.

[6]Note that this implies and sharpens the strong $\mathcal{NP}$-hardness result of Kouvelis and Yu [13].

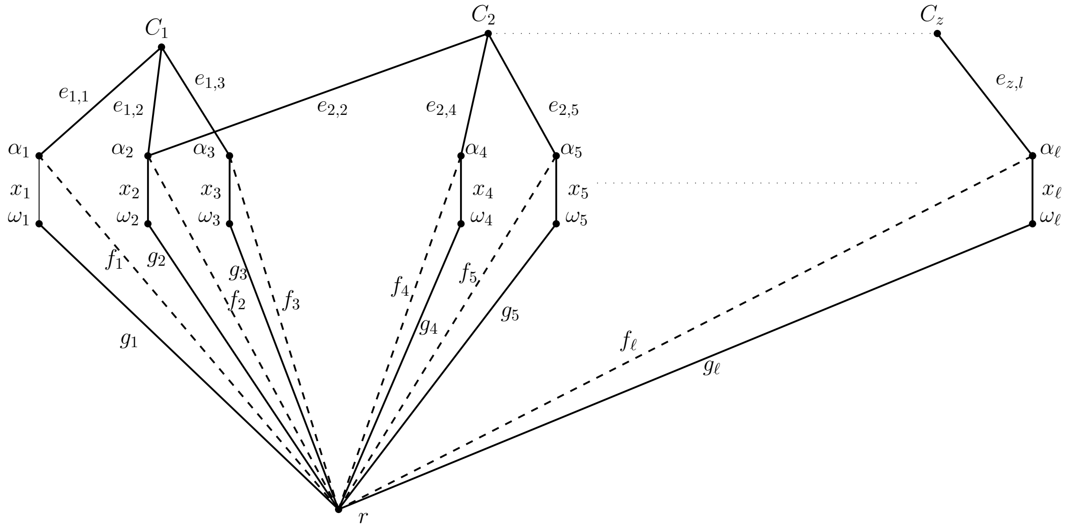


Figure 1: Undirected Graph G derived from instance U_1 of monotone 1-in-3SAT.

5.1 Approval voting and Borda voting

Our first result shows that MMVS is $\mathcal{NP}$-hard already for weak orders if the voters' scoring functions have the simple structure of approval functions.

Theorem 5.1 *Under approval voting MMVS is $\mathcal{NP}$-hard.*

Proof. We will polynomially transform an arbitrary instance of monotone 1-in-3SAT to an instance of MMVS with approval voting.
Let U_1 be an instance of monotone 1-in-3SAT with $X := \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_\ell\}$ being the set of variables (= literals) and $C := \{\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_z\}$ being a collection of clauses over X. W.l.o.g. we assume clause $\tilde{C}_1$ to contain the literals $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$. We construct the undirected graph $G = (V, E)$ by the following procedure (see Fig. 1):
Let $V = \emptyset$ and $E = \emptyset$. For each literal $\tilde{x}_j \in X$ add two nodes α_j and ω_j to V. For each clause $\tilde{C}_i \in C$ add node C_i to V. Add node r to V. Next for each literal $\tilde{x}_j \in X$

- add edge x_j to E connecting the nodes α_j and ω_j
- add edge f_j to E connecting α_j and r
- add edge g_j to E connecting ω_j and r
- if $\tilde{x}_j$ is contained in clause $\tilde{C}_i \in C$ add edge $e_{i,j}$ to E connecting the nodes C_i and α_j.

Note that $n = |V| = z + 2\ell + 1$ and $m = |E| = 3\ell + 3z$.
We now establish the voter preference profile π and the corresponding values of the voters' approval functions (see Table 1 and 2). First, we introduce voters χ_j, $1 \leq j \leq \ell$, whose approval functions are given by

$$a_{\chi_j}(e) = \begin{cases} 0 & \text{if } e \in \{x_j, f_j\} \\ 1 & \text{otherwise.} \end{cases}$$

χ_1		χ_2		χ_3		$\cdots$	χ_ℓ	
edge	a_{χ_1}	edge	a_{χ_2}	edge	a_{χ_3}		edge	a_{χ_ℓ}
x_1	**0**	x_2	**0**	x_3	**0**		x_ℓ	**0**
f_1	**0**	f_2	**0**	f_3	**0**		f_ℓ	**0**
$e_{1,1}$	**1**	$e_{1,1}$	**1**	$e_{1,1}$	**1**		$e_{1,1}$	**1**
$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**		$\vdots$	**1**
$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**		$\vdots$	**1**
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
x_ℓ	**1**	x_ℓ	**1**	x_ℓ	**1**		x_{l-1}	**1**

Table 1: Preference profile of voters χ_j, $1 \leq j \leq \ell$, and the values of the corresponding approval functions.

c_i		$c_i^{j_1}$		$c_i^{j_2}$		$c_i^{j_3}$		$c_i^{f_{j_1}}$		$c_i^{f_{j_2}}$		$c_i^{f_{j_3}}$	
edge	a_{c_i}	edge	$a_{c_i^{j_1}}$	edge	$a_{c_i^{j_2}}$	edge	$a_{c_i^{j_3}}$	edge	$a_{c_i^{f_{j_1}}}$	edge	$a_{c_i^{f_{j_2}}}$	edge	$a_{c_i^{f_{j_3}}}$
x_{j_1}	**0**	x_{j_1}	**0**	x_{j_2}	**0**	x_{j_3}	**0**	f_{j_1}	**0**	f_{j_2}	**0**	f_{j_3}	**0**
x_{j_2}	**0**	e_{i,j_2}	**0**	e_{i,j_1}	**0**	e_{i,j_1}	**0**	e_{i,j_1}	**0**	e_{i,j_2}	**0**	e_{i,j_3}	**0**
x_{j_3}	**0**	e_{i,j_3}	**0**	e_{i,j_3}	**0**	e_{i,j_2}	**0**		**1**		**1**		**1**
	1		**1**		**1**		**1**		**1**		**1**		**1**
other	**1**	other	**1**	other	**1**	other	**1**	other	**1**	other	**1**	other	**1**
edges	$\vdots$	edges	$\vdots$	edges	$\vdots$	edges	$\vdots$	edges	$\vdots$	edges	$\vdots$	edges	$\vdots$
	1		**1**		**1**		**1**		**1**		**1**		**1**

Table 2: Preference profile (and corresponding approval functions) derived from clause $\tilde{C}_i$ containing the literals $\tilde{x}_{j_1}, \tilde{x}_{j_2}, \tilde{x}_{j_3}$.

c_1		c_1^1		c_1^2		c_1^3		$c_1^{f_1}$		$c_1^{f_2}$		$c_1^{f_3}$	
edge	a_{c_i}	edge	$a_{c_i^{j_1}}$	edge	$a_{c_i^{j_2}}$	edge	$a_{c_i^{j_3}}$	edge	$a_{c_i^{f_{j_1}}}$	edge	$a_{c_i^{f_{j_2}}}$	edge	$a_{c_i^{f_{j_3}}}$
x_1	**0**	x_1	**0**	x_2	**0**	x_3	**0**	f_1	**0**	f_2	**0**	f_3	**0**
x_2	**0**	$e_{1,2}$	**0**	$e_{1,1}$	**0**	$e_{1,1}$	**0**	$e_{1,1}$	**0**	$e_{1,2}$	**0**	$e_{1,3}$	**0**
x_3	**0**	$e_{1,3}$	**0**	$e_{1,3}$	**0**	$e_{1,2}$	**0**	$e_{1,2}$	**1**	$e_{1,1}$	**1**	$e_{1,1}$	**1**
$e_{1,1}$	**1**	$e_{1,1}$	**1**	$e_{1,2}$	**1**	$e_{1,3}$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**
$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**
$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**
$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**	$\vdots$	**1**
x_ℓ	**1**	x_ℓ	**1**	x_ℓ	**1**	x_ℓ	**1**	x_ℓ	**1**	x_ℓ	**1**	x_ℓ	**1**

Table 3: Preference profile derived from clause $\tilde{C}_1$ which is made up of the literals $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$.

The rest of the voter preference profile is established as follows. Let a clause $\tilde{C}_i \in C$ contain the literals $\tilde{x}_{j_1}, \tilde{x}_{j_2}, \tilde{x}_{j_3}$ – which means node α_y and node C_i are adjacent, $y \in \{j_1, j_2, j_3\}$. Add seven voters denoted by c_i, $c_i^{j_1}$, $c_i^{j_2}$, $c_i^{j_3}$, $c_i^{f_{j_1}}$, $c_i^{f_{j_2}}$ and $c_i^{f_{j_3}}$ to π. Voter c_i assigns value 0 to the edges $x_{j_1}, x_{j_2}, x_{j_3}$ and value 1 to all other edges. Voter c_i^y, $y \in \{j_1, j_2, j_3\}$ assigns value 1 to all edges but to x_y and to the edges $e_{i,u}$ with $u \in \{j_1, j_2, j_3\}$, $u \neq y$, which get value 0. And voter $c_i^{f_y}$, $y \in \{j_1, j_2, j_3\}$, assigns value 0 to the edges f_y and $e_{i,y}$, and assigns value 1 to all the other edges (see Table 2). To illustrate the voter preference profile π, an example is given in Table 3 with the preferences and approval functions of the seven voters corresponding to clause $\tilde{C}_1$ which is made up of the literals $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$.
Having treated all clauses in the way just described the voter preference profile is made up of $k := \ell + 7z$ voters. Note that the instance of MMVS under approval voting defined by $G = (V, E)$, π and the corresponding approval functions can be constructed in polynomial time (with respect to the size of U_1).

Claim 1. There exists a truth assignment for X such that each clause in C contains exactly one true literal if and only if there exists a $T \in \tau$ such that for all p, $1 \leq p \leq k$, $a_p(T) \geq n - 2$ holds.

Proof of Claim 1.
"⇒": For a satisfying truth assignment t_S let S be the set of literals set "TRUE" under t_S. Create tree T as follows. Set $T = \emptyset$. For all $\tilde{x}_j \in S$:

- add x_j and g_j to T
- add $e_{i,j}$ to T for all i, $1 \leq i \leq z$, for which edge $e_{i,j} \in G$

For all $\tilde{x}_j \in X \setminus S$, i.e., literals set "FALSE" in t_S, add f_j and g_j to T. Summarizing, we get for $1 \leq j \leq \ell$ the following four properties:

1. $g_j \in T$
2. $x_j \in T \Leftrightarrow \tilde{x}_j$ is set "TRUE" under t_S
3. $x_j \in T \Leftrightarrow e_{i,j} \in T$ for all $i : e_{i,j} \in G$
4. $x_j \in T \Leftrightarrow f_j \notin T$

Since t_S constitutes a satisfying truth assignment, each node C_i, $1 \leq i \leq z$, is connected to node r in T. Obviously, all other nodes of V are connected to r in T as well and thus T is connected. Because of $|T| = |S| + z + \ell + (\ell - |S|) = z + 2\ell$ we get $|T| = n - 1$ and hence the subgraph T is a tree. Due to $|T| = n - 1$ and property 4. we get $a_{\chi_j}(T) = n - 2$ for all $j \in \{1, 2, \ldots, \ell\}$.
As above, let clause $\tilde{C}_i$ be made up of the literals $\tilde{x}_{j_1}, \tilde{x}_{j_2}, \tilde{x}_{j_3}$. The fact that exactly one of the literals $\tilde{x}_{j_1}, \tilde{x}_{j_2}, \tilde{x}_{j_3}$ is set "TRUE" under t_S means exactly one of the edges $x_{j_1}, x_{j_2}, x_{j_3}$ is contained in T. Together with $|T| = n - 1$ this yields $a_{c_i}(T) = n - 2$. Let us now consider the voters $c_i^{j_1}$, $c_i^{j_2}$, $c_i^{j_3}$: W.l.o.g. we may assume that $\tilde{x}_{j_1}$ is set "TRUE" under t_S. Thus $x_{j_1} \in T$, $x_{j_2} \notin T$, $x_{j_3} \notin T$. Due to property 3. we hence get $e_{i,j_1} \in T$, $e_{i,j_2} \notin T$, $e_{i,j_3} \notin T$. This implies

$$a_{c_i^{j_y}}(T) = n - 2$$

for all $y \in \{j_1, j_2, j_3\}$.[7] Finally, properties 3. and 4. yield $a_{c_i^{f_y}}(T) = n - 2$ for all $y \in \{j_1, j_2, j_3\}$.

[7]Clearly, assuming that instead of $\tilde{x}_{j_1}$ either $\tilde{x}_{j_2}$ or $\tilde{x}_{j_3}$ is set "TRUE" under t_S yields $a_{c_i^{j_y}}(T) = n - 2$ as well.

"$\Leftarrow$": Let now Q be a spanning tree with $a_p(Q) \geq n-2$ for all p, $1 \leq p \leq k$. Thus for each voter p in our voter preference profile at most one edge e with $a_p(e) = 0$ is contained in Q. Hence because of voters χ_j the edges x_j and f_j cannot both be contained in Q, $1 \leq j \leq \ell$. Analogously due to voters c_i, $1 \leq i \leq z$, for any clause $\tilde{C}_i$ made up of some literals $\tilde{x}_{j_1}, \tilde{x}_{j_2}, \tilde{x}_{j_3}$ at most one of the edges x_{j_1}, x_{j_2}, x_{j_3} is contained in Q. Next we show that for $1 \leq j \leq \ell$

$$x_j \in Q \Leftrightarrow e_{i,j} \in Q$$

holds for all i with $e_{i,j} \in G$.
Assume $x_j = x_{j_1} \in Q$ and let node C_i be adjacent to nodes $\alpha_{x_{j_1}}$, $\alpha_{x_{j_2}}$ and $\alpha_{x_{j_3}}$ (i.e., in our monotone 1-in-3SAT instance clause $\tilde{C}_i$ is again made up of the literals $\tilde{x}_{j_1}, \tilde{x}_{j_2}, \tilde{x}_{j_3}$). Because of voter $c_i^{j_1}$ we have $e_{i,j_2} \notin Q$ and $e_{i,j_3} \notin Q$. Note that the degree of node C_i equals three and thus $e_{i,j_1} \in Q$ since otherwise C_i would be isolated. Thus $x_j \in Q$ implies $e_{i,j} \in Q$ for all i such that $e_{i,j} \in G$.
On the other hand, let $e_{i,j_1} \in Q$ for some i, $1 \leq i \leq z$, and some j_1, $1 \leq j_1 \leq \ell$. Now $a_{c_i^{j_2}}(Q) \geq n-2$ implies $e_{i,j_3} \notin Q$ and $a_{c_i^{j_3}}(Q) \geq n-2$ implies $e_{i,j_2} \notin Q$. In other words, node C_i is a leaf. Due to voter $c_i^{f_{j_1}}$ we have $f_{j_1} \notin Q$. If there is no u, $1 \leq u \leq z$, $u \neq i$, such that $e_{u,j_1} \in Q$ then it is easy to see that x_{j_1} must be contained in Q since otherwise nodes r and C_i would not be connected. If such an edge e_{u,j_1} is contained in Q, then as a consequence of

$$a_{c_u^{j_1}}(Q) \geq n-2$$

node C_u must be a leaf as well and thus the same argument applies. Hence $x_j \in Q \Leftrightarrow e_{i,j} \in Q$ holds for all i with $e_{i,j} \in G$, $1 \leq j \leq \ell$.
But since node C_i is a leaf, $1 \leq i \leq z$, for each such node there is exactly one j, $1 \leq j \leq \ell$, such that both x_j and $e_{i,j}$ are contained in Q. In other words, the truth assignment $t_{\tilde{S}}$ defined by letting $\tilde{S} := \{\tilde{x}_j | x_j \in Q\}$ be the whole set of literals set "TRUE" under $t_{\tilde{S}}$ is a satisfying truth assignment for the considered instance of monotone 1-in-3SAT. This proves the claim. $\Diamond$
Claim 1 implies that any arbitrary instance of monotone 1-in-3SAT can be reduced to an instance of MMVS under approval voting. As stated before, the instance of MMVS under approval voting can be constructed in polynomial time. Thus it is proven that monotone 1-in-3SAT polynomially transforms to MMVS under approval voting. $\square$

Remark. Note that in case of dichotomous preferences the sets of optimal solutions of MMVS under approval voting and of MMVS under Borda voting obviously coincide.[8] Thus from Theorem 5.1 it follows that, given weak preference orders, MMVS under Borda voting is $\mathcal{NP}$-hard as well.[9]

Proposition 5.2 *MMVS under Borda voting is $\mathcal{NP}$-hard.*

Since dichotomous preferences over the edges induce approval functions in a natural way, it follows from Theorem 5.1 that MMVS is $\mathcal{NP}$-hard for any dichotomous preferences already. Furthermore it can easily be shown that MMVS is $\mathcal{NP}$-hard in the cases of multichotomous preferences as well.

Corollary 5.3 *Let $\pi = (\succsim_1, \succsim_2, \ldots, \succsim_k)$ be a voter preference profile such that $\succsim_i$ is multichotomous for all $1 \leq i \leq k$. Then MMVS is $\mathcal{NP}$-hard.*

[8]Therefore the general result shown in [22] that, given dichotomous preferences, Borda's method and approval voting are equivalent, applies for MMVS as well.

[9]We can show that this result still holds if the voters' preferences are strict orders. I.e., MMVS is also $\mathcal{NP}$-hard if the voters' scoring functions are bijections to $\{1, 2, \ldots, m\}$ and thus no two edges receive the same value.

Proof. Let $q > 2$ be the number of preference classes. Create a graph H from the graph $G = (V, E)$ used in the proof of Theorem 5.1 by concatenating a path p of length $q-2$ to node r. Let $n := |V|$ and $m := |E|$. We now derive from the profile π used in the proof of Theorem 5.1 a profile $\tilde{\pi}$ on the edges of graph H such that $\tilde{\pi}$ consists of q preference classes in two steps. Firstly, we derive from π a preference profile π_1 on G such that every voter i who disapproves of three edges in π is in π_1 replaced by three voters who disapprove of two edges only. Secondly, using the profile π_1 and path p, we assign the edges of H to the preference classes.
In order to get π_1, a voter γ who disapproves of edges $\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ is replaced by the following three voters: voter γ_1 who disapproves of edges $\{\varepsilon_1, \varepsilon_2\}$, voter γ_2 who disapproves of edges $\{\varepsilon_2, \varepsilon_3\}$ and voter γ_3 who disapproves of edges $\{\varepsilon_1, \varepsilon_3\}$.
Denote the preference classes that make up $\tilde{\pi}$ by A_{ij}, $0 \le j \le q-1$, for all voters i, $1 \le i \le k$. Let these preference classes be such that each edge in A_{ij} be strictly preferred to each edge in $A_{ij'}$ for $0 \le j' < j \le q-1$. Now for each voter i let $A_{i0} := \{e \in E | a_i(e) = 0\}$ and let $A_{i(q-1)} := \{e \in E | a_i(e) = 1\}$ according to π_1. Note that $|A_{i0}| = 2$ and $|A_{i(q-1)}| = m-2$. Assign the $q-2$ edges of the path p to the classes A_{ij}, $1 \le j \le q-2$, in an arbitrary way such that each of these classes contains exactly one edge. Assume Borda voting is being used. Then for every i, voter i's Borda values of the edges are given as follows:

$$b_i(e) = \begin{cases} 2q + (m-3) & \text{if } e \in A_{i(q-1)} \\ 2(j+1) & \text{if } e \in A_{ij},\ 1 \le j \le q-2 \\ 1 & \text{if } e \in A_{i0} \end{cases}$$

Obviously each edge of the path p must be contained in a spanning tree of H. Since $2q+(m-3) > 2$ the following two decision problems (D1) and (D2) are equivalent:

(D1)	GIVEN:	Graph G and preference profile π.
	QUESTION:	Is there a spanning tree T of G such that $a_i(T) \ge n-2$ for all i, $1 \le i \le k$?
(D2)	GIVEN:	Graph H and preference profile $\tilde{\pi}$.
	QUESTION:	Is there a spanning tree T_1 of H such that $b_i(T_1) \ge (n-2)(2q+(m-3)) + \sum_{j=1}^{q-2} 2(j+1)$ for all i, $1 \le i \le k$?

Thus, the corollary follows. □

5.2 Vote-against-t elections and choose-t elections

As a consequence of the proof of Theorem 5.1 in the previous subsection, for any integer $t \ge 2$ MMVS under vote-against-t elections is $\mathcal{NP}$-hard as well. The proof of this result uses the same approach as the one of Theorem 5.1 and is therefore omitted in this paper.

Corollary 5.4 *Let $t \in \mathbb{N}$, $t \ge 2$. Under vote-against-t elections MMVS is $\mathcal{NP}$-hard.*

It is worth noting that the above corollary does not hold for MMVS under vote-against-1 elections. In this case a solution of MMVS can be found in the following way: Remove from the considered graph G all edges e that have $v_i(e) = 0$ for at least one voter i. If the remaining graph is connected, then the objective function value is $n-1$, otherwise it is $n-2$. This observation yields the following statement.

Proposition 5.5 *Under vote-against-1 elections MMVS can be solved in $\mathcal{O}(mk)$ time.*

From Proposition 4.3 we know that MMVS under plurality voting, i.e., choose-1 elections, can be solved within the polynomial time bound of $\mathcal{O}(mk)$. By a reduction from the classical 3SAT problem we can show that, in contrast, MMVS under choose-t elections is $\mathcal{NP}$-hard for each fixed $t \geq 2$. Therefore, as for vote-against-t elections, with the step from $t = 1$ to $t = 2$ the computational complexity of MMVS under choose-t elections jumps from polynomial time solvable to $\mathcal{NP}$-hard.

Theorem 5.6 *MMVS under choose-t elections is $\mathcal{NP}$-hard for every fixed $t \geq 2$.*

6 Conclusion

We have considered the maximin voter satisfaction problem under both the scenarios that the number of voters is constant and may vary. It is known from Aissi et al. [1] that MMVS is polynomially solvable when the number of voters is fixed. The main contribution of this paper has dealt with the question of computational complexity of MMVS in the case of a *variable number of voters*. We improve upon an $\mathcal{NP}$-hardness result of Kouvelis and Yu [13] for general scoring functions by showing that, for a varying number of voters, MMVS is $\mathcal{NP}$-hard under very basic voting rules already. In particular, we have shown that MMVS is computationally intractable under approval voting, vote-against-t elections and choose-t elections for $t \geq 2$. We have proven that the problem is $\mathcal{NP}$-hard both in the cases of dichotomous voter preferences and multichotomous voter preferences. Furthermore, MMVS under Borda voting is $\mathcal{NP}$-hard, irrespective of the underlying voter preferences constituting weak orders or strict orders on the set of edges. Among the voting methods under consideration MMVS has turned out to be polynomially solvable only for the structurally most simple ones: plurality voting and vote-against-1 elections. Thus, when allowing each voter to approve or disapprove of more than one edge, the computational complexity of MMVS jumps from polynomial time solvable to $\mathcal{NP}$-hard. In these $\mathcal{NP}$-hard cases however, it is natural to ask if MMVS is fixed-parameter tractable when parametrized by the number of voters. Following the approach of Aissi et al. [1], we can show that MMVS is fixed-parameter tractable under choose-t elections and under vote-against-t elections, for each $t \geq 2$. Whether or not MMVS under Borda voting is fixed-parameter tractable remains an interesting open question.

References

[1] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximating min-max (regret) versions of some polynomial problems. In D.Z. Chen and D.T. Lee, editors, *COCOON*, volume 4112 of *Lecture Notes in Computer Science*, pages 428–438, Taipei, 2006. Springer, Berlin.

[2] S. Barbera, W. Bossert, and P.K. Pattanaik. Ranking sets of objects. In *Handbook of Utility Theory*, volume 2, pages 893–977. Springer, Berlin, 2004.

[3] C.G. Bird. On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 6:335–350, 1976.

[4] A. Bogomolnaia and H. Moulin. Sharing the cost of a minimal cost spanning tree: beyond the folk solution. *Games and Economic Behavior*, 69(2):238–248, 2010.

[5] S.J. Brams and P.C. Fishburn. *Approval Voting*. Birkhäuser, Boston, 1983.

[6] S.J. Brams and P.C. Fishburn. Voting procedures. In K.J. Arrow, A.K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. Elsevier, Amsterdam, 2002.

[7] S.J. Brams and A.D. Taylor. *Fair Division: From cake-cutting to dispute resolution.* Cambridge University Press, Cambridge, 1996.

[8] A. Darmann, C. Klamler, and U. Pferschy. Finding socially best spanning trees. To appear in *Theory and Decision.*

[9] A. Darmann, C. Klamler, and U. Pferschy. Maximizing the minimum voter satisfaction on spanning trees. *Mathematical Social Sciences*, 58:238–250, 2009.

[10] A. Darmann, C. Klamler, and U. Pferschy. A note on maximizing the minimum voter satisfaction on spanning trees. *Mathematical Social Sciences*, 60:82–85, 2010.

[11] B. Dutta and A. Kar. Cost monotonicity, consistency and minimum cost spanning tree games. *Games and Economic Behavior*, 48:223–248, 2004.

[12] A. Kar. Axiomatization of the shapley value on minimum cost spanning tree games. *Games and Economic Behavior*, 38(2):265–277, 2002.

[13] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications.* Kluwer Academic Publishers, Boston, 1997.

[14] H. Nurmi. *Voting Procedures under Uncertainty.* Springer, Berlin, 2007.

[15] H. Peters, S. Roy, and T. Storcken. Manipulation under k-approval scoring rules. METEOR Research Memorandum 08/056, University of Maastricht, 2008.

[16] J. Rawls. *A Theory of Justice.* Clarendon Press, Oxford, 1972.

[17] F.S. Roberts. *Measurement theory, with applications to Decision Making, Utility and the Social Sciences.* Addison-Wesley, Boston, 1979.

[18] F.S. Roberts. Characterizations of the plurality function. *Mathematical Social Sciences*, 21(2):101–127, April 1991.

[19] D.G. Saari. *Basic Geometry of Voting.* Springer, Berlin, 1995.

[20] T.J. Schaefer. The complexity of satisfiability problems. In *STOC '78: Proceedings of the 10th annual ACM symposium on Theory of Computing*, pages 216–226, San Diego, 1978. ACM, New York.

[21] W. Thomson. Fair allocation rules. Working Paper No.539, 2007. University of Rochester.

[22] M. Vorsatz. Scoring rules on dichotomous preferences. *Social Choice and Welfare*, 31(1):151–162, 2008.

Andreas Darmann, Christian Klamler
Institute of Public Economics
University of Graz
8010 Graz, Austria
Email: {andreas.darmann,christian.klamler}@uni-graz.at

Ulrich Pferschy
Institute of Statistics and Operations Research
University of Graz
8010 Graz, Austria
Email: pferschy@uni-graz.at

Approximate Judgement Aggregation

Ilan Nehama

Abstract

We analyze judgement aggregation problems in which a group of agents independently votes on a set of complex propositions that has some interdependency constraint between them (e.g., transitivity when describing preferences). We generalize the current results by studying approximate judgment aggregation. That is, we relax the main two constraints assumed in the current literature. We relax the consistency constraint by measuring the fraction of inputs for which an aggregation mechanism returns an inconsistent result and we relax the independence constraint by defining a measure for the dependance of the aggregation for an issue on the votes on other issues. We define the problem of measuring the impact of such small relaxation on the class of satisfying aggregation mechanisms and raise the question of whether there exists an agenda for which the expansion of this class is non-trivial. We show that the recent works for preference aggregation of Kalai and Mossel fit into this framework. We prove that, as in the case of preference aggregation, in the case of a subclass of premise-conclusion agendas, the set of satisfying aggregation mechanisms does not extend non-trivially when relaxing the constraints.
A corollary from our result for the xor premise-conclusion agenda is a generalization of the classic result for local property testing of linearity of boolean functions.

Keywords: approximate aggregation, discursive dilemma, premise-conclusion agenda, inconsistency index, dependency index

1 Introduction

Assume a committee of three referees needs to review a paper for a conference. Each of the referees judges the paper individually for originality and for quality (assumed to be pass/fail questions) and approves the paper only if it passes both criteria. The three referees cast their votes simultaneously and we assume no strategic behavior on their behalf. Now assume that both the first and second referee think that the paper is original enough and both the second and third referee think it stands in the quality standards of the conference. Then we have that although a minority of the committee (one out of three) thinks the paper should pass, for each issue separately there is a supporting majority (two out of three). This discrepancy between the majority vote on premises (quality and originality) and the majority vote on the conclusion (pass) was presented by Kornhauser and Sager in 1986[13] and was later named 'The Doctrinal Paradox'. Such discrepancy phenomena can happen when the 'accepted opinions' is restricted to be other sets as well (e.g., Condorcet Paradox for preference aggregation) and is the subject of a growing body of works in economics, political science, philosophy, law, and other related disciplines. (A survey of this field can be found in [14])

Abstract aggregation can be formalized in the following way. There is a committee of n individuals (also called voters) that needs to decide on m boolean issues (that is, each question has exactly two possible answers `True` and `False`[1]). Each individual holds an **opinion** which is an answer for each of the issues. We denote the answer of the $i^{\underline{th}}$ voter

The research was supported by a grant from the Israeli Science Foundation (ISF).

[1] There is some literature also on aggregating non-boolean issues, e.g., [20] and [7], but this is outside the scope of this paper.

for the $\mathrm{j}^{\underline{\mathrm{th}}}$ issue by X_i^j and the vector of all opinions in the committee (called **profile**) by $X \in (\{0,1\}^m)^n$ (For the ease of presentation we will identify `True` with 1 and `False` with 0). Like in the example above, not all opinions are acceptable (one cannot accept a non-original paper). We assume a non-empty set $\mathbb{X}$ of $\{0,1\}^m$ called the **agenda** is given. The opinions in $\mathbb{X}$ are called the **consistent** opinions and only these opinions are held by voters[2]. For instance the **conjunction agenda**, which is the agenda described in the example, is defined to be the set $\{000, 010, 100, 111\}$[3]. Another example is the **preference agenda**. In this agenda the consistent opinions represent the linear orders over a set of candidates $\{c_1, c_2, \ldots, c_s\}$ and the issues are the $\binom{s}{2}$ pair-wise comparisons between candidates[4,5].

An aggregation mechanism is a function that defines for any profile the **aggregated opinion** ($F : (\{0,1\}^m)^n \to \{0,1\}^m$). There are two desired properties for aggregation mechanism, independence and consistency. **Independence** states that the aggregated opinion on the $\mathrm{j}^{\underline{\mathrm{th}}}$ issue, $F^j(X)$ depends solely on the opinions on that issue X^j. **Consistency** of the aggregation mechanism states that whenever all the members of the committee hold consistent opinions, i.e., $X \in \mathbb{X}^n$, F returns a consistent opinion as well, i.e., $F(X) \in \mathbb{X}$.

For instance, issue-wise majority satisfies independence but also, as can be seen in the accept-paper example, might lead to an inconsistent result for the conjunction agenda and hence does not satisfy consistency. Similarly, the Condorcet Paradox[15] shows that, for the preference agenda, issue-wise majority might lead to an inconsistent result. The natural question is whether one can find other aggregation mechanisms that satisfy independence and consistency. Answering this question, Arrow's theorem[1] shows that (under mild and natural constraint[6]) the only aggregation mechanisms that satisfy independence and consistency are the dictatorships. For other agendas one can find similar theorems that characterize the class of consistent and independent aggregation mechanism to be a very small and unnatural class. For instance, for the conjunction agenda (under the same mild and natural constraint[6]) the only aggregation mechanisms that satisfy independence and consistency are the oligarchies (The oligarchy of a coalition S returns for each issue `True` if all voters in S voted `True` for that issue). In a recent work Dokow and Holzman ([5],[6]) proved a generalization of these results characterizing the set of consistent and independent aggregation mechanism for several large families of agendas.

Lately there is a series of works coping with impossibility results in Social Choice using approximations (e.g., [11] and [10]). The version of approximation we define in this work is studying independence aggregation mechanisms that are almost consistent in the sense that they return a consistent aggregated opinion for the vast majority of the inputs [7]. We quantify being almost consistent by defining the **inconsistency index**.

Definition 1.1 (Inconsistency Index).
For an agenda $\mathbb{X}$ and an aggregation mechanism F for that agenda, the inconsistency index

[2] For instance those might be the legal opinions, logic consistent opinions, or rational according to other criteria so one can assume that any 'reasonable' individual should hold only consistent opinions.

[3] I.e., the third bit ia a conjunction of the first two.

[4] For instance, for $s = 3$ the issues are '$c_1 \overset{?}{\geqslant} c_2$', '$c_2 \overset{?}{\geqslant} c_3$', and '$c_3 \overset{?}{\geqslant} c_1$' and the consistent opinions are $\{001, 010, 100, 110, 101, 011\}$.

[5] A related model that can be found in the literature is 'Judgement Aggregation'. In this model the issues are logical propositions over a set of variables and a consistent opinion is an assignment to these variables (so not every combination of truth values for the proposition is achievable). From our perspective the model we describe is more general since we allow any agenda. Dokow and Holzman[5] proved that the two models are equivalent in the sense that each set of consistent opinions can be described using a proposition set (although not uniquely).

[6] Pareto - Whenever all the voters hold the same opinion, this is the aggregated opinion.

[7] In most of this work we leave the independence constraint intact and relax the consistency constraint. However, as we show in section 6, one can relax the independence constraint as well and get similar results.

is defined to be the probability to get an inconsistent result.

$$IC^{\mathbb{X}}(F) = \Pr\left[F(X) \notin \mathbb{X} \mid X \in \mathbb{X}^n\right]$$

assuming uniform probability over the inputs.

This definition assumes a uniform distribution over the opinions for each voter and that voters draw their opinions independently (**Impartial Culture Assumption**). This assumption, while certainly unrealistic, is the natural choice in this kind of work and is discussed further in section 2.

In addition we use the usual Hamming distance between two aggregation mechanisms ($d^{\mathbb{X}}(F,G) = \Pr\left[F(X) \neq G(X) \mid X \in \mathbb{X}^n\right]$) and derive from it a distance between an aggregation mechanism and a collection of aggregation mechanisms ($d^{\mathbb{X}}(F,\mathcal{G}) = \min\limits_{G\in\mathcal{G}} d^{\mathbb{X}}(F,G)$).

It is easy to see that when F is close to G and G is consistent, F is close to being consistent, i.e., $IC(F)$ is small. Our main question is whether there are other aggregation mechanisms that are close to being consistent (Formally, $IC^{\mathbb{X}}(F) \leqslant d^{\mathbb{X}}(F,G))$).

For the preference agenda, recent works of Kalai[12] and Mossel[17] prove such bounds

Theorem ([12])**.** *There exists an absolute constant K such that the following holds: For any $\epsilon > 0$ and any aggregation mechanism F for the preference agenda over 3 candidates that satisfies: F is balanced*[8]*, F is independent, and $IC(F) < K\epsilon$, there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F,G) < \epsilon$.*

In this paper we prove similar theorems for a family of agendas: premise-conclusion agendas in which every issue is either a premise or a conclusion of at most two premises. In a premise-conclusion agendas the issues are divided into two types: premises and conclusions. Each conclusion j is characterized by a boolean function Φ_j over the premises and an opinion is consistent if the answers to the conclusion issues are attained by applying the function Φ_j on the answers to the premise issues.

$$\mathbb{X} = \left\{x \in \{0,1\}^m \mid x^j = \Phi^j(\text{premises}) \quad \text{for every conclusion issue } j\right\}$$

For instance the conjunction agenda is a premise-conclusion agenda with two premises and one conclusion and we mark this by notating the agenda as $\langle A, B, A \wedge B\rangle$. In some cases the division to premises and conclusion might be non-unique. For instance for the xor agenda $\mathbb{X} = \{001, 010, 100, 111\}$ one can define it as a premise-conclusion agenda both as $\langle A, B, A \oplus B\rangle$ and as $\langle A, A \oplus C, C\rangle$.

The main result of this paper is:

Theorem (Theorem 4.1)**.** *For any $\epsilon > 0$ and $n \geqslant 1$, there exists $\delta = poly\left(\frac{1}{n}, \epsilon\right)$, such that for every premise-conclusion agenda in which each issue is either a premise, or a conclusion of at most two premises, if F is an aggregation mechanism for $\mathbb{X}$ over n voters satisfying independence and $IC(F) < \delta$, then there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F,G) < \epsilon$.*
Moreover, one may take $\delta = Cn^{-2}\epsilon^5$ for some absolute constant C.

From the theorem it follows that, whenever the inconsistency index of is small enough ($O(n^{-7})$), the distance to the class of independent consistent mechanisms is small too (poly(n)-small. I.e., bounded from above by one over a polynomial of n) and hence proves that for these agendas the class of satisfying aggregation mechanisms does not expand much when relaxing the consistency constraint.

[8]For every pair of candidates, a and b, it holds that the probability that F ranks a above b is exactly 1/2.

The general statement follows easily from the analysis of three basic cases: The conjunction agenda $\langle A, B, A \wedge B\rangle$, the xor agenda $\langle A, B, A \oplus B\rangle$, and the id agenda $\langle A, A\rangle$

We use two different techniques in the proofs. For the conjunction agenda we study influence measures[9] of voters on the issue-aggregating functions and for the xor agenda we use Fourier analysis of the issue-aggregating functions.[10]

Notice the question of approximate aggregation has a close relation to the field of local property testing. In this field we query a function at a small number of (random) points testing for a global property (In our case the property is being a consistent independent aggregation mechanism). And indeed one can see our characterization for the xor agenda as a generalization of the result of Blum, Luby, and Rubinfeld ([3], [2]) that shows that a function f that passes the linearity test with high probability[11] is close to linear.

An open question is whether one can find such bounds for any agenda or whether there exists an agenda for which the class of aggregation mechanisms that satisfy consistency and independence expands non trivially when we relax the consistency and independence constraints.

We proceed to describe the structure of the current paper. In Section 2 we describe the formal model of aggregation mechanisms. In section 3 we give the two main examples we deal with, preference aggregation and premise-conclusion aggregation. In section 4 we state the motivation to deal with approximate aggregation, we describe the known results for preference approximate aggregation by Kalai and Mossel and state our main result for approximate aggregation for premise-conclusion agendas. In sections 5 we outline the proof of the main theorem. In section 6 we define a measure that relaxes the independence constraint and show that any result for approximate aggregation for independent aggregation mechanisms (which is the case in our main theorem) can be translated to the more general definition relaxing both constraints. Section 7 concludes.

2 The model

We define the model similarly to [5] (which is Rubinstein and Fishburn's model [20] for the boolean case)

We consider a **committee** of n individuals that needs to decide on m issues. An **opinion** is a vector $x = (x_1, x_2, \ldots, x_m) \in \{0,1\}^m$ denoting an answer to each of the issues. An opinion **profile** is a matrix $X \in (\{0,1\}^m)^n$ denoting the opinions of the committee members so an entry X_i^j denotes the vote of the $i^{\underline{th}}$ voter for the $j^{\underline{th}}$ issue, the $i^{\underline{th}}$ row of it X_i states the votes of the $i^{\underline{th}}$ individual on all issues, and the $j^{\underline{th}}$ column of it X^j states the votes of each of the individuals on the $j^{\underline{th}}$ issue. In addition we assume that an **agenda** $\mathbb{X} \in \{0,1\}^m$ of the **consistent** opinions is given.

The basic notion in this field is an **aggregation mechanism** which is a function that returns an **aggregated opinion** (not necessarily consistent) for every profile $\left(F : (\{0,1\}^m)^n \to \{0,1\}^m\right)$[12].

An aggregation mechanism satisfies **Independence** (and we say that the mechanism is **independent**) if for any two consistent profiles X and Y and an issue j, if $X^j = Y^j$ (all individuals voted the same on the $j^{\underline{th}}$ issue in both profiles) then $(F(X))^j = (F(Y))^j$ (the aggregated opinion for the $j^{\underline{th}}$ issue is the same for both profiles). This means that F

[9] Both the known influence (Banzhaf power index) and a new measure we define:The ignorability of a voter.

[10] The proof for the id case is trivial.

[11] which is equivalent to that the aggregation mechanism for $\langle A, B, A \oplus B\rangle$ that uses f for each of the issues has small inconsistency index.

[12] We define the function for all profiles for simplicity but we are not interested in the aggregated opinion in cases one of the voters voted an inconsistent opinion.

satisfies independence if one can find m boolean functions $f^1, f^2, \ldots, f^m : \{0,1\}^n \to \{0,1\}$ s.t. $F(X) \equiv \big(f^1(X^1), f^2(X^2), \ldots, f^m(X^m)\big)$. Notice this property is a generalization of the IIA property for social welfare functions (aggregation mechanism for the preference agenda) so a social welfare function satisfies IIA iff it satisfies independence as defined here (when the issues are the pair-wise comparisons). An independent aggregation mechanism satisfies **systematicity** if $F(X) = \big\langle f(X^1), \ldots, f(X^m)\big\rangle$ for some issue aggregating function, i.e., all issues are aggregated using the same function. We will use the notation $\big\langle f^1, f^2, \ldots, f^m\big\rangle$ for the independent aggregation mechanism that aggregates the $\text{j}^{\underline{\text{th}}}$ issue using f^j.

The main measure we study in this paper is the **inconsistency index** $IC^{\mathbb{X}}(F)$ of a given aggregation mechanism F and a given agenda $\mathbb{X}$ (as defined in the introduction). This measure is a relaxation of the **consistency** criterion that is usually assumed in current works[13]. We defining this measure by

$$IC^{\mathbb{X}}(F) = \Pr\left[F(X) \notin \mathbb{X} \mid X \in \mathbb{X}^n\right]$$

assuming uniform distribution of the profiles. In cases the context is clear we omit the agenda and notate it by $IC(F)$.

This definition includes two major assumptions on the opinion profile distribution. First, we assume the voters pick their opinions independently and from the same distribution. Second, we assume a uniform distribution over the (consistent) opinions for each voter(**Impartial Culture Assumption**). The uniform distribution assumption, while certainly unrealistic, is the natural choice for proving 'lower bounds' on $IC(F)$. That means, proving results of the format 'Every aggregation mechanism of a given class has inconsistency index of at least ...'. In particular, the lower bound, up to a factor δ, applies also to any distribution that gives each preference profile at least a δ fraction of the probability given by the uniform distribution. Note that we cannot hope to get a reasonable bound result for every distribution. For instance, since for every aggregation mechanism we can take a distribution on profiles for which it returns a consistent opinion.

2.1 Boolean Functions

Since this work deals with binary functions (for aggregating issues), we need to define several notions for this framework as well. To ease the presentation, throughout this paper we will identify `True` with 1 and `False` with 0 and use logical operators on bits and bit vectors (using entry-wise semantics).

Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function. f is the **oligarchy** of a coalition S if it is of the form: $f(x) = \prod_{i \in S} x_i$. This means that f returns 1 if all the members of S voted 1. We denote by **`Olig`** the class of all 2^n oligarchies. Two special cases of oligarchies are the constant 1 function which is the oligarchy of the empty coalition and the dictatorships which are oligarchies of a single voter.

f is a **linear** function if is it of the form $f(x) = \bigoplus_{i \in S} x_i$ for some coalition S[14]. This means that f returns 1 if an even number of the members of S voted 1. We denote by **`Lin`** the class of all 2^n linear functions. Two special cases of linear functions are the constant 1 function which is the xor function over the empty coalition and the dictatorships which are xor of a single voter.

We say that f satisfies the **Pareto** criterion is $f(\bar{0}) = 0$ and $f(\bar{1}) = 1$[15]. I.e., when all

[13] F satisfies consistency if $IC(F) = 0$.

[14] An equivalent definition is: $\forall x, y :\ f(x) + f(y) = f(x + y)$ when the addition is in $\mathbb{Z}_2$ and $\mathbb{Z}_2^n$, respectively.

[15] In the literature this criterion is sometimes referred to as Unanimity, e.g., in [14]. We choose to follow [6] and refer to it as Pareto to distinguish between it and the unanimity function which is the oligarchy of $\{1, 2, \ldots, n\}$.

the individuals voted unanimously 0 then f should return 0 and similarly for the case of 1.

We define two different measures for the influence of an individual on a function $f : \{0,1\}^n \to \{0,1\}$. Both definitions use the uniform distribution over $\{0,1\}^n$ (which is consistent with the assumption we have on the profile distribution).

- The **influence**[16] of a voter i on f is defined to be the probability that he can flip the result by changing his vote.

$$I_i(f) = \Pr[f(x) \neq f(x \oplus e_i)]$$

($x \oplus e_i$: e_i = the $\mathrm{i}^{\underline{\mathrm{th}}}$ elementary vector. It is equivalent to flipping the $\mathrm{i}^{\underline{\mathrm{th}}}$ bit $0 \leftrightarrow 1$)

- The (zero-)**ignorability** of a voter i on f is defined to be the probability that f returns 1 when i voted 0.

$$P_i(f) = \Pr[f(x) = 1 \mid x_i = 0)]$$

(We did not find a similar index defined in the voting literature or in the cooperative games literature).

In addition we define a distance function over the boolean functions. The distance between two functions $f, g : \{0,1\}^n \to \{0,1\}$ is defined to be the probability of getting a different result (normalized Hamming distance). $d(f,g) = \Pr\left[f(x) \neq g(x)\right]$. From this measure we will derive a distance from a function to a set of functions by $d(f, \mathcal{G}) = \min_{g \in \mathcal{G}} d(f,g)$

One more notation we are using in this paper is x_J for a binary vector $x \in \{0,1\}^n$ and a coalition $J \subseteq \{1, 2, \ldots n\}$ for notating the entries of x that correspond to J.

3 Agenda Examples

A lot of natural problems can be formulated in the framework of aggregation mechanisms. In this paper we concentrate on two examples: (strict) preference aggregation and the class of premise-conclusion agendas. Among other interesting natural agendas in this framework that were studied one can find the equivalence agenda[9] and the membership agenda [21][16].

3.1 Preference Aggregation

Aggregation of preferences is one of the oldest aggregation frameworks studied. In this framework there are s candidates and each individual holds a full strict order over them. We are interested in Social Welfare Functions which are functions that aggregate n such orders to an aggregated order. As seen in [18] and [4], this problem can be stated naturally in our framework by defining $\binom{s}{2}$ issues[17].

3.2 Premise-conclusion agendas

In a premise-conclusion agendas the issues are divided into two types: k premises and $(m-k)$ conclusions. The conclusion issues are boolean functions over the k premises, $\Phi : \{0,1\}^k \to \{0,1\}^{m-k}$. An opinion is consistent if the answers to the conclusion issues are attained by applying the function Φ on the premise issues.

$$\mathbb{X} = \left\{ x \in \{0,1\}^m \mid x^j = \Phi^j(x_1, \ldots, x_k) \quad j = k+1, \ldots, m \right\}$$

In this paper we prove results to the following two specific premise-conclusion agendas. We later derive results to a general family of premise-conclusion agendas.

[16] In the simple cooperative games regime, this is also called the Banzhaf power index of player i in the game f.

[17] The issue $\langle i, j \rangle$ (for $i < j$) represents whether an individual prefers c_i over c_j.

3.2.1 Conjunction Agenda (Doctrinal Paradox Agenda)

In the (2-premises) conjunction agenda $\langle A, B, A \wedge B\rangle$ there are three issues to decide on and the consistency criterion is defined to be that the third issue is a conjunction of the first two. A common description of the problem is of a group of judges or jurors that should decide whether a defendant is liable under a charge of breach of contract. Each of them should decide on three issues: whether the contract was valid (p), whether there was a breach (q) and whether the defendant is liable (r). In their decision making they are constrained by the legal doctrine that the defendant is only liable if the contract was valid and if there was indeed a breach ($r \iff (p \wedge q)$).

3.2.2 Xor Agenda

Similarly, in the (2-premises) xor agenda $\langle A, B, A \oplus B\rangle$ there are three issues to decide on and the consistency criterion is defined to be that the third issue is `True` if the first two answers are equal. An equivalent way to define this agenda is constraining the number of `True` answers to be odd.

4 Approximate Aggregation Results

In this paper we are interested in studying whether relaxing the consistency constraint, i.e., taking $IC(F) = \Pr\left[F(X) \notin \mathbb{X} \mid X \in \mathbb{X}^n\right]$ to be small (while restricting ourselves to independent aggregation mechanisms), extends non-trivially the set of satisfying aggregation mechanisms, i.e. entails that $d(F, \mathcal{C}(\mathbb{X})) = \min_{G \in \mathcal{C}(\mathbb{X})} \Pr\left[F(X) \neq G(X) \mid X \in \mathbb{X}^n\right]$ is small (taking $\mathcal{C}(\mathbb{X})$ to be the class of aggregation mechanisms that satisfies consistency and independence). More specifically we are interested in theorems of the following form (For a given agenda $\mathbb{X}$):

Theorem. *For any $\epsilon > 0$ and $n \geqslant 1$, there exists $\delta = \delta\left(\frac{1}{n}, \epsilon\right)$, such that if F is an aggregation mechanism for $\mathbb{X}$ over n voters satisfying independence and $IC(F) < \delta$, then there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F, G) < \epsilon$.*

Notice that such a theorem can be trivially satisfied by $\delta(\epsilon, n) = 0$. We seek better bounds. Particulary, we are interested that whenever ϵ is small (e.g., $\frac{1}{\mathrm{poly}(n)}$), then so is δ. E.g., taking δ to be *poly* $\left(\frac{1}{n}, \epsilon\right)$.

We find the motivation for dealing with the field of approximate aggregation in three different disciplines.

- The consistent characterization are often regarded as 'impossibility results' in the sense that they 'permit' a very restrictive set of aggregation mechanisms. (e.g., Arrow's theorem tells us that there is no 'reasonable' way to aggregate preferences). Extending this theorems to approximate aggregation characterizations sheds light on these impossibility results by relaxing the constraints.

- The questions of Aggregation Theory have often roots in Philosophy, Law, or Political Science. Results on approximate aggregations support the discussion that started in the works of Arrow[1] and Kornhauser and Sager[13] and searches for ways to deal with scenarios in which it is needed to aggregate such opinions.

- The CS field of Local Property Checking of Boolean Functions deals with the problem of deciding whether a given function has a given property (e.g., linearity) or whether it is 'far' from any object having the property. The works in the field consider randomized algorithms that query the function at points of their choice, and seek algorithms which query the function at relatively few points (For a survey of this field see [8]). The question of checking locally for a global property is very close to the framework of approximate aggregation (whether there exists an aggregation mechanism that is far from the set of independent and consistent aggregation mechanisms but still does not fail for most profiles). And indeed, the analysis of such randomized algorithm deals with very similar expressions to the inconsistency index and hence results from the field of approximate aggregation can be easily translated to the field of property testing for the property 'belongs to the class of consistent aggregation mechanism'. Special interest should be in results that restrict the aggregation mechanisms to systematic aggregation mechanisms (For instance Blum, Luby, and Rubinfeld's result ([3],[2]) can be seen as a result for approximate aggregation using systematic aggregation mechanisms for the xor agenda.).

The first work studying approximate aggregation was done for the preference agenda over three candidate by Kalai[12] (although without stating the general framework of approximate aggregation). In this paper he proved the following bound for approximate aggregation mechanisms.

Theorem ([12])**.** *There exists an absolute constant K such that the following holds: For any $\epsilon > 0$ and any aggregation mechanism F for the preference agenda over 3 candidates that satisfies: F is balanced*[18]*, F is independent, and $IC(F) < K\epsilon$, there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F,G) < \epsilon$.*

This theorem was extended by Mossel[17] for preference agendas over any number of candidates and non-balanced aggregation mechanisms but with worse dependence of $IC(F)$ in ϵ (instead of linear as above).

Our main theorem gives bounds for every premise-conclusion agenda in which every conclusion is a function of at most two of the premises.

Theorem 4.1 (Main theorem)**.**
For any $\epsilon > 0$ and $n \geqslant 1$, there exists $\delta = poly\left(\frac{1}{n}, \epsilon\right)$, such that for every premise-conclusion agenda in which each issue is a premise, a conclusion of one premise, or a conclusion of two premises, if F is an aggregation mechanism for $\mathbb{X}$ over n voters satisfying independence and $IC(F) < \delta$, then there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F,G) < \epsilon$.
Moreover, one may take $\delta = Cn^{-2}\epsilon^5$ for some absolute constant C.

5 Proof Sketch

We prove this theorem by proving it explicitly for three specific agendas: the id agenda $\langle A, A\rangle$, the xor agenda $\langle A, B, A \oplus B\rangle$, and the conjunction agenda $\langle A, B, A \wedge B\rangle$. Since every boolean function on two bits can be reduced to one of the cases $f(x, y) = x$, $f(x, y) = y$, $f(x, y) = x \wedge y$, and $f(x, y) = x \oplus y$ by negating the inputs and output (which is renaming of opinions in our framework) we get theorem 4.1 using induction on the number of conclusions.

Below we sketch the proof idea for the xor agenda and conjunction agenda. The proofs of the more technical lemmas can be found in the full version.

[18]For every pair of candidates, a and b, it holds that the probability that F ranks a above b is exactly 1/2.

5.1 Proof for the xor agenda

For the agenda $\langle A, B, A \oplus B\rangle$ we prove:

Theorem 5.1. *For any $\epsilon < \frac{1}{6}$ and any independent aggregation mechanism F: If $IC(F) \leqslant \epsilon$, then there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F,G) \leqslant 3\epsilon$.*

Proof sketch.
Technique[19]: The proof uses the Fourier representation of boolean functions. That means representing the functions as linear combinations of the linear boolean functions.

Given an independent aggregation mechanism $F = \langle f, g, h\rangle$ we analyze the expression $\mathbb{E}[f(x)g(y)h(xy)]$ when x and y are sampled uniformly and independently. On one hand we show that $\mathbb{E}[f(x)g(y)h(xy)] = 1 - 2IC(F)$. On the other hand we show that $\mathbb{E}[f(x)g(y)h(xy)] = \sum_{\chi \in \mathtt{Lin}} \widehat{f}(\chi)\widehat{g}(\chi)\widehat{h}(\chi)$ when $\left|\widehat{f}(\chi)\right|$ equals $1 - 2\min\left(d\left(f,\chi\right), d\left(f,-\chi\right)\right)$. Hence, when $IC(F)$ is small then this sum is close to one and hence there exists a linear function such that f,g, and h are close to it (up to negation). Noticing that for any linear function χ, $\langle \chi,\chi,\chi\rangle$ and the permutations of $\langle -\chi,-\chi,\chi\rangle$ are consistent independent aggregation mechanism for this agenda gives us the result.

5.2 Proof for the conjunction agenda

For the agenda $\langle A, B, A \wedge B\rangle$ we prove:

Theorem 5.2. *For any $\epsilon > 0$ and any independent aggregation mechanism F: If $IC(F) \leqslant \epsilon$, then there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F,G) < 5\sqrt[5]{n^2\epsilon}$.*

Proof sketch.
Technique: The main insight in the proof is that we can bound the product of the influence of a voter on f and the ignorability of the same voter for g (and vice versa) using the inconsistency index of F by $P_i(f) \cdot I_i(g) \leqslant 4IC(F)$.

Let $F = \langle f, g, h\rangle$ be an aggregation mechanism that satisfies $IC(F) \leqslant \epsilon$. In case that f (or g) is close enough to the constant zero function, F is close to the consistent aggregation mechanism $\langle 0, g, 0\rangle$.

Otherwise, we define for a given function $f : \{0,1\}^n \to \{0,1\}$ and a coalition J (the junta), the junta function $f^J : \{0,1\}^n \to \{0,1\}$. It is derived from f in the following way:

$$f^J(x) = \text{majority}\left\{f(y) \mid y_J = x_J\right\}.$$

I.e., for a given input, f^J reads only the votes of the junta members, iterates over all the possible votes for the members outside the junta, and returns the more frequent result (assuming uniform distribution over the votes of the voters outside J).
We define f^J and g^J with regard to the junta of all the voters with small ignorability for either f or g. We prove that f^J and g^J are close to f and g, respectively and that there exists an issue aggregation function $h^\star$ such that $\left\langle f^J, g^J, h^\star\right\rangle$ is a consistent aggregation mechanism that is close to F.

There is a known characterization of the consistent independent aggregation mechanism for the conjunction agenda. (This characterization is a direct corollary from a series of works in the more general framework of aggregation, E.g., [19], [5]. We include a proof of it in the full version)

[19] The proof is similar to the analysis of the BLR (Blum-Luby-Rubinfeld) linearity test done in [2].

Lemma 5.3.
Let $f, g, h : \{0,1\}^n \to \{0,1\}$ be three voting functions satisfying $IC(\langle f, g, h\rangle) = 0$. Then either $f = h \equiv 0$, or $g = h \equiv 0$, or $f = g = h \in$ `Olig`.

A corollary from this theorem and theorem 5.2 is a characterization of the approximate aggregation mechanisms for this agenda. Actually ,in the proof of theorem 5.2 we get a tighter characterization that distinguishes between the two cases of consistent independent aggregation mechanism.

6 General Definition of Approximate Aggregation

In this paper we defined approximate aggregation by leaving the independence constraint intact and relaxing the consistency constraint. In this section we show that under a more general definition of approximate aggregation that relaxes both constraints we get similar results for any agenda and hence we do not lose much by restricting ourselves to the narrower definition.

Let $\mathbb{X}$ be an agenda and let F be an aggregation mechanism for that agenda. We define the **dependency index** as a measure for 'not satisfying independence'.

Definition 6.1 (dependency index)**.**
For an agenda $\mathbb{X}$ and an aggregation mechanism F for that agenda, the dependency index $DI^{\mathbb{X}}(F)$ is defined by

$$DI^{\mathbb{X}}(F) = \max_{j=1,\ldots,m} DI^{j,\mathbb{X}}(F) \quad \text{when} \quad DI^{j,\mathbb{X}}(F) = \mathop{\mathbb{E}}_{X \in \mathbb{X}^n}\left[\Pr_{Y \in \mathbb{X}^n}\left[F(X) \neq F(Y) | X^j = Y^j\right]\right]$$

That is, $DI^{j,\mathbb{X}}(F)$ is the probability that the following test for dependence of aggregating issue j on other issues fails (returns `False`):

Choose a profile X uniformly at random.
Choose a profile Y that agrees with X on issue j uniformly at random.
Return whether $F(X) \neq F(Y)$

We are interested in theorems of the form (for a given agenda $\mathbb{X}$):

Theorem. *For any $\epsilon > 0$ and $n \geqslant 1$, there exist $\delta_{IC}, \delta_{DI} > 0$*[20]*, such that if F is an aggregation mechanism for $\mathbb{X}$ over n voters satisfying $IC(F) \leqslant \delta_{IC}$ and $DI(F) \leqslant \delta_{DI}$, then there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F, G) < \epsilon$.*

It is easy to see that theorems of this form are generalizations of theorems of the form we proved in this paper and one can easily derive approximate aggregation results for independent aggregation mechanisms ($DI(F) = 0$) from theorems of the above general form.

It turns out that one can derive theorems the other way too using the following proposition.[21]

Proposition 6.1. *Let G be an aggregation mechanism for an agenda over m issues that satisfies $DI(G) \leqslant \delta_{DI}$. Then there exists an independent aggregation mechanism F that satisfies $d(F, G) \leqslant 2m\delta_{DI}$*

Given a result in the following format (which is the format we proved for in this paper):

Let $\delta : [0,1] \to [0,1]$ be a function s.t. for any $\epsilon > 0$: If F is an aggregation mechanism satisfying independence and $IC(F) \leqslant \delta(\epsilon)$, then there exists an aggregation mechanism G that satisfies consistency and independence such that $d(F, G) < \epsilon$.

[20] We would like δ_{IC}, δ_{DI} to not be too small. For instance we would like them to be *poly* $\left(\frac{1}{n}, \epsilon\right)$.
[21] Due to space limitations we omit the proof. It can be found in the full version of this paper.

We will define $\delta_{IC} = \frac{1}{2}\delta\left(\frac{\epsilon}{2}\right)$, $\delta_{DI} = \frac{1}{4m}\min\left(\delta\left(\frac{\epsilon}{2}\right), \epsilon\right)$. Now, let G be an aggregation mechanism that satisfies $IC(G) \leqslant \delta_{IC}$ and $DI(G) \leqslant \delta_{DI}$. Then based on proposition 6.1 there is an independent aggregation mechanism F such that $d(F, G) \leqslant 2m\delta_{DI}$. It is easy to see that $IC(F) \leqslant IC(G) + d(F, G)$ and for any aggregation mechanism H, $d(G, H) \leqslant d(F, H) + d(F, G)$ and hence there exists an aggregation mechanism H that satisfies consistency and independence such that $d(G, H) < \epsilon$.[22]

Notice that the dependency of δ_{IC} and δ_{DI} in ϵ and n (for instance, being polynomial in these parameters) is 'inherited' from the dependency of δ in ϵ and n. Therefore, such result will be similar in quality to the result for approximate aggregation mechanism that satisfies independence and we do not lose much by restricting ourselves to studying approximate aggregation by mechanisms that satisfying independence when analyzing a given agenda.

7 Summary and Future Work

In this paper we defined the issue of approximate aggregation which is a generalization of the study of aggregation mechanisms that satisfy consistency and independence. We defined measures for the relaxation of the consistency constraint (inconsistency index IC) and for the relaxation of the independence constraint (dependency index DI) .

We proved that relaxing these constraints does not extend the set of satisfying aggregation mechanisms in a non-trivial way for any premise-conclusion agenda in which every conclusion can be stated as a function of at most two of the premises. Particulary we calculated the dependency between the extension of this class (ϵ) and the inconsistency index ($\delta(\epsilon)$) (although maybe not strictly) for any premise-conclusion agenda of three issues. The relation we proved includes dependency on the number of voters (n). In both the works that preceded us for preference agendas (Kalai[12] and Mossel[17]) the relation did not include such a dependency. An interesting question is whether such a dependency is inherent for premise-conclusion agendas or whether it is possible to prove a relation that does not depend on n.

A major assumption in this paper is the uniform distribution over the inputs which is equivalent to assuming i.i.d uniform distribution over the premises. We think that our results can be extended for other distributions (still assuming voters' opinions are distributed i.i.d) over the space over premises' opinions which seem more realistic.

Immediate extensions for this work can be to extend our result to more complex premise-conclusion agendas and generalize our results for three issues premise-conclusion agenda and Kalai and Mossel's works for the preference agenda to get a unified bound for any three issues agenda.

A major open question is whether one can find an agenda for which relaxing the constraints of independence and consistency extends the class of satisfying aggregation mechanisms in a non-trivial way.

References

[1] K. J. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58:328, 1950.

[2] M. Bellare, D. Coppersmith, et al. Linearity testing in characteristic two. In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, p. 432. IEEE Computer Society, Washington, DC, USA, 1995. ISBN 0-8186-7183-1.

[22] For specific agendas, one might be able to strengthen some of these inequalities using the structure of $\mathbb{X}$ to get a stronger bounds on δ_{IC} and δ_{DI}.

[3] M. Blum, M. Luby, et al. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549 – 595, 1993. ISSN 0022-0000.

[4] F. Dietrich and C. List. Arrows theorem in judgment aggregation. *Social Choice and Welfare*, 29(1):19–33, 2007.

[5] E. Dokow and R. Holzman. Aggregation of binary evaluations for truth-functional agendas. *Social Choice and Welfare*, 32(2):221–241, February 2009.

[6] —. Aggregation of binary evaluations. *Journal of Economic Theory*, 145(2):495 – 511, 2010. ISSN 0022-0531.

[7] —. Aggregation of non-binary evaluations. *Advances in Applied Mathematics*, In Press, Corrected Proof:–, 2010. ISSN 0196-8858.

[8] E. Fischer. The art of uninformed decisions. *Bulletin of the EATCS*, 75:97, 2001.

[9] P. Fishburn and A. Rubinstein. Aggregation of equivalence relations. *Journal of Classification*, 3(1):61–65, March 1986.

[10] E. Friedgut, G. Kalai, et al. Elections can be manipulated often. In *FOCS '08: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pp. 243–249. IEEE Computer Society, Washington, DC, USA, 2008. ISBN 978-0-7695-3436-7.

[11] N. K. Ioannis Caragiannis, Christos Kaklamanis and A. D. Procaccia. Socially desirable approximations for dodgson's voting rule. In *Proc. 11th ACM Conference on Electronic Commerce*. 2010. ISBN 0-7695-2468-0.

[12] G. Kalai. A fourier-theoretic perspective on the condorcet paradox and arrow's theorem. *Adv. Appl. Math.*, 29(3):412–426, 2002. ISSN 0196-8858.

[13] L. A. Kornhauser and L. G. Sager. Unpacking the court. *The Yale Law Journal*, 96(1):82–117, 1986. ISSN 00440094.

[14] C. List and C. Puppe. Judgement aggregation: A survey. In P. P. P. Anand and C. Puppe, eds., *The Handbook of Rational and Social Choice*. Oxford University Press, USA, 2009.

[15] Marie Jean Antoine Nicolas Caritat de Condorcet. *Essai sur l'application de l'analyse la probabilit des dcisions rendues la pluralit des voix*. 1785.

[16] A. D. Miller. Group identification. *Games and Economic Behavior*, 63(1):188 – 202, 2008. ISSN 0899-8256.

[17] E. Mossel. A quantitative arrow theorem, 2010.

[18] K. Nehring. Arrows theorem as a corollary. *Economics Letters*, 80(3):379–382, 2003.

[19] K. Nehring and C. Puppe. Consistent judgement aggregation: The truth-functional case. *Social Choice and Welfare*, 31(1):41–57, 2008.

[20] A. Rubinstein and P. C. Fishburn. Algebraic aggregation theory. *Journal of Economic Theory*, 38(1):63–77, February 1986.

[21] A. Rubinstein and A. Kasher. On the question "Who is a J?": A social choice approach. Princeton Economic Theory Papers 00s5, Economics Department, Princeton University, Sep. 1998.

Ilan Nehama
Center for the Study of Rationality
& The Selim and Rachel Benin School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel
Email: `ilan.nehama@mail.huji.ac.il`

Complexity of Winner Determination and Strategic Manipulation in Judgment Aggregation

Ulle Endriss, Umberto Grandi, and Daniele Porello

Abstract

Judgment aggregation is an area of social choice theory that analyses procedures for aggregating the judgments of a group of agents regarding a set of interdependent propositions (modelled as formulas in propositional logic). The judgment aggregation framework gives rise to a number of algorithmic problems, including (1) computing a collective judgment from a profile of individual judgments (the winner determination problem), and (2) deciding whether a given agent can influence the outcome of a judgment aggregation procedure in her favour by reporting insincere judgments (the manipulation problem). We study the computational complexity of both these problems for two concrete judgment aggregation procedures that are complete and consistent and that have been argued to be useful in practice: the premise-based procedure and (a new variant of) distance-based merging. Our results suggest that manipulating these procedures is significantly harder than solving the corresponding winner determination problem.

1 Introduction

Judgment aggregation (JA) is an area of social choice theory that analyses procedures for aggregating the judgments of a group of agents regarding a set of interdependent propositions (List and Puppe, 2009). In JA, we are given a set of propositional formulas (the agenda) and ask several agents to report which of these formulas they judge to be true. How should we aggregate this information into a collective judgment? And under what circumstances will the collective judgment be consistent? To date, most technical contributions to the JA literature have been of an axiomatic flavour, establishing characterisations and impossibility theorems (e.g. List and Pettit, 2002; Dietrich, 2006). In recent work, we have begun to investigate the computational properties of the JA framework (Endriss et al., 2010). Here, we want to extend the scope of this work and suggest a framework for analysing the computational complexity of two algorithmic problems associated with concrete JA procedures: the *winner determination problem*, i.e., the problem of computing the collective judgment from a profile of individual judgments, and the *manipulation problem*.

In the context of voting, a player is said to be able to *manipulate* a voting rule when there exists a situation in which voting in a manner that does not truthfully reflect her preferences will result in an outcome that she prefers to the outcome that would be realised if she were to vote truthfully (Gaertner, 2006). What would constitute an appropriate definition of manipulation in the context of JA? This is not immediately clear, because in JA there is no notion of preference. Here, we follow Dietrich and List (2007) and assume that a player's individual judgment set is also her most preferred outcome and amongst any two outcomes she will prefer the one that is "closer" to that most preferred outcome. We will measure "closeness" using the Hamming distance. So, we will call an aggregation procedure F *manipulable* if it permits a situation where an agent can change the outcome to a judgment set that is closer to her true judgment set by reporting untruthfully. A procedure that cannot be manipulated is called *strategy-proof*.

Dietrich and List (2007) show that F is strategy-proof if and only if it is *independent* and *monotonic*. Thus, for a meaningful study of the computational complexity of strategic manipulation, we have to restrict attention to rules that are not both independent and monotonic.[1] Furthermore, for this initial study of the subject, we choose to focus on rules that produce *consistent* and *complete* judgment sets. Specifically, we analyse two rules: the *premise-based procedure* (Kornhauser and Sager, 1993; Dietrich and Mongin, 2010) and (a new variant of) *distance-based merging* (Pigozzi, 2006). For both procedures, we compare the complexity of manipulation with the complexity of winner determination.

For the premise-based procedure, we show that manipulating the procedure is NP-hard, while winner determination is possible in polynomial time. Thus, misuse of the procedure is significantly harder than using it in the intended manner (under the common assumption that P $\neq$ NP). For distance-based merging, we show that (the decision problem corresponding to) winner determination is in NP and we conjecture that manipulation is Σ_2^p-complete (which would place the latter problem at the second level of the polynomial hierarchy). That is, under the common assumption that the polynomial hierarchy does not collapse, this would, again, make manipulation considerably harder than winner determination.

The remainder of this paper is organised as follows.[2] In Section 2 we recall the framework of JA and define the winner determination and manipulation problems. The premise-based procedure is analysed in Section 3 and distance-based merging in Section 4. We conclude with a brief discussion of related work in Section 5.

2 Judgment Aggregation

In this section we recall the basic formal framework of JA familiar from the literature (List and Pettit, 2002; Dietrich, 2006; List and Puppe, 2009) and introduce a particular notion of strategic manipulation originally proposed by Dietrich and List (2007). To make the problem amenable to a complexity-theoretic investigation, we then formulate manipulation as a decision problem, and we do the same for the winner determination problem.

2.1 The Basic Framework

We now define the basic framework for JA.[3] Let PS be a set of propositional variables, and $\mathcal{L}_{PS}$ the set of propositional formulas built from PS (using the usual connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, and the constants $\top$ and $\bot$). If α is a propositional formula, define $\sim\alpha$, the complement of α, as $\neg\alpha$ if α is not negated, and as β if $\alpha = \neg\beta$. An **agenda** is a finite nonempty set $\Phi \subseteq \mathcal{L}_{PS}$ not containing any doubly-negated formulas that is closed under complementation (i.e., if $\alpha \in \Phi$ then $\sim\alpha \in \Phi$). Denote with Φ^+ the set of positive formulas in Φ. A **judgment set** J on an agenda Φ is a subset of the agenda $J \subseteq \Phi$. Define $J(\varphi) = 1$ if $\varphi \in J$, and $J(\varphi) = 0$ if $\varphi \notin J$. We call a judgment set J **complete** if $\alpha \in J$ or $\sim\alpha \in J$ for all $\alpha \in \Phi$; **complement-free** if for no $\alpha \in \Phi$ both α and $\sim\alpha$ are in J; and **consistent** if there exists an assignment that makes all formulas in J true. Denote with $\mathcal{J}(\Phi)$ the set of all complete consistent subsets of Φ. Given a set $N = \{1, \ldots, n\}$ of $n \geqslant 3$ **agents**, denote with $\mathbf{J} = (J_1, \ldots, J_n)$ a **profile** of judgment sets, one for each agent.

Definition 1 (Aggregation procedure)**.** *A (resolute) aggregation procedure for an agenda Φ and a set of n individuals is a function $F : \mathcal{J}(\Phi)^n \rightarrow 2^{\Phi}$.*

[1] Independent and monotonic aggregation procedures are not very attractive: they are either dictatorial or risk producing inconsistent outcomes unless the agenda is structurally very simple (List and Puppe, 2009).

[2] We shall assume familiarity with the basics of complexity theory up to the notion of NP-completeness (see e.g. Papadimitriou, 1994). We also make reference to two complexity classes at the second level of the polynomial hierarchy: Σ_2^p, the class of problems for which a certificate can be verified in polynomial time by a machine equipped with an NP oracle, and Π_2^p, the class of problems that are complements of those in Σ_2^p.

[3] Following our earlier work (Endriss et al., 2010), to allow for a precise analysis of the computational aspects of JA, we make slight changes to the standard framework (see e.g. List and Puppe, 2009): e.g., we allow for tautologies in the agenda and we make a clear distinction between purely "syntactic" and "logical" criteria (complement-freeness *vs.* consistency). We also permit irresolute JA procedures.

That is, F maps each profile of individual judgment sets to a collective judgment set. (In Section 4 we will also introduce an **irresolute** procedure that returns a *set of* collective judgment sets.) An aggregation procedure F, defined on an agenda Φ, is said to be complete (complement-free, consistent) if $F(\mathbf{J})$ is complete (complement-free, consistent) for every $\mathbf{J} \in \mathcal{J}(\Phi)$. Here, we are only interested in procedures that are complete and consistent (and thus also complement-free). As discussed at length in the literature, these are not easy criteria to satisfy. The **majority rule**, for instance, which accepts a formula if and only if a majority of agents do, fails to satisfy consistency (Kornhauser and Sager, 1993).

Axioms provide a normative framework in which to state what the desirable (or essential) properties of aggregation procedures are. Important axioms include **anonymity**, stating that the procedure should treat all agents the same; **neutrality**, requiring symmetry with respect to propositions; **independence**, postulating that collective acceptance of φ should only depend on individual acceptance patterns of φ; and **monotonicity**, specifying that additional support for a collectively accepted formula φ should never cause φ to get rejected.[4] While all of these axioms are intuitively appealing, several *impossibility theorems*, establishing inconsistencies between certain combinations of axioms with other desiderata, have been proved in the literature. The original impossibility theorem of List and Pettit (2002), for instance, shows that there can be no consistent and complete aggregation procedure satisfying anonymity, neutrality, and independence.

2.2 Strategic Manipulation

We now define the notion of strategic manipulation for JA sketched in the introduction. Our definition is an instance of a more general definition proposed by Dietrich and List (2007), which is based on the idea that we can induce a preference relation over judgment sets by assuming that an agent's true judgment set J is her most preferred outcome, and between any two outcomes the one that is "closer" to J is preferred. One of the most appealing choices for such a notion of "closeness" is the Hamming distance.

Definition 2 (Hamming distance). *Given an agenda Φ, let $J, J' \in 2^{\Phi}$ be two complete and complement-free judgment sets for Φ. The Hamming distance $H(J, J')$ between J and J' is the number of positive formulas on which they differ:*

$$H(J, J') \quad = \quad \sum_{\varphi \in \Phi^+} |J(\varphi) - J'(\varphi)|$$

That is, $H(J, J')$ is an integer between 0 (complete agreement) and $\frac{|\Phi|}{2}$ (complete disagreement). For example, if the agenda is $\Phi = \{p, \neg p, q, \neg q, p \wedge q, \neg(p \wedge q)\}$, then the Hamming distance between $J = \{\neg p, q, \neg(p \wedge q)\}$ and $J' = \{p, \neg q, \neg(p \wedge q)\}$ is $H(J, J') = 2$. Intuitively, if J_i is the true judgment set of agent i, then i "prefers" J over J' if $H(J_i, J) < H(J_i, J')$.

Definition 3 (Manipulability). *Let Φ be an agenda, let $F : \mathcal{J}(\Phi)^n \to 2^{\Phi}$ be an aggregation procedure for that agenda, and let $\mathbf{J} = (J_1, \ldots, J_i, \ldots, J_n) \in \mathcal{J}(\Phi)^n$ be a profile. Then F is said to be manipulable at $\mathbf{J}$, if there exist an alternative judgment set $J_i' \in \mathcal{J}(\Phi)$ for some agent $i \in N$ such that $H(J_i, F(J_i', \mathbf{J}_{-i})) < H(J_i, F(\mathbf{J}))$.*

That is, by reporting J_i' rather than her truthful judgment set J_i, agent i can achieve the outcome $F(J_i', \mathbf{J}_{-i})$ and that outcome is closer (in terms of the Hamming distance) to her truthful (and most preferred) set J_i than the outcome $F(\mathbf{J})$ that would get realised if she were to truthfully report J_i. A procedure that is not manipulable at any profile is called *strategy-proof*. Dietrich and List (2007) have shown that a JA procedure is strategy-proof if and only if it is independent and monotonic. Thus, to study the complexity of manipulation, we have to restrict ourselves to procedures that are either not independent or not monotonic.

[4] See (List and Puppe, 2009) or (Endriss et al., 2010) for formal presentations of these axioms.

2.3 Strategic Manipulation as a Decision Problem

To study the complexity of strategic manipulation, we formulate manipulation as a decision problem. We propose the following simple definition, parametrised by the judgment aggregation procedure F under consideration.

Manipulable(F)
Instance: Agenda Φ, judgment set $J_i \in \mathcal{J}(\Phi)$, partial profile $\mathbf{J}_{-i} \in \mathcal{J}(\Phi)^{n-1}$.
Question: Is there a $J_i' \in \mathcal{J}(\Phi)$ s.t. $H(J_i, F(J_i', \mathbf{J}_{-i})) < H(J_i, F(J_i, \mathbf{J}_{-i}))$?

That is, agent i is the manipulator and her true judgment set is J_i. The other agents' judgments are given by $\mathbf{J}_{-i}$. If agent i does not manipulate, then the outcome will be $F(J_i, \mathbf{J}_{-i})$, and the Hamming distance of this outcome to her most preferred outcome (which is also J_i) is $H(J_i, F(J_i, \mathbf{J}_{-i}))$. The question we are asking is whether there exists another judgment set J_i' that agent i could report instead that would lead to an outcome $F(J_i', \mathbf{J}_{-i})$ that is closer to J_i in terms of the Hamming distance. That is, we are asking *whether* she can manipulate successfully, rather than *how*.

2.4 Winner Determination as a Decision Problem

Next, we also formulate winner determination as a decision problem:

WinDet(F)
Instance: Agenda Φ, profile $\mathbf{J} \in \mathcal{J}(\Phi)^n$, formula $\varphi \in \Phi$.
Question: Is φ an element of $F(\mathbf{J})$?

By solving WinDet once for each formula in the agenda, we can compute the collective judgment set from an input profile (and, *vice versa*, any algorithm for computing the collective judgment set can be used to solve WinDet).

3 Premise-based Judgment Aggregation

There are two basic (types of) JA procedures that (can be set up so as to) produce consistent outcomes that have been discussed in the JA literature from its very beginnings, namely the *premise-based* (or *issue-based*) and the *conclusion-based* (or *case-based*) procedure (Kornhauser and Sager, 1993; Dietrich and Mongin, 2010). The basic idea is to divide the agenda into premises and conclusions. In the premise-based procedure, we apply the majority rule to the premises and then infer which conclusions to accept given the collective judgments regarding the premises;[5] under the conclusion-based procedure we directly ask the agents for their judgments on the conclusions and leave the premises unspecified in the collective judgment set. That is, the conclusion-based procedure does not result in complete outcomes, which is why we shall not consider it any further here. The premise-based procedure, on the other hand, can be set up in a way that guarantees consistent and complete outcomes, which provides a usable procedure of some practical interest—despite its well-documented shortcomings (Kornhauser and Sager, 1993; Pigozzi, 2006).

In this section, we first formally introduce the precise variant of the premise-based procedure we shall analyse. We then study the complexity of the winner determination and manipulation problems for this procedure. For ease of exposition, throughout this section, we shall assume that the number of agents n is odd.

[5] This is what is commonly understood by "premise-based procedure". Dietrich and Mongin (2010), who call this rule *premise-based majority voting*, have also investigated a more general class of premise-based procedures in which the procedure used to decide upon the premises need not be the majority rule.

3.1 Definition of the Procedure

For many JA problems, it will be natural to divide the agenda into premises and conclusions.

Definition 4 (Premise-based procedure)**.** *Let $\Phi = \Phi_p \uplus \Phi_c$ be an agenda divided into a set of premises Φ_p and a set of conclusions Φ_c, each of which is closed under complementation. The premise-based procedure* PBP : $\mathcal{J}(\Phi)^n \to 2^\Phi$ *for Φ_p and Φ_c is the function mapping each profile $\mathbf{J} = (J_1, \ldots, J_n) \in \mathcal{J}(\Phi)^n$ to the following judgment set:*

$$\begin{aligned} \mathrm{PBP}(\mathbf{J}) \quad &= \quad \Delta \cup \{\varphi \in \Phi_c \mid \Delta \models \varphi\}, \\ & \quad\ \ \textit{where } \Delta = \{\varphi \in \Phi_p \mid \#\{i \mid \varphi \in J_i\} > \frac{n}{2}\} \end{aligned}$$

If we want to ensure that the PBP always returns judgment sets that are consistent and complete, then we have to impose certain restrictions:

- If we want to guarantee *consistency*, we have to impose restrictions on the premises. It is well-known that the majority rule is guaranteed to be consistent if and only if the agenda Φ satisfies the so-called *median property*, i.e., if every inconsistent subset of Φ has itself an inconsistent subset of size $\leqslant 2$ (Nehring and Puppe, 2007; List and Puppe, 2009). This result immediately transfers to the PBP: it is consistent if and only if the set of premises satisfies the median property.
- If we want to guarantee *completeness*, we have to impose restrictions on the conclusions: for any assignment of truth values to the premises, the truth value of each conclusion has to be fully determined.

Deciding whether a set of formulas satisfies the median property is known to be Π_2^p-hard (Endriss et al., 2010). That is, in its most general form, deciding whether the PBP can be applied correctly is a highly intractable problem (and, as we shall see, a problem that is most likely considerably harder than either using or manipulating the PBP). For a meaningful analysis, we therefore restrict attention to the following case. First, we assume that the agenda Φ is closed under propositional variables: $p \in \Phi$ for any propositional variable p occurring within any of the formulas in Φ. Second, we equate the set of premises with the set of literals. Clearly, the above-mentioned conditions for consistency and completeness are satisfied under these assumptions.

So, to summarise, the procedure we consider in this section is defined as follows: Under the assumption that the agenda is closed under propositional variables, the PBP accepts a literal ℓ if and only if more individual agents accept ℓ than do accept $\sim\ell$, and the PBP accepts a compound formula if and only if it is entailed by the accepted literals. For consistent and complete profiles, and under the assumption that n is odd, this leads to a resolute JA procedure that is consistent and complete.

3.2 Winner Determination

Winner determination is a tractable problem for the premise-based procedure:

Proposition 1. WinDet(PBP) *is in P.*

Proof. Counting the number of agents accepting each of the premises and checking for each premise whether the positive or the negative instance has the majority is easy. This determines the collective judgment set as far as the premises are concerned. Deciding whether a given conclusion should be accepted by the collective now amounts to a model checking problem (is the conclusion φ true in the model induced by the accepted premises/literals?), which can also be done in polynomial time. □

3.3 Strategic Manipulation

Manipulating the premise-based procedure, on the other hand, is intractable:

Theorem 2. Manipulability(PBP) *is NP-complete.*

Proof. We first establish NP-membership. An untruthful judgment set J_i' yielding a preferred outcome can serve as a certificate. Checking the validity of such a certificate means checking that (a) J_i' is actually a complete and consistent judgment set and that (b) the outcome produced by J_i' is better than the outcome produced by the truthful set J_i. As for (a), checking completeness is easy. Consistency can also be decided in polynomial time: for every propositional variable p in the agenda, J_i' must include either p or $\neg p$; this admits only a single possible model; all that remains to be done is checking that all compound formulas in J_i' are satisfied by that model. As for (b), we need to compute the outcomes for J_i and J_i' (by Proposition 1, this is polynomial), compute their Hamming distances from J_i, and compare those two distances.

Next, we prove NP-hardness by reducing Sat to Manipulability(PBP). Suppose we are given a propositional formula φ and want to check whether it is satisfiable. We will build a judgment profile for three agents such that the third agent can manipulate the aggregation if and only if φ is satisfiable. Let $p_1, \ldots, p_m$ be the propositional variables occurring in φ, and let q_1, q_2 be two additional propositional variables. Define an agenda Φ that contains all atoms $p_1, \ldots, p_m, q_1, q_2$ and their negation, as well as $m + 2$ syntactic variants of the formula $q_1 \vee (\varphi \wedge q_2)$ and their negation. For instance, if $\psi := q_1 \vee (\varphi \wedge q_2)$, we might use the syntactic variants ψ, $\psi \wedge \top$, $\psi \wedge \top \wedge \top$, and so forth. The judgment profile $\mathbf{J}$ is defined by the following table (the rightmost column has a "weight" of $m + 2$):

	p_1	p_2	$\cdots$	p_m	q_1	q_2	$q_1 \vee (\varphi \wedge q_2)$
J_1	1	1	$\cdots$	1	0	0	?
J_2	0	0	$\cdots$	0	0	1	?
J_3	1	1	$\cdots$	1	1	0	1
$F(\mathbf{J})$	1	1	$\cdots$	1	0	0	0

The judgments of agents 1 and 2 regarding $q_1 \vee (\varphi \wedge q_2)$ are irrelevant for our argument, so they are indicated as "?" in the table (but note that they can be determined in polynomial time; in particular, $J_1(q_1 \vee (\varphi \wedge q_2)) = 0$ for any φ).

If agent 3 reports her judgment set truthfully (as shown in the table), then the Hamming distance between J_3 and the collective judgment set will be $1 + (m + 2) = m + 3$. Note that agent 3 is decisive about all propositional variables (i.e., premises) except q_1 (which will certainly get rejected). Now:

- If φ is satisfiable, then agent 3 can report judgments regarding $p_1, \ldots, p_m$ that correspond to a satisfying assignment for φ. If she furthermore accepts q_2, then all $m + 2$ copies of $q_1 \vee (\varphi \wedge q_2)$ will get accepted in the collective judgment set. Thus, the Hamming distance from J_3 to this new outcome will be at most $m + 2$, i.e., agent 3 will have manipulated successfully.

- If φ is not satisfiable, then there is no way to get any of the $m + 2$ copies of $q_1 \vee (\varphi \wedge q_2)$ accepted (and q_1 will get rejected in any case). Thus, agent 3 has no means of improving over the Hamming distance of $m + 3$ she can guarantee for herself by reporting truthfully.

Hence, φ is satisfiable if and only if agent 3 can manipulate successfully, and our reduction from Sat to Manipulability(PBP) is complete. □

Thus, manipulating the premise-based procedure is significantly harder than using it—at least in terms of worst-case complexity (and under the common assumption that P $\neq$ NP).

4 Distance-Based Judgment Aggregation

Pigozzi (2006) has shown that ideas from belief merging (Konieczny and Pino Pérez, 2002) can be imported into JA to yield practical aggregation procedures that are complete and consistent. Specifically, Pigozzi proposes a procedure that works roughly as follows: associate with each individual judgment set the model(s) satisfying that judgment set; merge the resulting set of models to obtain a new collection of models that minimise the sum of the (minimal) Hamming distances to the individual models; and return a collective judgment set corresponding to that collection of models. In this section, we introduce a new variant of this procedure and we study the computational complexity of its winner determination and manipulation problems.

4.1 A New Procedure: "Syntactic" Distance-based Merging

The merging procedure of Pigozzi (2006) has the drawback of being defined for a somewhat restricted class of profiles: the agenda is assumed to be closed under propositional variables and all compound formulas (the integrity constraints) are unanimously accepted (or rejected) by all agents. Most importantly, the syntactic information contained in the agenda is discarded by moving the aggregation from the level of formulas to the level of models. Our own proposal for distance-based merging in JA consists of a syntactic variant of this procedure, where we merge judgment sets rather than models corresponding to judgment sets. It is an *irresolute* procedure, returning a (nonempty) set of collective judgment sets.

Definition 5 (Distance-based procedure)**.** *Given an agenda Φ, the distance-based procedure* DBP *is the function mapping each profile $\mathbf{J} = (J_1, \ldots, J_n) \in \mathcal{J}(\Phi)^n$ to the following set of judgment sets:*

$$\mathrm{DBP}(\mathbf{J}) \quad = \quad \arg\min_{J \in \mathcal{J}(\Phi)} \sum_{i=1}^{n} H(J, J_i)$$

A collective judgment set under the DBP minimises the amount of disagreement with the individual judgment sets. Note that in cases where the majority rule leads to a consistent outcome, the outcome of the DBP coincides with that of the majority rule (making it a resolute procedure over these profiles). In all other profiles the consistent judgment sets that are the closest with respect to the Hamming distance are chosen as collective outcomes.

The DBP can be made *resolute* by introducing a tie-breaking rule (e.g., a lexicographic tie-breaking rule). Note that the DBP does not coincide with the procedure of Pigozzi (2006), even for agendas closed under propositional variables. The main reason is that the DBP is sensitive to logical correlations between formulas of the agenda: accepting an atom that is correlated with other formulas in the agenda "counts" more in our procedure than accepting an independent one. We find this an appealing property for a JA procedure, since it does not discard the syntactic information contained in the agenda. Also note that the DBP shares many features with the *Kemeny rule* for preference aggregation (Kemeny, 1959). We will elaborate more on this similarity in the proof of Lemma 4.

4.2 Winner Determination

Next, we want to analyse the complexity of the winner determination problem for the DBP. As the DBP is not resolute, we cannot work with the decision problem WinDet(DBP). The reason is that when there is more than one winning set, each query to WinDet (to settle the assignment for one formula at a time) may relate to a *different* winning set. We therefore formulate a new decision problem specifically for the DBP:

WinDet*(DBP)
Instance: Agenda Φ, profile $\mathbf{J} \in \mathcal{J}(\Phi)^n$, formula $\varphi \in \Phi$, $K \in \mathbb{N}$.
Question: Is there a $J^\star \in \mathcal{J}(\Phi)$ with $\varphi \in J^\star$ s.t. $\sum_{J \in \mathbf{J}} H(J^\star, J) \leqslant K$?

That is, we ask whether there is a J with Hamming distance at most K that accepts φ. To see that this is an appropriate formulation for a decision problem corresponding to the task of computing *some* winning set, note that we can compute a winner using a polynomial number of queries to WinDet*(DBP) as follows. We first use it to find the smallest K for which φ_1 can be accepted, as well as the smallest K for which $\sim\varphi_1$ can be accepted ($n \cdot m$ is an obvious upper bound for K, so this can be done with a polynomial number of queries). Then we accept either φ_1 or $\sim\varphi_1$, whichever did yield the smaller K (choose either one in case of a tie). Now leave K fixed for the rest of the process. Next, substitute φ_1 with the appropriate truth value throughout $\mathbf{J}$. Then check whether φ_2 can be accepted yielding distance K; if not, $\sim\varphi_2$ must be acceptable with distance K. Accept the appropriate formula and make the appropriate substitutions in $\mathbf{J}$; then continue with φ_3, and so forth.

Unsurprisingly, the DBP is much more complex a procedure than the PBP. Nevertheless, as we show next, the complexity of winner determination does at least not exceed NP.

Lemma 3. WinDet*(DBP) *is in NP.*

Proof. We will show that WinDet*(DBP) can be modelled as an integer program (without an objective function). This proves membership in NP (Papadimitriou, 1981).

Suppose we want to answer an instance of WinDet*(DBP). The number of subformulas of propositions occurring in the agenda Φ is linear in the *size* (not *cardinality*) of Φ. We introduce a binary decision variable for each of these subformulas: $x_i \in \{0,1\}$ for the ith subformula. We first write constraints that ensure that the chosen outcome will correspond to a consistent judgment set (i.e., that $J^\star \in \mathcal{J}(\Phi)$). Note that we can rewrite any formula in terms of negation, conjunction, and bi-implication without resulting in a superpolynomial (or even superlinear) increase in size.[6] So we only need to show how to encode the constraints for these connectives. The following table indicates how to write these constraints.

$\varphi_2 = \neg\varphi_1$	$x_2 = 1 - x_1$
$\varphi_3 = \varphi_1 \wedge \varphi_2$	$x_3 \leqslant x_1$ and $x_3 \leqslant x_2$ and $x_1 + x_2 \leqslant x_3 + 1$
$\varphi_3 = \varphi_1 \leftrightarrow \varphi_2$	$x_1 + x_2 \leqslant x_3 + 1$ and $x_1 + x_3 \leqslant x_2 + 1$ and $x_2 + x_3 \leqslant x_1 + 1$ and $1 \leqslant x_1 + x_2 + x_3$

Before we continue, consider the following way of rewriting the sum of distances featuring in the definition of WinDet*(DBP):

$$\begin{aligned}\sum_{J \in \mathbf{J}} H(J^\star, J) &= \sum_{i=1}^{n} \sum_{\varphi \in \Phi^+} |J^\star(\varphi) - J_i(\varphi)| \\ &= \frac{1}{2} \cdot \sum_{\varphi \in \Phi} \sum_{i=1}^{n} |J^\star(\varphi) - J_i(\varphi)| \\ &= \frac{1}{2} \cdot \sum_{\varphi \in \Phi} |n \cdot J^\star(\varphi) - \sum_{i=1}^{n} J_i(\varphi)|\end{aligned}$$

We will need to bound this sum from above. Now suppose that variables x_i with indices $i \in \{1, \ldots, m\}$ with $m = |\Phi|$ are those that correspond to the propositions that are elements of Φ. Let a_i be the number of individuals that accept the ith proposition in Φ (according

[6] For instance, any occurrence of $A \vee B$ can be rewritten as $\neg(\neg A \wedge \neg B)$. Note that rewriting a formula with nested bi-implications in terms of $\neg$ and $\wedge$ alone may result in an exponential blow-up.

to $\mathbf{J}$). To compute a winner under the DBP, we need to find a consistent judgement set $J^\star$ (characterised by variables $x_1, \ldots, x_m$) that minimises the sum $|n \cdot x_1 - a_1| + \cdots + |n \cdot x_m - a_m|$. We do this by introducing an additional set of integer variables $y_i > 0$ for $i = 1, \ldots, m$. We can ensure that $y_i = |n \cdot x_i - a_i|$ by adding the following constraints:

$$\begin{array}{ll} (\forall i \leqslant m) & n \cdot x_i - a_i \leqslant y_i \\ (\forall i \leqslant m) & a_i - n \cdot x_i \leqslant y_i \end{array}$$

Now the sum $\frac{1}{2} \cdot \sum_{i=1}^{m} y_i$ corresponds to the Hamming distance between the winning set and the profile. To ensure it does not exceed K, we can add the following constraint:

$$\frac{1}{2} \cdot \sum_{i=1}^{m} y_i \leqslant K$$

Finally, let $x_{i^\star}$ be the the variable corresponding to the formula $\varphi \in \Phi$ for which we want to answer WinDet*(DBP). We can force that φ gets accepted by adding one last constraint:

$$x_{i^\star} = 1$$

Now, by construction, the integer program we have presented is feasible if and only if the instance of WinDet*(DBP) we have started out with should be answered in the positive. □

Our proof also produces an algorithm for performing distance-based merging in practice. Observe that the following integer program (now *with* an objective function) can be used to find (some) winning judgment set under the DBP:

$$\min \sum_{i=1}^{m} y_i \quad \text{subject to all of the above constraints}$$

The solution can be read off from the values of the x_i. Note that the implementation details of the IP solver used will implicitly determine a tie-breaking rule. If required, other tie-breaking rules can be implemented explicitly.

Next, we show that the upper bound established by Lemma 3 is tight. Here, the similarity of the DBP to the Kemeny rule in preference aggregation allows us to build on a known NP-hardness result from the literature (Bartholdi et al., 1989b; Hemaspaandra et al., 2005).

Lemma 4. WinDet*(DBP) *is NP-hard.*

Proof sketch. We build a reduction from the problem Kemeny Score, as defined by Hemaspaandra et al. (2005). An instance of this problem consists of a set of candidates C, a profile of linear orders[7] $\mathbf{P} = (P_1, \ldots, P_n)$ over C, a designated candidate c, and a positive integer K. The *Kemeny score* of candidate c is given by the following expression:

$$\mathit{KemenyScore}(c, \mathbf{P}) = \min\{\sum_{i=1}^{n} d(P_i, Q) \mid \mathrm{top}(Q) = c\}$$

where $d(P_i, Q)$ is the Hamming distance between preference profiles and $\mathrm{top}(Q)$ is the most preferred candidate. The problem asks whether the Kemeny score of c is less than K.

We now build an instance of WinDet*(DBP) to decide this problem. Define an agenda Φ_C in the following way. First add propositional variables p_{ab} for all ordered pairs of candidates a, b in C; these variables can encode a linear order over C as a binary relation

[7]Although the Kemeny rule is defined for weak orders, the problem is known to remain NP-complete also in the case of linear orders (Bartholdi et al., 1989b, Lemma 3).

(where p_{ab} stands for $a > b$). Then add m^2 (where $m = |C|$) syntactic variants of the formula $p_{ab} \wedge p_{bc} \rightarrow p_{ac}$ for all suitable combinations of ordered pairs of candidates; these formulas encode the transitivity of the linear order encoded by the first set of variables. Finally, add an additional variable top_c. Given a preference profile $\mathbf{P}$ we can build a judgment profile $\mathbf{J_P}$ by encoding all strict orders P_i over C in a judgment set J_P over Φ_C. Due to space constraints we just show this procedure for a simple example with three candidates:

$$P = \{a > b > c\} \quad \Rightarrow \quad J_P = \{p_{ab}, p_{bc}, \neg p_{ca}, \neg top_c, \text{all transitivity constraints}\}$$

To conclude, it is sufficient to notice that $d(P,Q) = H(J_P, J_Q)$ in case P and Q share the same top candidate, otherwise the difference is 1. It is therefore sufficient to ask a query to WinDet*(DBP) using $\mathbf{J_P}$ as a profile, a suitable K' as a bound, and top_c as the fixed formula φ, to obtain an answer to the initial Kemeny Score instance with parameter K. The key step is to notice that judgment sets encoding intransitive preferences will not be considered in the minimisation process, since every disagreement on a transitivity formula will cause a much greater loss in the Hamming distance than what can be gained by modifying the variables encoding the candidate rankings. □

Putting Lemma 3 and 4 together yields a complete characterisation of the complexity of winner determination under distance-based merging:

Theorem 5. WinDet*(DBP) *is NP-complete.*

4.3 Strategic Manipulation

Next, we discuss the complexity of manipulating the DBP. Note that our definition of manipulation was tailored to resolute aggregation procedures, while the DBP (in its most general form) is irresolute and may return a set of winners. One interesting line of research to pursue in future work would be to define appropriate notions of manipulation and strategy-proofness for irresolute JA procedures. Here, instead, we shall assume that the DBP comes with a fixed tie-breaking rule (say, a lexicographic rule, or even the tie-breaking rule implicit in the IP formulation of the procedure given above, for a specific IP implementation). We do assume that this tie-breaking rule does not increase the complexity of winner determination beyond NP (this is the case for the two examples mentioned). Let Manipulability(DBP^t) be the manipulation problem for the DBP with such a fixed tie-breaking rule.

Establishing the precise complexity of manipulation for distance-based merging is currently an open problem. However, we are able to provide an upper bound:

Lemma 6. Manipulability(DBP^t) *is in* Σ_2^p.

Proof sketch. To show membership in Σ_2^p we need to show that it is possible to verify a certificate in polynomial time on a machine that has access to an NP oracle. Recall from the first part of the proof of Theorem 2 that an appropriate certificate is a judgment set J_i' for the manipulator that is complete and consistent and that produces an outcome that is closer to the manipulator's true judgment set J_i than the outcome produced if she reports J_i. This involves three non-trivial steps, all of which can be resolved by the NP oracle: deciding consistency of J_i' is in NP (this is just Sat), and computing the winners for J_i and J_i' is also in NP (by Lemma 3). Thus, the certificate can be verified using three calls to the oracle; the remainder of the computation is clearly polynomial. □

We conjecture that the above bound is tight, i.e., that Manipulability(DBP^t) is Σ_2^p-complete.[8] If this conjecture is correct, then manipulation is significantly harder than winner determination, also in the case of distance-based merging.

[8]To the best of our knowledge, there are currently no known results on the complexity of the (presumably) closely related problem of manipulating Kemeny elections.

5 Related Work

We conclude by briefly reviewing some related work regarding (1) alternative notions of manipulation in JA, (2) other complexity-theoretic questions in JA, (3) manipulation and strategy-proofness in belief merging, and (4) the complexity of manipulation in voting.

As mentioned earlier, our definition of strategic manipulation in JA is based on the work of Dietrich and List (2007). This definition crucially rests on the idea that we can induce a preference ordering over judgment sets from an agent's true judgment set an a metric for measuring "closeness". The Hamming distance is one such metric; Dietrich and List (2007) also discuss the concept of "closeness-respecting" preferences (and the corresponding notions of strategic manipulation) in more general terms. Other than that there has been preciously little work on manipulation in JA to date. One exception is the work of Pigozzi et al. (2009), who introduce a notion of *full manipulability*, which asks whether an agent can change the outcome to fully coincide with her own judgment set by means of an insincere judgment. But (as clearly recognised by the authors) the guarantee of the absence of full manipulation is probably a property that is simply too easy to satisfy to lead to interesting characterisations of JA procedures.

In previous work (Endriss et al., 2010), we have analysed the complexity of another aspect of the JA framework: for a given set of axioms characterising a class of aggregation procedures, how hard is it to check whether a given agenda is *safe* for all procedures belonging to that class, in the sense that no profile of complete and consistent individual judgment sets will ever result in a collective judgment set that is not consistent? (Our results suggest that deciding *safety of the agenda* is Π_2^p-complete for most natural combinations of the standard axioms.) To the best of our knowledge, this is the only other work on the computational complexity of JA to date.

The field of *belief merging* is closely related to judgment aggregation (Konieczny and Pino Pérez, 2002; Pigozzi, 2006). A definition of strategy-proofness for belief merging operators has been proposed by Everaere et al. (2007), and the same authors have discussed the problem of manipulation for a range of belief merging operators. While this work does include the study of the complexity of belief merging, the complexity of manipulation has, to the best of our knowledge, not yet been addressed in the belief merging literature.

Finally, there are of course close connections between our work and the line of work in computational social choice that has studied the complexity of both the winner determination and the manipulation problem for a range of voting rules in depth, starting with the seminal work of Bartholdi et al. (1989a,b). Some of this work has been reviewed by Chevaleyre et al. (2007), who give many references. Recent discussion in the literature on the complexity of manipulation of elections has centred on the question of whether worst-case results (such as NP-hardness results) are sufficient deterrents against manipulation in practice (see e.g. Procaccia and Rosenschein, 2007). They probably are not; what is really needed is a better understanding of the average-case complexity of manipulation. The very same questions will have to be asked for JA as well; our (worst-case intractability) result and conjecture are only the first step.

References

J. J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989a.

J. J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989b.

Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *Proc. 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007)*. Springer-Verlag, 2007.

F. Dietrich. Judgment aggregation: (im)possibility theorems. *Journal of Economic Theory*, 126(1):286–298, 2006.

F. Dietrich and C. List. Strategy-proof judgment aggregation. *Economics and Philosophy*, 23(3):269–300, 2007.

F. Dietrich and P. Mongin. The premiss-based approach to judgment aggregation. *Journal of Economic Theory*, 145:562–582, 2010.

U. Endriss, U. Grandi, and D. Porello. Complexity of judgment aggregation: Safety of the agenda. In *Proc. 9th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010)*, 2010.

P. Everaere, S. Konieczny, and P. Marquis. The strategy-proofness landscape of merging. *Journal of Artificial Intelligence Research*, 28(1):49–105, 2007.

W. Gaertner. *A Primer in Social Choice Theory.* LSE Perspectives in Economic Analysis. Oxford University Press, 2006.

E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

J. Kemeny. Mathematics without numbers. *Daedalus*, 88:577–591, 1959.

S. Konieczny and R. Pino Pérez. Merging information under constraints: A logical framework. *Journal of Logic and Computation*, 12(5):773–808, 2002.

L. A. Kornhauser and L. G. Sager. The one and the many: Adjudication in collegial courts. *California Law Review*, 81(1):1–59, 1993.

C. List and P. Pettit. Aggregating sets of judgments: An impossibility result. *Economics and Philosophy*, 18(1):89–110, 2002.

C. List and C. Puppe. Judgment aggregation: A survey. In *Handbook of Rational and Social Choice*. Oxford University Press, 2009.

K. Nehring and C. Puppe. The structure of strategy-proof social choice. Part I: General characterization and possibility results on median spaces. *Journal of Economic Theory*, 135(1):269–305, 2007.

C. H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.

C. H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28 (4):765–768, 1981.

G. Pigozzi. Belief merging and the discursive dilemma. *Synthese*, 152(2):285–298, 2006.

G. Pigozzi, M. Slavkovik, and L. van der Torre. A complete conclusion-based procedure for judgment aggregation. In *Proc. 1st International Conference on Algorithmic Decision Theory (ADT-2009)*. Springer-Verlag, 2009.

A. D. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, 2007.

Ulle Endriss, Umberto Grandi and Daniele Porello
Institute for Logic, Language and Computation
University of Amsterdam
Postbus 94242, 1090 GE Amsterdam, The Netherlands
Email: `{u.endriss,u.grandi,d.porello}@uva.nl`

Multivariate Complexity Analysis of Swap Bribery

Britta Dorn and Ildikó Schlotter[1]

Abstract

We consider the computational complexity of a problem modeling *bribery* in the context of voting systems. In the scenario of Swap Bribery, each voter assigns a certain price for swapping the positions of two consecutive candidates in his preference ranking. The question is whether it is possible, without exceeding a given budget, to bribe the voters in a way that the preferred candidate wins in the election. We initiate a parameterized and multivariate complexity analysis of Swap Bribery, focusing on the case of k-approval. We investigate how different cost functions affect the computational complexity of the problem. We identify a special case of k-approval for which the problem can be solved in polynomial time, whereas we prove NP-hardness for a slightly more general scenario. We obtain fixed-parameter tractability as well as W[1]-hardness results for certain natural parameters.

1 Introduction

In the context of voting systems, the question of how to manipulate the votes in some way in order to make a preferred candidate win the election is a very interesting question. One possibility is *bribery*, which can be described as spending money on changing the voters' preferences over the candidates in such a way that a preferred candidate wins, while respecting a given budget. There are various situations that fit into this scenario: The act of remunerating the voters in order to make them change their preferences, or paying money in order to get into the position of being able to change the submitted votes, but also the setting of systematically spending money in an election campaign in order to convince the voters to change their opinion on the ranking of candidates.

The study of bribery in the context of voting systems was initiated by Faliszewski, Hemaspaandra, and Hemaspaandra in 2006 [12]. Since then, various models have been analyzed. In the original version, each voter may have a different but fixed price which is independent of the changes made to the bribed vote. The scenario of nonuniform bribery introduced by Faliszewski [11] and the case of microbribery studied by Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe in [13] allow for prices that depend on the amount of change the voter is asked for by the briber.

In addition, the Swap Bribery problem as introduced by Elkind, Faliszewski, and Slinko [10] takes into consideration the ranking aspect of the votes: In this model, each voter may assign different prices for swapping two consecutive candidates in his preference ordering. This approach is natural, since it captures the notion of small changes and comprises the preferences of the voters. Elkind et al. [10] prove complexity results for this problem for several election systems such as Borda, Copeland, Maximin, and approval voting. In particular, they provide a detailed case study for k-approval. In this voting system, every voter can specify a group of k preferred candidates which are assigned one point each, whereas the remaining candidates obtain no points. The candidates which obtain the highest sum of points over all votes are the winners of the election. Two prominent special cases of k-approval are plurality, (where $k = 1$, i.e., every voter can vote for exactly one candidate) and veto (where $k = m - 1$ for m candidates, i.e., every voter assigns one point

[1]Supported by the Hungarian National Research Fund (OTKA 67651).

	Result	Reference
$k = 1$ (plurality)	P	[10]
$k = m - 1$ (veto)	P	[10]
$1 \leq k \leq m$, m or n constant	P	[10]
$1 \leq k \leq m$, all costs $= 1$	**P**	Thm. 1
$k = 2$	NP-complete	[2]
$3 \leq k \leq m - 2$, costs in $\{0, 1, 2\}$	NP-complete	[10]
$2 \leq k \leq m - 2$, costs in $\{0, 1\}$ and $\beta = 0$	**NP-complete**	[2], Prop. 2
$2 \leq k \leq m - 2$ is part of the input, costs in $\{0, 1\}$ and $\beta = 0$, n constant	**NP-complete**	[3], Prop. 2
$2 \leq k \leq m - 2$, costs in $\{\delta_1, \delta_2\}$, $\delta_2 \geq 2\delta_1 > 0$	**NP-complete**, **W[1]-hard** (β)	Thm. 3
$1 \leq k \leq m$	**FPT** (m)	Thm. 4
$1 \leq k \leq m$ is part of the input	**FPT** (β, n) **by kernelization**	Thm. 5
$1 \leq k \leq m$	**FPT** (β, n, k) **by kernelization**	Thm. 5

Table 1: Overview of known and new results for Swap Bribery for k-approval. The results obtained in this paper are printed in bold. Here, m and n denote the number of candidates and votes, respectively, and β is the budget. For the parameterized complexity results, the parameters are indicated in brackets. If not stated otherwise, the value of k is fixed.

to all but one disliked candidate). Table 1 shows a summary of research considering Swap Bribery for k-approval, including both previously known and newly achieved results.

This paper contributes to the further investigation of the case study of k-approval that was initiated in [10], this time from a parameterized point of view. The main goal of this approach is to find *fixed-parameter algorithms* confining the combinatorial explosion which is inherent in NP-hard problems to certain problem-specific parameters, or to prove that their existence is implausible. This line of research has been pioneered by Downey and Fellows [9], see also [15, 21] for two more recent monographs, and naturally expands into the field of multivariate algorithmics, where the influence of "combined" parameters is studied, see the recent survey by Niedermeier [22]. These approaches seem to be appealing in the context of voting systems, where NP-hardness is a desired property for various problems, like Manipulation, Lobbying, Control, or, as in our case, Swap Bribery. However, NP-hardness does not necessarily constitute a guarantee against such dishonest behavior. As Conitzer et al. [8] pointed out for the Manipulation problem, an NP-hardness result in these settings would lose relevance if an efficient fixed-parameter algorithm with respect to an appropriate parameter was found. Parameterized complexity can hence provide a more robust notion of hardness. The investigation of problems from voting theory under this aspect has started, see for example [1, 3, 4, 7, 20].

We show NP-hardness as well as fixed-parameter intractability of Swap Bribery for certain very restricted cases of k-approval if the parameter is the budget, whereas we identify a natural special case of the problem which can be solved in polynomial time. By contrast, we obtain fixed-parameter tractability with respect to the parameter 'number of candidates' for k-approval and a large class of other voting systems, and a polynomial kernel for k-approval if we consider certain combined parameters.

The paper is organized as follows. After introducing notation in Section 2, we investigate the complexity of Swap Bribery depending on the cost function in Section 3, where we show the connection to the Possible Winner problem, identify a polynomial-time solvable case of k-approval and a hardness result. In Section 4, we consider the parameter 'number of candidates' and obtain an FPT result for Swap Bribery for a large class of voting systems. We also consider the combination of parameters 'number of votes' and 'size of the budget'. We conclude with a discussion of open problems and further directions that might be interesting for future investigations.

2 Preliminaries

Elections. An *election* is a triple $E = (V, C, \mathcal{E})$, where $V = \{v_1, \ldots, v_n\}$ denotes the set of *votes* or *voters*, $C = \{c_1, \ldots, c_m\}$ is a set of *candidates*, and $\mathcal{E}$ is the *election system* which is a function mapping (V, C) to a set $W \subseteq C$ called the *winners* of the election. We will express our results for the *winner case* where several winners are possible, but our results can be adapted to the *unique winner case* where W consists of a single candidate only.

In our context, each vote is a strict linear order over the set C, and we denote by $\mathrm{rank}(c, v)$ the position of candidate $c \in C$ in a vote $v \in V$.

For an overview of different election systems, we refer to [6]. We will mainly focus on election systems that are characterized by a given *scoring rule*, expressed as a vector $(s_1, s_2, \ldots, s_m)$ where $m = |C|$. Given such a scoring rule, the *score* of a candidate c in a vote v, denoted by $\mathrm{score}(c, v)$, is $s_{\mathrm{rank}(c,v)}$. The score of a candidate c in a set of votes V is $\mathrm{score}(c, V) = \sum_{v \in V} \mathrm{score}(c, v)$, and the winners of the election are the candidates that receive the highest score in the given votes.

The election system we are particularly interested in is k-approval, which is defined by the scoring vector $(1, \ldots, 1, 0, \ldots, 0)$, starting with k ones. In the case of $k = 1$, this is the *plurality* rule, whereas $(m-1)$-approval is also known as *veto*. Given a vote v, we will say that a candidate c with $1 \leq \mathrm{rank}(c, v) \leq k$ takes a *one-position* in v, whereas a candidate c' with $k + 1 \leq \mathrm{rank}(c', v) \leq m$ takes a *zero-position* in v.

Swap Bribery, Possible Winner, Manipulation. Given V and C, a *swap* in some vote $v \in V$ is a triple (v, c_1, c_2) where $\{c_1, c_2\} \subseteq C, c_1 \neq c_2$. Given a vote v, we say that a swap $\gamma = (v, c_1, c_2)$ is *admissible in* v, if $\mathrm{rank}(c_1, v) = \mathrm{rank}(c_2, v) - 1$. Applying this swap means exchanging the positions of c_1 and c_2 in the vote v, we denote by v^γ the vote obtained this way. Given a vote v, a set Γ of swaps is *admissible in* v, if the swaps in Γ can be applied in v in a sequential manner, one after the other, in some order. Note that the obtained vote, denoted by v^Γ, is independent from the order in which the swaps of Γ are applied. We also extend this notation for applying swaps in several votes, in the straightforward way.

In a Swap Bribery instance, we are given V, C, and $\mathcal{E}$ forming an election, a preferred candidate $p \in C$, a cost function c mapping each possible swap to a non-negative integer, and a budget $\beta \in \mathbb{N}$. The task is to determine a set of admissible swaps Γ whose total cost is at most β, such that p is a winner in the election $(V^\Gamma, C, \mathcal{E})$. Such a set of swaps is called a *solution* of the Swap Bribery instance. The underlying decision problem is the following.

> Swap Bribery
> **Given:** An election $E = (V, C, \mathcal{E})$, a preferred candidate $p \in C$, a cost function c mapping each possible swap to a non-negative integer, and a budget $\beta \in \mathbb{N}$.
> **Question:** Is there a set of swaps Γ whose total cost is at most β such that p is a winner in the election $(V^\Gamma, C, \mathcal{E})$?

We will also show the connection between Swap Bribery and the Possible Winner problem. In this setting, we have an election where some of the votes may be *partial* orders over C instead of complete linear ones. The question is whether it is possible to extend the partial votes to complete linear orders in such a way that a preferred candidate wins the election. For a more formal definition, we refer to the article by Konczak and Lang [18] who introduced this problem. The corresponding decision problem is defined as follows.

> Possible Winner
> **Given:** A set of candidates C, a set of partial votes $V' = (v'_1, \ldots, v'_n)$ over C, an election system $\mathcal{E}$, and a preferred candidate $p \in C$.
> **Question:** Is there an extension $V = (v_1, \ldots, v_n)$ of V' such that each v_i extends v'_i, and p is a winner in the election $(V, C, \mathcal{E})$?

A special case of Possible Winner is Manipulation (see e.g. [8, 17]). Here, the given set of partial orders consists of two subsets; one subset contains linearly ordered votes and the other one completely unordered votes.

Parameterized complexity, Multivariate complexity. Parameterized complexity is a two-dimensional framework for studying the computational complexity of problems [9, 15, 21]. One dimension is the size of the input I (as in classical complexity theory) and the other dimension is the parameter k (usually a positive integer). A problem is called *fixed-parameter tractable* (FPT) with respect to a parameter k if it can be solved in $f(k) \cdot |I|^{O(1)}$ time, where f is an arbitrary computable function [9, 15, 21]. Multivariate complexity is the natural sequel of the parameterized approach when expanding to multidimensional parameter spaces, see [22]. For example, if we regard two parameters, say k_1 and k_2, then the desired FPT algorithm should run in time $f(k_1, k_2) \cdot |I|^{O(1)}$ for some f.

The first level of (presumable) parameterized intractability is captured by the complexity class W[1]. A *parameterized reduction* reduces a problem instance (I, k) in $f(k) \cdot |I|^{O(1)}$ time to an instance (I', k') such that (I, k) is a yes-instance if and only if (I', k') is a yes-instance, and k' only depends on k but not on $|I|$.

We will use the following W[1]-hard problem [14] for the hardness reduction in this work:

> Multicolored Clique
> **Given:** An undirected graph $G = (V_1 \cup V_2 \cup \cdots \cup V_k, E)$ with $V_i \cap V_j = \emptyset$ for $1 \leq i < j \leq k$ where the vertices of V_i induce an independent set for $1 \leq i \leq k$.
> **Question:** Is there a complete subgraph (clique) of G of size k?

We will also make use of a *kernelization* algorithm in this work, which is a standard technique for obtaining fixed-parameter results, see [5, 16, 21]. The idea is to transform the input instance (I, k) in a polynomial time preprocessing step via *data reduction rules* into a "reduced" instance (I', k') such that two conditions hold: First, (I, k) is a yes-instance if and only if (I', k') is a yes-instance, and second, the size of the reduced instance depends on the parameter only, i.e., $|I'| + |k'| \leq g(p)$ for some arbitrary computable function g. The reduced instance (I', k') is then referred to as the *problem kernel*. If in addition g is a polynomial function, we say that the problem admits a *polynomial kernel*. The existence of a problem kernel is equivalent to fixed-parameter tractability of the corresponding problem with respect to the particular parameter [21].

3 Complexity depending on the cost function

In this section, we focus our attention on Swap Bribery for k-approval. We start with the case where all costs are equal to 1, for which we obtain polynomial-time solvability.

Theorem 1. Swap Bribery *for k-approval is polynomial-time solvable, if all costs are* 1.

Proof. Let V be the set of votes and C be the set of candidates. The score of any candidate is an integer between 0 and $|V|$. Our algorithm finds out for each possible s^* with $1 \leq s^* \leq |V|$ whether there is a solution in which the preferred candidate p wins with score s^*.

Given a value s^*, we answer the above question by solving a corresponding minimum cost maximum flow problem. We will define a network $N = (G, s, t, g, w)$ on a directed graph $G = (D, E)$ with a source vertex s and a target vertex t, where g denotes the capacity function and w the cost function defined on E. First, we introduce the vertex sets $A = \{a_{v,c} \mid v \in V, c \in C, \mathrm{rank}(c, v) \leq k\}$, $A' = \{a'_{v,c} \mid v \in V, c \in C\}$ and $B = \{b_c \mid c \in C\}$, and we set $D = \{s, t, x\} \cup A \cup A' \cup B$. We define the arcs E as the union of the sets $E_S = \{sa \mid a \in A\}$, $E_A = \{a_{v,c}a'_{v,c} \mid \mathrm{rank}(c, v) \leq k\}$, $E_{A'} = \{a_{v,c}a'_{v,c'} \mid \mathrm{rank}(c, v) \leq k, \mathrm{rank}(c', v) > k\}$, $E_B = \{a'_{v,c}b_c \mid v \in V, c \in C\}$, $E_X = \{b_c x \mid c \in C, c \neq p\}$, plus the arcs $b_p t$ and xt. We set the cost function w to be 0 on each arc except for the arcs of $E_{A'}$, and we set $w(a_{v,c}a'_{v,c'}) = \mathrm{rank}(c', v) - \mathrm{rank}(c, v)$. We let the capacity g be 1 on the arcs of $E_S \cup E_A \cup E_{A'} \cup E_B$, we set it to be s^* on the arcs of $E_X \cup \{b_p t\}$, and we set $g(xt) = |V|k - s^*$.

The soundness of the algorithm and hence the theorem itself follows from the following observation (for a detailed proof, see the full version): there is a flow of value $|V|k$ on N having total cost at most β if and only if there exists a set Γ of swaps with total cost at most β such that $\mathrm{score}(p, V^\Gamma) = s^*$ and $\mathrm{score}(c, V^\Gamma) \leq s^*$ for any $c \in C, c \neq p$. □

Theorem 1 also implies a polynomial-time approximation algorithm for SWAP BRIBERY for k-approval with approximation ratio δ, if all costs are in $\{1, \delta\}$ for some $\delta \geq 1$.

Proposition 2 shows the connection between SWAP BRIBERY and POSSIBLE WINNER. This result is an easy consequence of a reduction given by Elkind et al. [10]. For the proof of the other direction, see again the full version.

Proposition 2. *The special case of* SWAP BRIBERY *where the costs are in* $\{0, \delta\}$ *for some* $\delta > 0$ *and the budget is zero is equivalent to the* POSSIBLE WINNER *problem.*

As a corollary, SWAP BRIBERY with costs in $\{0, \delta\}$, $\delta > 0$ and budget zero is NP-complete for almost all election systems based on scoring rules [2]. For many voting systems such as k-approval, Borda, and Bucklin, it is NP-complete even for a fixed number of votes [3].

We now turn to the case with two different positive costs, addressing 2-approval.

Theorem 3. *(1)* SWAP BRIBERY *for 2-approval, with costs in* $\{1, 2\}$*, is NP-complete.*
(2) SWAP BRIBERY *for 2-approval, with costs in* $\{1, 2\}$*, is W[1]-hard, if the parameter is the budget* β*, or equivalently, the maximum number of swaps allowed.*

Proof. We present a reduction from the MULTICOLORED CLIQUE problem. Let $F = (V, E)$ with the k-partition $V = V_1 \cup V_2 \cup \cdots \cup V_k$ be the given instance of MULTICOLORED CLIQUE. For each $1 \leq i < j \leq k$ we let $E^{i,j} = \{xy \mid x \in V_i, y \in V_j, xy \in E\}$. We construct an instance I_F of SWAP BRIBERY as follows.

The set C of candidates will be $C = \bigcup_{i \in [k]}(A_i \cup B_i \cup C_i) \cup D \cup G \cup \{p\}$ where $A_i = \{a_v^j \mid j \in [k], v \in V_i\}$, $B_i = \{b_v^j \mid j \in [k], v \in V_i\}$, and $C_i = \{c^{i,j} \mid j \in [k]\}$. (Here and later, we write $[k]$ for $\{1, 2, \ldots, k\}$.) Our preferred candidate is p. The sets $D = \{d_1, d_2, \ldots\}$ and $G = \{g_1, g_2, \ldots\}$ will contain *dummies* and *guards*, respectively. Our budget will be $\beta = 6k^2 - k$. Regarding the indices i and j, we will suppose $i, j \in [k]$ if not stated otherwise.

The set of votes will be $W = W_G \cup W_I \cup W_S \cup W_C$. Votes in W_G will define guards (explained later), votes in W_I will set the initial scores, votes in W_S will represent the selection of $\binom{k}{2}$ edges and k vertices, and finally, votes in W_C will be responsible for checking that the selected edges connect selected vertices. We construct W such that the following will hold for some fixed even integer K (determined later):

$\mathrm{score}(p, W) = K$.
$\mathrm{score}(c^{i,j}, W) = K + 1$ for each i and j,
$\mathrm{score}(q, W) = K$ for each $q \in \bigcup_{i \in [k]}(A_i \cup B_i) \cup G$, and
$\mathrm{score}(d, W) \leq 1$ for each $d \in D$.

We define the cost function c such that each swap has cost 1 or 2. We will define each cost to be 1 if not explicitly stated otherwise. Using that each cost is at least 1, we get that none of the candidates ranked after the position $\beta + 2$ in a vote v can receive non-zero score in v without violating the budget. Thus, we can represent votes by listing only their first $\beta + 2$ positions. We say that a candidate does not *appear* in some vote, if he is not contained in these positions.

Dummies, guards, and truncation. First, let us clarify the concept of dummy candidates: we will ensure that no dummy can receive more than one score in total, by letting each $d \in D$ appear in exactly one vote. This can be ensured easily by using at most $|W|(\beta + 2)$ dummies in total. We will use the sign $*$ to denote dummies in votes.

Now, we define $\beta + 2$ guards using the votes W_G. We let W_G contain votes of the form $w_G(h)$ for each $h \in [\beta + 2]$, each such vote having multiplicity $K/2$ in W_G. We let $w_G(h) = (g_h, g_{h+1}, g_{h+2}, \dots, g_{\beta+2}, g_1, g_2, \dots g_{h-1})$. Clearly, $\text{score}(g, W_G) = K$ for each $g \in G$, and the total score obtained by the guards in W_G cannot decrease. As we will make sure that our preferred candidate cannot receive more than K scores without exceeding the budget, this yields that in any possible solution, each guard must have score exactly K.

Using guards, we can *truncate* votes at any position $h > 2$ by putting arbitrarily chosen guards at the positions $h, h + 1, \dots, \beta + 2$. This way we ensure that only candidates on the first $h - 1$ positions can receive a score in this vote. We will denote truncation at position h by using a sign $\dagger$ at that position.

Setting initial scores. Using dummies and guards, we define W_I to adjust the initial scores of the relevant candidates as follows. We put the following votes into W_I:

$$\begin{gathered}
(p, *, \dagger) \text{ with multiplicity } K, \\
(c^{i,j}, *, \dagger) \text{ with multiplicity } K + 1 - |E^{i,j}| \text{ for each } i \neq j, \\
(c^{i,i}, *, \dagger) \text{ with multiplicity } K + 1 - |V_i| \text{ for each } i \in [k], \text{ and} \\
(q, *, \dagger) \text{ with multiplicity } K - 1 \text{ for each } q \in \textstyle\bigcup_{i \in [k]} (A_i \cup B_i).
\end{gathered}$$

The preferred candidate p will not appear in any other vote, implying $\text{score}(p, W) = K$.

Selecting edges and vertices. The set W_S consists of the following votes:

$$\begin{gathered}
w_S(i, x) = (*, c^{i,i}, a^i_x, \dagger) \text{ for each } i \in [k] \text{ and } x \in V_i, \text{ and} \\
w_S(i, j, x, y) = (c^{i,j}, c^{j,i}, a^j_x, a^i_y, \dagger) \text{ for each } i < j,\ x \in V_i,\ y \in V_j,\ xy \in E.
\end{gathered}$$

The cost of swapping $c^{i,j}$ with $c^{j,i}$ and the cost of swapping a^i_x with a^j_y in $w_S(i, j, x, y)$ is 2.

Checking incidency. The set W_C will contain the votes

$$w_C(i, x) = (a^i_x, b^{i-1}_x, b^i_x, *, \dagger) \text{ for each } i \in [k] \text{ and } x \in V_i.$$

Here $i - 1$ is taken modulo k. In $w_C(i, x)$ we let the cost of swapping a^i_x with b^{i-1}_x and also the cost of swapping b^i_x with the neighboring dummy be 2.

It remains to define K properly. To this end, we let K be the minimum even integer not smaller than the integers $|E^{i,j}|$ for every $1 \leq i < j \leq k$ and $|V_i|$ for each $i \in [k]$. This finishes the construction. Note that the initial scores of the candidates are as claimed above.

Construction time. Observe $|W_G| = (\beta + 2)K/2$, $|W_I| = O(Kk|V|)$, $|W_S| = |E| + |V|$, and $|W_C| = |V|$. Hence, the number of votes is polynomial in the size of the input graph F. This also implies that the number of candidates is polynomial as well, and the whole construction takes polynomial time. Note also that β is only a function of k, hence this yields an FPT reduction as well.

Our aim is to show the following: F has a k-clique if and only if the constructed instance is a yes-instance of SWAP BRIBERY. This will prove both (1) and (2).

Direction $\Longleftarrow$. Suppose that I_F is solvable, and there is a set Γ of swaps transforming W into W' with total cost at most β such that p wins in W' according to 2-approval. We also assume w.l.o.g. that Γ is a solution having minimum cost.

As argued above, $\text{score}(p, W') \leq K$ and $\text{score}(g, W') \geq K$ for each $g \in G$ follow directly from the construction. Thus, only $\text{score}(p, W') = \text{score}(g, W') = K$ for each $g \in G$ is possible. Thus, for any $i, j \in [k]$, by $\text{score}(c^{i,j}, W) = K + 1$ we get that $c^{i,j}$ must lose at least one score during the swaps. Considering $c^{i,i}$ (and the optimality of Γ), this means that each $c^{i,i}$ is swapped with a^i_x by Γ in $w_S(i, x)$ for some unique $x \in V_i$. We use the notation $\sigma(i)$ to denote this vertex x, i.e. we let $\sigma(i) = x$. We will show that the vertices $\sigma(1), \sigma(2), \dots, \sigma(k)$ form a k-clique in F.

Let us denote by Γ_{vs} the set of those swaps in Γ that swap $c^{i,i}$ with $a^i_{\sigma(i)}$ for some $i \in [k]$. Clearly, Γ_{vs} has total cost k.

Let us fix i and j now, assuming $i < j$. Since both $c^{i,j}$ and $c^{j,i}$ have the same score in W_I as in W_I^Γ, $c^{i,j}$ must lose a score due to swaps in $w_S(i, j, x_1, y_1)$ for some x_1 and y_1, and similarly, $c^{j,i}$ must lose a score due to swaps in $w_S(i, j, x_2, y_2)$ for some x_2 and y_2. Let $\Gamma_{\text{es}}(i, j)$ be the swaps applied in these two votes. There are three possibilities for $\Gamma_{\text{es}}(i, j)$:

(a) $w_S(i, j, x_1, y_1) = w_S(i, j, x_2, y_2)$, and the swaps in $\Gamma_{\text{es}}(i, j)$ transform the vote $(c^{i,j}, c^{j,i}, a^j_{x_1}, a^i_{y_1}, \dagger)$ into $(a^j_{x_1}, a^i_{y_1}, c^{i,j}, c^{j,i}, \dagger)$ through 4 swaps having total cost 4.

(b) $w_S(i, j, x_1, y_1) \neq w_S(i, j, x_2, y_2)$ and as a result of the swaps in $\Gamma_{\text{es}}(i, j)$, $c^{i,j}$ gets to the third position of $w_S(i, j, x_1, y_1)$, and $c^{j,i}$ gets to the third position of $w_S(i, j, x_2, y_2)$. In this case, $|\Gamma_{\text{es}}(i, j)| \geq 3$ and $c(\Gamma_{\text{es}}(i, j)) \geq 4$.

(c) $w_S(i, j, x_1, y_1) \neq w_S(i, j, x_2, y_2)$ and after the swaps in $\Gamma_{\text{es}}(i, j)$, at least one of $c^{i,j}$ and $c^{j,i}$ is placed on the fourth position in one of the votes $w_S(i, j, x_1, y_1)$ or $w_S(i, j, x_2, y_2)$. This means $|\Gamma_{\text{es}}(i, j)| \geq 4$ and $c(\Gamma_{\text{es}}(i, j)) \geq 5$.

From the above discussion, the cost of the swaps in $\Gamma_{\text{es}}(i, j)$ is at least 4. Moreover, as a result of the swaps in $\Gamma_{\text{es}}(i, j)$, the candidates in $a^j_{x_1}, a^i_{y_1}, a^j_{x_2}, a^i_{y_2}$ receive a total of 2 additional scores with respect to their initial score in W.

Let A^* denote those candidates in $\bigcup_{i \in [k]} A_i$ which receive an additional score as a result of the swaps in Γ_{vs} or in $\Gamma_{\text{es}}(i, j)$ for some $i < j$. The total score gained by the candidates in A^* during these swaps is exactly k^2. Since the initial score of each candidate in A^* is K, we know that the remaining swaps of Γ must force these candidates to lose a total of k^2 scores. Observe that this can only happen through swaps applied in W_C, and moreover, each candidate can lose at most one score with such swaps. This implies $|A^*| = k^2$.

Let Γ_c be the set of swaps in Γ applied in W_C, transforming W_C into a set of votes W'_C. The above discussion yields that $\text{score}(a, W_C) > \text{score}(a, W'_C)$ holds for each $a \in A^*$. Since Γ is a solution, we also obtain that $\text{score}(q, W_C) \leq \text{score}(q, W'_C)$ must hold for each $q \in \bigcup_{i \in [k]} B_i \cup G$. We will prove the following claim below.

Claim. $c(\Gamma_c) \geq 4k^2$, and equality can only be reached if

$$\{a^j_x \mid j \in [k]\} \cap A^* = \emptyset \text{ or } \{a^j_x \mid j \in [k]\} \subseteq A^* \text{ holds for each } x \in V. \tag{1}$$

Using this claim, $c(\Gamma) = c(\Gamma_{\text{vs}}) + \sum_{i<j} c(\Gamma_{\text{vs}}(i, j)) + c(\Gamma_c) \geq k + 4\binom{k}{2} + 4k^2 = 6k^2 - k = \beta$ follows. Thus, equalities must hold everywhere, resulting in the following consequences.

First, (1) implies that A^* is the union of sets of the form $\{a^1_x, a^2_x, \dots a^k_x\}$ for exactly k vertices x. By $a^i_{\sigma(i)} \in A^*$, this yields $A^* = \bigcup_{i,j \in [k]} \{a^j_{\sigma(i)}\}$. Recall that by our construction of the votes $w_S(i, x)$, we know $\sigma(i) \in V_i$ for each i.

Second, note that $c(\Gamma_{\text{es}}(i, j)) = 4$ shows that case (c) cannot happen for the swaps $\Gamma_{\text{es}}(i, j)$. Moreover, from (1) we have $|A^* \cap A_i| = k$ for each $i \in [k]$, which implies that case (b) can neither happen. Thus, the only possibility is case (a), meaning that the swaps of $\Gamma_{\mathrm{es}}(i,j)$ transform the vote $(c^{i,j}, c^{j,i}, a^j_x, a^i_y, \dagger)$ for some x and y into a vote $(a^j_x, a^i_y, c^{i,j}, c^{j,i}, \dagger)$. However, by the definition of $w_S(i,j,x,y)$ we know $x \in V_i$, $y \in V_j$, and $xy \in E$. But from $A^* = \bigcup_{i,j\in[k]}\{a^j_{\sigma(i)}\}$, we get that only $x = \sigma(i)$ and $y = \sigma(j)$ is possible. Hence, $\sigma(i)$ and $\sigma(j)$ are neighboring for each $i < j$, proving the first direction.

Before proving the other direction, it remains to show our claim. Let us fix some $x \in V$, and let us suppose $\{a^j_x \mid j \in [k]\} \cap A^* \neq \emptyset$. Let $|A^* \cap \{a^j_x \mid j \in [k]\}| = a^*_x$, and let $c(i)$ be the total cost of the swaps in Γ_c applied to $w_C(i,x)$. We are trying to show that $\sum_{i\in[k]} c(i) \geq 4a^*_x$ and equality implies $a^*_x = k$.

Recall that a^i_x appears only in the vote $w_C(i,x) = (a^i_x, b^{i-1}_x, b^i_x, *, \dagger)$ in W_C. We will use 0-1 variables α_i and β_i to denote whether the score of a^i_x and b^i_x, respectively, are changed in $w_C(i,x)$ as a result of the swaps in Γ_c. The following are elementary observations (sometimes we also use that Γ_c is of minimum cost, and we take $i-1$ modulo k):

1. If $\alpha_i = 1$ and $\beta_i = 0$ then $c(i) = 5$. (In this case, $\beta_{i-1} = 0$ must hold.)
2. If $\alpha_i = 0$ and $\beta_i = 1$ then $c(i) = 1$. (In this case, $\beta_{i-1} = 1$ must hold.)
3. If $\alpha_i = 0$, $\beta_i = 0$, and $\beta_{i-1} = 0$ then $c(i) = 0$.
4. If $\alpha_i = 0$, $\beta_i = 0$, and $\beta_{i-1} = 1$ then $c(i) = 3$.
5. If $\alpha_i = 1$, $\beta_i = 1$, and $\beta_{i-1} = 0$ then $c(i) = 3$.
6. If $\alpha_i = 1$, $\beta_i = 1$, and $\beta_{i-1} = 1$ then $c(i) = 4$.
7. If $\beta_i = 0$ and $\beta_{i-1} = 1$, then $\alpha_i = 1$ is not possible.

First, note that if $\beta_i = 1$ for every $i \in [k]$, then $\sum_{i\in[k]} c(i) = 4a^*_x + (k - a^*_x)$ follows directly by 2 and 6 above. Thus, $\sum_{i\in[k]} c(i) \geq 4a^*_x$ holds, and equality indeed implies $a^*_x = k$.

Otherwise, let us call a maximal series of indices $i, i+1, \ldots, j$ in $[k]$ a *segment*, if $\beta_i = \beta_{i+1} = \cdots = \beta_{j-1} = 1$ but $\beta_j = 0$. We think of such series in a cyclic manner, so $i > j$ is possible. First, observe that the cycle $1, 2, \ldots, k$ can be decomposed into a certain number of segments and a remaining set H of indices h for which $\beta_h = \beta_{h-1} = 0$. Let us write $I^* = \{i \mid a^i_x \in A^*\}$ for the set of indices associated with A^*. From claims 1 and 3, we know $\sum_{h\in H} c(h) = 5|I^* \cap H|$.

Now, consider a segment $i, i+1, \ldots, j$, and let S denote the set of its elements. By claims 7 and 4 we get $\alpha_j = 0$ and $c(j) = 3$. Since case 5 above can only apply for i, by an easy calculation we obtain $\sum_{h\in S} c(h) \geq c(j) + \sum_{h\in S\cap I^*} c(h) > 4|S \cap I^*|$. Taking into account all segments together with the set H, we get $\sum_{i\in[k]} c(i) > 4a^*_x$. From this, the claim follows.

Direction $\Longrightarrow$. Let $\sigma(1), \sigma(2), \ldots, \sigma(k)$ form a k-clique in F where $\sigma(i) \in V_i$ for each i. It is straightforward to check that the following swaps of total cost β yield a solution for I_F:
1. For each $i \in [k]$, swap $c^{i,i}$ with $a^i_{\sigma(i)}$ in $w_S(i, \sigma(i))$.
2. For each $i < j$, swap both $c^{i,j}$ and $c^{j,i}$ with both $a^j_{\sigma(i)}$ and $a^i_{\sigma(j)}$ in $w_S(i,j,\sigma(i),\sigma(j))$.
3. For each $i, j \in [k]$, swap both $a^j_{\sigma(i)}$ and $b^{j-1}_{\sigma(i)}$ with $b^j_{\sigma(i)}$ and the dummy in $w_C(i, \sigma(i))$. □

Looking into the proof of Theorem 3, we can see that the results hold even if the costs are uniform in the sense that swapping two given candidates has the same price in any vote, and the maximum number of swaps allowed in a vote is at most 4. By applying minor modifications to the given reduction, Theorem 3 can be generalized to hold for the following modified versions as well.

- If all costs are in $\{\delta_1, \delta_2\}$ such that $\delta_2 \geq 2\delta_1 > 0$: we only have to replace costs 1 and 2 with new costs δ_1 and δ_2, respectively.
- If we want p to be the unique winner: we only have to set $\mathrm{score}(p, W) = K + 1$.
- If we use k-approval for some $3 \leq k \leq |C| - 2$ instead of 2-approval: it suffices to insert $k-2$ dummies into the first $k-2$ positions of each vote.

Hence, Theorem 3 shows that Swap Bribery remains hard even if we consider such natural parameters as the maximum number of swaps allowed in a vote, the maximum number of different possible costs, or the maximum ratio of two different costs to have a fixed value.

4 Other parameterizations

In this section, we will consider different kinds of parameterizations. First, we will look at the parameter 'number of candidates'. For this case, the following observation is helpful.

Let $S_m = \{\pi_1, \pi_2, \ldots, \pi_{m!}\}$ be the set of permutations of size m. We say that an election system is *described by linear inequalities*, if for a given set $C = \{c_1, c_2, \ldots, c_m\}$ of candidates it can be characterized by $f(m)$ sets $A_1, A_2, \ldots A_{f(m)}$ (for some computable function f) of linear inequalities over $m!$ variables $x_1, x_2, \ldots, x_{m!}$ in the following sense: if n_i denotes the number of those votes in a given election E that order C according to π_i, then the first candidate c_1 is a winner of the election if and only if for at least one index i, the setting $x_j = n_j$ for each j satisfies all inequalities in A_i.

It is easy to see that many election systems can be described by linear inequalities: any system based on scoring rules, Copeland$^\alpha$ ($0 \leq \alpha \leq 1$), Maximin, Bucklin, Ranked pairs.

Theorem 4. Swap Bribery *is FPT if the parameter is the number of candidates, for any election system described by linear inequalities.*

Proof. Let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of candidates given, and let $A_1, A_2, \ldots A_{f(m)}$ be the sets of linear inequalities over variables $x_1, \ldots, x_{m!}$ that describe the given election system $\mathcal{E}$. For some $\pi_i \in S_m$, let v_i denote the vote that ranks C according to π_i. We describe the given set V of votes by writing n_i for the multiplicity of the vote v_i in V.

Our algorithm solves $f(m)$ integer linear programs with variables $T = \{t_{i,j} \mid i \neq j, 1 \leq i, j \leq m!\}$. We will use $t_{i,j}$ to denote the number of votes v_i that we transform into votes v_j; we will require $t_{i,j} \geq 0$ for each $i \neq j$. Let V^T denote the set of votes obtained by transforming the votes in V according to the variables $t_{i,j}$ for each $i \neq j$. Such a transformation from V is feasible if $\sum_{j \neq i} t_{i,j} \leq n_i$ holds for each $i \in [m!]$ (inequality $\mathcal{A}$).

By an observation in [10], we can compute the price $c_{i,j}$ of transforming the vote v_i into v_j in $O(m^3)$ time. Transforming V into V^T can be done with total cost at most β, if $\sum_{i,j \in [m!]} t_{i,j} c_{i,j} \leq \beta$ (inequality $\mathcal{B}$).

We can express the multiplicity x'_i of the vote v_i in V^T as $x'_i = n_i + \sum_{j \neq i} t_{j,i} - \sum_{i \neq j} t_{i,j}$. For some $i \in [f(m)]$, let A'_i denote the set of linear inequalities over the variables in T that are obtained from the linear inequalities in A_i by substituting x_i with the above given expression for x'_i. Using the description of $\mathcal{E}$ with the given linear inequalities, we know that the preferred candidate c_1 wins in $(V^T, C, \mathcal{E})$ for some values of the variables $t_{i,j}$ if and only if these values satisfy the inequalities of A'_i for at least one $i \in [f(m)]$. Thus, our algorithm solves Swap Bribery by finding a non-negative assignment for the variables in T that satisfies both the inequalities $\mathcal{A}$, $\mathcal{B}$, and all inequalities in A'_i for some i.

Solving such a system of linear inequalities can be done in linear FPT time, if the parameter is the number of variables [19]. By $|T| = (m! - 1)m!$ the theorem follows. □

Similarly, we can also show fixed-parameter tractability for other problems if the parameter is the number of candidates, for example for Possible Winner (this result was already obtained for several election systems by Betzler et al., [3]), Manipulation (both for weighted and unweighted voters), several variations of Control (this result was already obtained for Llull and Copeland voting by Faliszewski et al., [13]), or Lobbying [7] (here, the parameter would be the number of issues in the election). Since our topic is Swap Bribery, we will not go into detail here.

Finally, we consider a combined parameter and obtain fixed-parameter tractability.

Theorem 5. *If the minimum cost is* 1*, then* Swap Bribery *for k-approval (where k is part of the input) with combined parameter $(|V|, \beta)$ admits a kernel with $O(|V|^2\beta)$ votes and $O(|V|^2\beta^2)$ candidates. Here, V is the set of votes and β is the budget.*

Proof. Let V, C, $p \in C$, and β denote the set of votes, the set of candidates, the preferred candidate, and the budget given, respectively. The idea of the kernelization algorithm is that not all candidates are interesting for the problem: only candidates that can be moved within the budget β from a zero-position to a one-position or vice versa are relevant.

Let Γ be a set of swaps with total cost at most β. Clearly, as the minimum possible cost of a swap is 1, we know that there are only 2β candidates c in a vote $v \in V$ for which $\text{score}(c, v) \neq \text{score}(c, v^\Gamma)$ is possible, namely, such a c has to fulfill $k - \beta + 1 \leq \text{rank}(c, v) \leq k + \beta$. Thus, there are at most $2\beta|V|$ candidates for which $\text{score}(c, V) \neq \text{score}(c, V^\Gamma)$ is possible; let us denote the set of these candidates by $\widetilde{C}$. Let c^* be a candidate in $C \setminus \widetilde{C}$ whose score is the maximum among the candidates in $C \setminus \widetilde{C}$.

Note that a candidate $c \in C \setminus (\widetilde{C} \cup \{c^*, p\})$ has no effect on the answer to the problem instance. Indeed, if $\text{score}(p, V^\Gamma) \geq \text{score}(c^*, V^\Gamma)$, then the score of c is not relevant, and conversely, if $\text{score}(p, V^\Gamma) < \text{score}(c^*, V^\Gamma)$ then p loses anyway. Therefore, we can disregard each candidate in $C \setminus \widetilde{C}$ except for c^* and p.

The kernelization algorithm constructs an equivalent instance K as follows. In K, nor the budget, nor the preferred candidate will be changed. However, we will change the value of k to be $\beta + 1$, so the kernel instance K will contain a $(\beta + 1)$-approval election[2]. We define the set V_K of votes and the set C_K of candidates in K as follows.

First, the algorithm "truncates" each vote v, by deleting all its positions (together with the candidates in these positions) except for the 2β positions between $k - \beta + 1$ and $k + \beta$. Then again, we shall make use of dummy candidates (see the proof of Theorem 3); we will ensure $\text{score}(d, V^\Gamma) \leq 1$ for each such dummy d. Swapping a dummy with any other candidate will have cost 1 in K. Now, for each obtained truncated vote, the algorithm inserts a dummy candidate in the first position, so that the obtained votes have length $2\beta + 1$. In this step, the algorithm also determines the set $\widetilde{C}$ and the candidate c^*. This can be done in linear time. We denote the votes[3] obtained in this step by V_r. We do not change the costs of swapping candidates of $\widetilde{C} \cup \{c^*, p\}$ in some vote $v \in V_r$.

Next, to ensure that K is equivalent to the original instance, the algorithm constructs a set V_d of votes such that $\text{score}(c, V_r \cup V_d) = \text{score}(c, V)$ holds for each candidate c in $\widetilde{C} \cup \{p, c^*\}$. This can be done by constructing $\text{score}(c, V) - \text{score}(c, V_r)$ newly added votes where c is on the first position, and all the next 2β positions are taken by dummies. This way we ensure $\text{score}(c, V_d) = \text{score}(c, V_d^\Gamma)$ for any set Γ of swaps with total cost at most β.

If D is the set of dummy candidates created so far, then let $C_K = \widetilde{C} \cup \{p, c^*\} \cup D$. To finish the construction of the votes, it suffices to add for each vote $v \in V_r \cup V_d$ the candidates not yet contained in v, by appending them at the end (starting from the $(2\beta+1)$-th position) in an arbitrary order. The obtained votes will be the votes V_K of the kernel.

The presented construction needs polynomial time. Using the above mentioned arguments, it is straightforward to verify that the constructed kernel instance is indeed equivalent to the original one. Thus, it remains to bound the size of K.

Clearly, $|\widetilde{C} \cup \{p, c^*\}| \leq 2|V|\beta + 2$. The number of dummies introduced in the first phase is exactly $|V_r| = |V|$. As the score of any candidate in V is at most $|V|$, the number of votes created in the second phase is at most $(2|V|\beta + 2)|V|$, which implies that the number of dummies created in this phase is at most $(2|V|\beta + 2)|V| \cdot 2\beta$. Therefore, we obtain $|C_K| \leq |V| + (2|V|\beta + 2)(2|V|\beta + 1) = O(|V|^2\beta^2)$, and also $|V_K| \leq (2|V|\beta + 3)|V| = O(|V|^2\beta)$. ☐

[2]We use $\beta + 1$ instead of β to avoid complications with the case $\beta = 0$.

[3]Actually, these vectors are not real votes in the sense that they do not contain each candidate, but at the moment we do not care about this.

Applying similar ideas, a kernel with $(|V|+k)\beta$ candidates is easy to obtain, which might be favorable to the above result in cases where k is small.

5 Conclusion

We have taken the first step towards parameterized and multivariate investigations of Swap Bribery under certain voting systems. We obtained W[1]-hardness for k-approval if the parameter is the budget β, while Swap Bribery could be shown to be in FPT for a very large class of voting systems if the parameter is the number of candidates. This revaluates previous NP-hardness results: Swap Bribery could be computed efficiently if the number of candidates is small, which is a common setting, e.g. in presidential elections.

However, we have shown this via an integer linear program formulation, using a result by Lenstra, which does not provide running times that are suitable in practice. Here, it would be interesting to give combinatorial algorithms that compute an optimal swap bribery. This might be particularly relevant for a scenario described by Elkind et al. [10], where bribery is not necessarily considered as an undesirable thing, like in the case of campaigning.

As Elkind et al. [10] pointed out, it would be nice to characterize further natural polynomial-time solvable cases of Swap Bribery. We provided one such example with Theorem 1 for k-approval in the case where costs are equal to 1. By contrast, already the case of two different costs δ_1, δ_2 with $\delta_2 \geq 2\delta_1 > 0$ becomes NP-complete for k-approval ($2 \leq k \leq m-2$) and W[1]-hard if the parameter is the budget β. We believe that this can be generalized to the case of two different (arbitrary) positive costs.

There are plenty of possibilities to carry on our initiations. First, there are more parameterizations to be looked at, and in particular the study of combined parameters in the spirit of Niedermeier [22], see e.g. [1], is an interesting approach.

Also, we have focused our attention to k-approval, but the same questions could be studied for other voting systems, or for the special case of Shift Bribery which was shown to be NP-complete for several voting systems [10], or other variants of the bribery problem as mentioned in the introduction. For instance, we have only looked at *constructive* swap bribery, but the case of *destructive* swap bribery (when our aim is to achieve that a disliked candidate does *not* win) is worth further investigation as well.

Acknowledgments. We thank Rolf Niedermeier for an inspiring initial discussion on this topic.

References

[1] N. Betzler. On problem kernels for possible winner determination under the k-approval protocol. In *Proc. of 35th MFCS*, 2010.

[2] N. Betzler and B. Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *J. Comput. Syst. Sci.*, In Press, 2010.

[3] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. of 21st IJCAI*, pages 53–58, 2009.

[4] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009.

[5] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *IWPEC*, pages 17–37, 2009.

[6] S. J. Brams and P. C. Fishburn. Voting procedures. In *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. Elsevier, 2002.

[7] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, November 2007.

[8] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *J. ACM*, 54(3):1–33, 2007.

[9] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[10] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proc. of 2nd SAGT*, volume 5814 of *LNCS*, pages 299–310. Springer, 2009.

[11] P. Faliszewski. Nonuniform bribery. In *Proc. 7th AAMAS*, pages 1569–1572, 2008.

[12] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of bribery in elections. In *Proc. of 21st AAAI*, pages 641–646, 2006.

[13] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res. (JAIR)*, 35:275–341, 2009.

[14] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.

[15] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.

[16] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[17] E. Hemaspaandra and L. A. Hemaspaandra. Dichotomy for voting systems. *J. Comput. Syst. Sci.*, 73(1):73–83, 2007.

[18] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. of IJCAI-2005 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[19] H. Lenstra. Integer programming with a fixed number of variables. *Math. of OR*, 8:538–548, 1983.

[20] H. Liu, H. Feng, D. Zhu, and J. Luan. Parameterized computational complexity of control problems in voting systems. *Theor. Comput. Sci.*, 410(27-29):2746–2753, 2009.

[21] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.

[22] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proc. of 27th STACS*, pages 17–32, 2010.

Britta Dorn
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen
Sand 13, 72076 Tübingen, Germany
Email: `bdorn@informatik.uni-tuebingen.de`

Ildikó Schlotter
Budapest University of Technology and Economics
H-1521 Budapest, Hungary
Email: `ildi@cs.bme.hu`

Parameterized Control Complexity in Bucklin Voting and in Fallback Voting[1]

Gábor Erdélyi and Michael R. Fellows

Abstract

We study the parameterized control complexity of Bucklin voting and of fallback voting, a voting system that combines Bucklin voting with approval voting. Electoral control is one of many different ways for an external agent to tamper with the outcome of an election. We show that even though the representation of the votes and the winner determination is different, the parameterized complexity of some standard control attacks is the same. In particular, we show that adding and deleting candidates in both voting systems are W[2]-hard for both the constructive and destructive case, parameterized by the amount of action taken by the external agent. Furthermore, we show that adding and deleting voters in both Bucklin voting and fallback voting are W[2]-hard for the constructive case, parameterized again by the amount of action taken by the external agent, and are in FPT for the destructive case.

1 Introduction

The study of algorithmic issues related to voting systems has become an important topic in contemporary computer science, due to the many applications of deciding between alternatives, or ranking information, in a wide variety of contexts.

Rich questions inevitably arise about the tractability of the election processes, and their susceptibility to manipulation. This paper is about this context of research.

We study the complexity of manipulation of elections based on Bucklin voting, and of *fallback voting*, a voting system that combines Bucklin voting with approval voting.

2 Preliminaries

Many different ways of changing the outcome of an election have been studied with respect to the computational complexity of the strategy, such as *manipulation* [BTT89, BO91, CSL07, HH07, FHHR09b], where a group of voters casts their votes strategically, *bribery* [FHH09, FHHR09a], where an external agent bribes a group of voters in order to change their votes, and *control* [BTT92, HHR07, FHHR09a, HHR09, ENR09, FHHR09b, EPR10], where an external agent—which is referred to as "The Chair"—changes the structure of the election (for example, by adding/deleting/partitioning either candidates or voters).

In this paper, we are concerned with *control issues* for the relatively recently introduced system of *fallback voting* (FV, for short) [BS09] and *Bucklin voting* (BV, for short). A voting system is said to be *immune* against a certain type of control if it is impossible to affect the outcome of the election via that type of control. If a voting system is not immune to a type of control, then it is said to be *susceptible*. When control is possible, the task of exerting control may still be NP-hard. In this case the voting system is said to be *resistant* against that type of control. If the chair's task can be solved in polynomial-time for a type of control then the voting system is said to be *vulnerable* to that type of control.

[1]This work was supported in part by the DFG under grants RO 1202/12-1 (within the European Science Foundation's EUROCORES program LogICCC: "Computational Foundations of Social Choice") and RO 1202/11-1. Work done in part while the first author was visiting the University of Newcastle.

We investigate the issues in the framework parameterized complexity. Many voting systems present NP-hard algorithmic challenges. Parameterized complexity is a particularly appropriate framework in many contexts of voting systems because it is concerned with exact results that exploit the structure of input distributions. It is not appropriate in political contexts, for example, to algorithmically determine a winner "approximately". The computational complexity of control problems under the parameterized complexity framework has been studied before. Betzler and Uhlmann [BU08] proved that constructive control by deleting candidates in plurality voting is W[2]-hard with respect to the number of deleted candidates, and destructive control by deleting candidates in plurality voting is W[1]-hard with respect to the number of deleted candidates. They also proved that constructive control by adding/deleting candidates in Copeland voting is W[2]-complete with respect to the number of added/deleted candidates. Recently, Liu et al. [LFZL09] proved that both constructive and destructive control by adding candidates in plurality voting is W[2]-hard with respect to the number of added candidates, constructive control by adding/deleting voters in Condorcet voting is W[1]-hard, constructive control by adding voters in approval voting is W[1]-hard, and constructive control by deleting voters in approval voting is W[2]-hard. In all four voter control results they parameterized by the natural parameterization, i.e., the number of added/deleted voters.

We study Bucklin voting and fallback voting, a voting system that combines Bucklin voting with approval voting. Fallback voting is the natural voting system with an easy winner-determination procedure, that currently has the most resistances for control attacks (19 out of 22) [EPR10].

2.1 Elections and Electoral Control

An election (C,V) consists of a finite set of candidates C and a finite collection of voters V who express their preferences over the candidates in C. A voting system is a set of rules determining the winners of an election. Votes can be represented in different ways, depending on the voting system used. We say that a voter $v \in V$ has a preference *weak order* $\succcurlyeq$ on C, if $\succcurlyeq$ is *transitive* (i.e., for any three distinct candidates $x,y,z \in C$, $x \succcurlyeq y$ and $y \succcurlyeq z$ imply $x \succcurlyeq z$) and *complete* (i.e., for any two distinct candidates $x,y \in C$, either $x \succcurlyeq y$ or $y \succcurlyeq x$). $x \succcurlyeq y$ means that voter v likes x at least as much as y. If ties are excluded in the voters' preference rankings, this leads to a *linear order* or *strict ranking*, denoted by $\succ$. A strict ranking is always antisymmetric (i.e., for any two distinct candidates $x,y \in C$ either $x \succ y$ or $y \succ x$ holds, but not both at the same time) and irreflexive (i.e., for each $x \in C$ the following does not hold: $x \succ x$). In this paper we will write $x \; y$, instead of $x \succ y$.

Definition 2.1. *Let (C,V) be an election with $\|C\| = m$ and $\|V\| = n$. Define the* strict majority threshold *(SMT, for short) as the value $M_t = \lfloor n/2 \rfloor + 1$. In* Bucklin voting *every voter $v \in V$ has to provide a strict ranking.*

The votes of a voter v are represented as a list of all candidates, where the leftmost candidate is v's most preferred candidate, the second candidate from left is v's second most preferred candidate and so on. In our constructions, we sometimes also insert a subset $B \subseteq C$ into such votes, where we assume some arbitrary, fixed order of the candidates in B (e.g., " $c_1 \; B \; c_5$ " means that c_1 is the voter's favourite candidate, c_5 is the voter's most despised candidate and all $b \in B$ are in between these two candidates). Let $score^i_{(C,V)}(c)$ denote the number of voters who rank candidate c on level i or higher in election (C,V). Define the Bucklin score *of candidate c as $score_B(c) = min\{i \mid score^i_{(C,V)}(c) \geq M_t\}$, i.e., the smallest level i where the level i score of c is at least as high as the SMT. The candidate with the lowest Bucklin score is the unique Bucklin winner of the election. If there are more than one candidates with a lowest Bucklin score, say i, then each candidate with the highest level i score is the Bucklin winner of the election.*

Note that there always exists a Bucklin winner.

Approval voting, introduced by Brams and Fishburn [BF78, BF83] is not a preference based voting system. In *approval voting* each voter has to vote "yes" or "no" for each candidate and the candidates with the most “yes” votes are the winners of the election. Clearly, approval voting completely ignores preference rankings.

Brams and Sanver [BS09] introduced two voting systems that combine preference-based with approval voting in a sense that each voter has to specify his or her approval vector and in addition has to give a strict ranking for the candidates he or she approved of. One of these systems is fallback voting.

Definition 2.2 ([BS09])**.** *Let (C,V) be an election with $\|C\| = m$ and $\|V\| = n$. Define the strict majority threshold M_t analogously as for BV. Every voter $v \in V$ has to divide the set of candidates C into two subsets $S_v \subseteq C$ indicating that v approves of all candidates in S_v and disapproves of all candidates in $C - S_v$. S_v is called v's* approval strategy*. In addition, each voter $v \in V$ provides also a strict ranking of all candidates in S_v.*

Representation of votes: Let $S_v = \{c_1, c_2, \dots, c_k\}$ for a voter v who ranks the candidates in S_v as follows. $c_1 \succ c_2 \succ \cdots \succ c_k$, where c_1 is v's most preferred candidate and c_k is v's least preferred candidate. We denote the vote v by

$$c_1 \ c_2 \ \cdots \ c_k \ | \ C - S_v,$$

where the approved candidates to the left of the approval line are ranked from left to the right and the disapproved candidates to the right of the approval line are not ranked and written as a set $C - S_v$.

Let $score_{(C,V)}(c) = \|\{v \in V \mid c \in S_v\}\|$ denote the number of voters who approve of candidate c, and let $score^i_{(C,V)}(c)$ be the level i score of c in (C,V)*, which is the number of c's approvals when ranked on position i or higher.*

Winner determination:

1. *On the first level, only the highest ranked approved candidates (if they exist) are considered in each voters' approval strategy. If there is a candidate $c \in C$ with $score^1_{(C,V)}(c) \geq M_t$ (i.e., $c \in C$ has a strict majority of approvals on this level), then c is the* (unique) level 1 FV winner of the election*, and the procedure stops.*

2. *If there is no level* 1 *winner, we ”fall back“ to the second level, where the two highest ranked approved candidates (if they exist) are considered in each voters' approval strategy. If there is exactly one candidate $c \in C$ with $score^2_{(C,V)}(c) \geq M_t$, then c is the* (unique) level 2 FV winner of the election*, and the procedure stops. If there are at least two such candidates, then every candidate with the highest level* 2 *score is a* level 2 FV winner of the election*, and the procedure stops.*

3. *If we haven't found a level* 1 *or level* 2 *FV winner, we in this way continue level by level until there is at least one candidate $c \in C$ on a level i with $score^i_{(C,V)}(c) \geq M_t$, If there is only one such candidate, he or she is the* (unique) level i FV winner of the election*, and the procedure stops. If there are at least two such candidates, then every candidate with the highest level i score is a* level i FV winner of the election*, and the procedure stops.*

4. *If for no $i \leq \|C\|$ there is a level i FV winner, every candidate with the highest $score_{(C,V)}(c)$ is a* FV winner of (C,V) by score.

Note that BV is a special case of FV, where each voter approves of each candidate. Although BV and FV seem to be alike, there are significant differences between them. A voting system is said to be *majority-consistent* if the winner of the election is always the majority winner, whenever one exists. (A majority winner is the candidate who gets ranked first by a strict majority of voters.) Clearly, BV is majority-consistent, if a majority winner exists he or she is also the unique level 1 Bucklin winner of the election. In contrast, FV is not majority-consistent. Consider the following election with three voters and two candidates: $v_1 = \ a \ | \ b$, $v_2 = \ | \ b \ a$, and $v_3 = \ | \ b \ a$. The FV winner of this election is candidate a by score but the majority winner would be candidate b.

We now formally define the computational problems that we study in our paper. In our paper we only consider the unique-winner model, where we want to have exactly one winner. We consider two different control types. In *constructive* control scenarios, introduced by Bartholdi, Tovey, and Trick [BTT92], the chair seeks to make his or her favourite candidate win the election. In a *destructive* control scenario, introduced by Hemaspaandra, Hemaspaandra, and Rothe [HHR07], the chair's goal is to prevent a despised candidate from winning the election. We will only state the constructive cases. The questions in the destructive cases can be asked similarly with the difference that we want the distinguished candidate *not to be* a unique winner.

We first define control via adding a limited number of candidates.

Name Control by Adding a Limited Number of Candidates.
Instance An election $(C \cup D, V)$, where C is the set of qualified candidates and D is the set of spoiler candidates, a designated candidate $c \in C$, and a positive integer k.
Parameter k.
Question Is it possible to choose a subset $D' \subseteq D$ with $||D'|| \leq k$ such that c is the unique winner of election $(C \cup D', V)$?

In the following control scenario, the chair seeks to reach his or her goal by deleting (up to a given number of) candidates.

Name Control by Deleting Candidates.
Instance An election (C, V), a designated candidate $c \in C$, and a positive integer k.
Parameter k.
Question Is it possible to delete up to k candidates (other than c) from C such that c is the unique winner of the resulting election?

Turning to voter control, we first specify the problem control by adding voters.

Name Control by Adding Voters.
Instance An election $(C, V \cup W)$, where V is the set of registered voters and W is the set of unregistered voters, a designated candidate $c \in C$, and a positive integer k.
Parameter k.
Question Is it possible to choose a subset $W' \subseteq W$ with $||W'|| \leq k$ such that c is the unique winner of election $(C, V \cup W')$?

Finally, the last problem we consider, control by deleting voters.

Name Control by Deleting Voters.
Instance An election (C, V), a designated candidate $c \in C$, and a positive integer k.
Parameter k.
Question Is it possible to delete up to k voters from V such that c is the unique winner of the resulting election?

The above defined problems are all natural problems, see the discussions in [BEH$^+$09, BTT92, HHR07, FHHR09a, HHR09].

2.2 Parameterized Complexity

The theory of parameterized complexity offers toolkits for two tasks: (1) the fine-grained analysis of the sources of the computational complexity of NP-hard problems, according to secondary measurements (the *parameter*) of problem inputs (apart from the overall input size n), and (2) algorithmic methods for exploiting parameters that contribute favorably to problem complexity. Formally, a parameterized decision problem is a language $\mathscr{L} \subseteq \Sigma^* \times N$. $\mathscr{L}$ is *fixed-parameter tractable* (FPT) if and only if it can be determined, for input (x,k) of size $n = |(x,k)|$, whether $(x,k) \in \mathscr{L}$ in time $O(f(k)n^c)$, for some computable function f.

A parameterized problem $\mathscr{L}$ *reduces* to a parameterized problem $\mathscr{L}'$ if there (x,k) can be transformed to (x',k') in FPT time so that $(x,k) \in \mathscr{L}$ if and only if $(x',k') \in \mathscr{L}'$, where $k' = g(k)$ (that is, k' depends only on k).

The main hierarchy of parameterized complexity classes is

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq W[P] \subseteq XP.$$

$W[1]$ is a strong analog of NP, as the k-Step Halting Problem for Nondeterministic Turing Machines is complete for $W[1]$ under the above notion of parameterized problem reducibility. The k-Clique problem is complete for $W[1]$, and the parameterized Dominating Set problem is complete for $W[2]$. These two parameterized problems are frequent sources of reductions that show likely parameterized intractability. See the Downey-Fellows [DF99] monograph for further background.

2.3 Graphs

Many problems proven to be W[2]-hard are derived from problems concerning graphs. We will prove W[2]-hardness via parameterized reduction from the problem Dominating Set, which was proved to be W[2]-complete by Downey and Fellows [DF99]. Before the formal definition of the Dominating Set problem, we first have to present some basic notions from graph theory.

An *undirected graph* G is a pair $G = (V,E)$, where $V = \{v_1,\ldots,v_n\}$ is a finite (nonempty) set of vertices and $E = \{\{v_i,v_j\} \mid 1 \leq i < j \leq n\}$ is a set of edges.[2] Any two vertices connected by an edge are called *adjacent*. The vertices adjacent to a vertex v are called the *neighbours* of v, and the set of all neighbours of v is denoted by $N[v]$ (i.e., $N[v] = \{u \in V \mid \{u,v\} \in E\}$). The *closed neighbourhood* of v is defined as $N_c[v] = N[v] \cup \{v\}$. The parameterized version of Dominating Set is defined as follows.

Name Dominating Set.
Instance A graph $G = (V,E)$, where V is the set of vertices and E is the set of edges.
Parameter A positive integer k.
Question Does G have a dominating set of size k (i.e., a subset $V' \subseteq V$ with $||V'|| \leq k$ such that for all $u \in V - V'$ there is a $v \in V'$ such that $\{u,v\} \in E$)?

3 Results

Table 1 shows our results on the parameterized control complexity of FV and BV. The FPT results in Table 1 are in parenthesis because the two results for FV are trivially inherited from the classical P results given by Erdélyi and Rothe [ER10], and since BV is a special case of FV, BV inherits the FPT upper bound from FV in both destructive voter cases. We won't prove the W[2]-hardness results for FV, since BV is a special case of FV, FV inherits the W[2]-hardness lower bound from BV in all six cases.

In all of our results we will prove W[2]-hardness by parameterized reduction from the W[2]-complete problem Dominating Set defined in Section 2.3. In these six proofs we will always start from a given Dominating Set instance $(G = (B,E),k)$, where $B = \{b_1,b_2,\ldots,b_n\}$ is the set of vertices with $n > 2$,[3] E the set of

[2]In this paper we will use the symbol V strictly for voters. From the next section on, we will use the symbol B instead of V for the set of vertices in a graph G.

[3]Note that the assumption $n > 2$ can be made without loss of generality, since the problem Dominating Set remains W[2]-complete.

Control by	Fallback Voting		Bucklin	
	Constructive	Destructive	Constructive	Destructive
Adding a Limited Number of Candidates	**W[2]-hard**	**W[2]-hard**	**W[2]-hard**	**W[2]-hard**
Deleting Candidates	**W[2]-hard**	**W[2]-hard**	**W[2]-hard**	**W[2]-hard**
Adding Voters	**W[2]-hard**	**(FPT)**	**W[2]-hard**	**(FPT)**
Deleting Voters	**W[2]-hard**	**(FPT)**	**W[2]-hard**	**(FPT)**

Table 1: Overview of results.

edges in graph G, and $k \leq n$ is a positive integer. In the following constructions, the set of candidates will always contain the set B which means that for each vertex $b_i \in B$ we will have a candidate b_i in our election. We will also refer to candidate set $N_c[b_i]$, which is the set of candidates corresponding to the vertices in G that are in $N_c[b_i]$.

3.1 Candidate Control

Theorem 3.1. *Both constructive and destructive control by adding candidates in BV are* W[2]*-hard.*

Proof. We first prove W[2]-hardness of constructive control by adding candidates. Let $(G = (B,E),k)$ be a given instance of Dominating Set as described above. Define the election (C,V), where $C = \{c,w\} \cup B \cup X \cup Y \cup Z$ with $X = \{x_1,x_2,\dots,x_{n-1}\}$, $Y = \{y_1,y_2,\dots,y_{n-2}\}$, $Z = \{z_1,z_2,\dots,z_{n-1}\}$ is the set of candidates, w is the distinguished candidate, and V is the following collection of $2n+1$ voters:

1. For each i, $1 \leq i \leq n$, there is one voter of the form:
$$N_c[b_i] \;\; X \;\; c \;\; ((B - N_c[b_i]) \cup Y \cup Z \cup \{w\}).$$
2. There are n voters of the form:
$$Y \;\; c \;\; w \;\; (B \cup X \cup Z).$$
3. There is one voter of the form:
$$Z \;\; w \;\; (B \cup X \cup Y \cup \{c\}).$$

Note that candidate w is not a unique Bucklin winner of the election $(C-B,V)$, since only candidates c and w reach the SMT until level n (namely, exactly on level n) with $score^n_{(C-B,V)}(w) = n+1 < 2n = score^n_{(C-B,V)}(c)$ thus, c is the unique level n Bucklin winner of the election $(C-B,V)$. Now, let $C-B$ be the set of qualified candidates and let B be the set of spoiler candidates.

We claim that G has a dominating set of size k if and only if w can be made the unique Bucklin winner by adding at most k candidates.

From left to right: Suppose G has a dominating set of size k. Add the corresponding candidates to the election. Now candidate c gets pushed at least one position to the right in each of the n votes in the first voter group. Thus, candidate w is the unique Bucklin winner of the election, since w is the only candidate on level n who passes the SMT.

From right to left: Suppose w can be made the unique Bucklin winner by adding at most k candidates denoted by B'. By adding candidates from candidate set B, only votes in voter group 1 are changed. Note that candidate c has already a score of n on level $n-1$ in voter group 2 thus, c cannot have any more approvals until level n (else, $score^n_{((C-B)\cup B',V)}(c) \geq n+1$ so, c would tie or beat w on level n). This is possible only if candidate c is pushed in all votes in voter group 1 at least one position to the right. This, however, is possible only if G has a dominating set of size k.

For the W[2]-hardness proof in the destructive case, we have to do minor changes to the above construction, and we will change the roles of candidates c and w[4]. Let $(G=(B,E),k)$ be a given instance of Dominating Set as described above. Define the election (C,V), where $C=\{c,w\}\cup B\cup X\cup Y\cup Z$ with $X=\{x_1,x_2,\ldots,x_{n-1}\}$, $Y=\{y_1,y_2,\ldots,y_{n-2}\}$, $Z=\{z_1,z_2,\ldots,z_{n-2}\}$ is the set of candidates, c is the distinguished candidate, and V is the following collection of $2n+1$ voters:

1. For each i, $1\leq i\leq n$, there is one voter of the form:

$$N_c[b_i]\ \ X\ \ c\ \ ((B-N_c[b_i])\cup Y\cup Z\cup\{w\}).$$

2. There are n voters of the form:

$$Y\ \ c\ \ w\ \ (B\cup X\cup Z).$$

3. There is one voter of the form:

$$Z\ \ w\ \ c\ \ (B\cup X\cup Y).$$

Note that again only candidates c and w pass the SMT until level n in election $(C-B,V)$, both passing it on level n with $score^n_{(C-B,V)}(w)=n+1<2n+1=score^n_{(C-B,V)}(c)$ thus, c is the unique Bucklin winner of the election $(C-B,V)$. Again, let $C-B$ be the set of qualified candidates and let B be the set of spoiler candidates.

We claim that G has a dominating set of size k if and only if c can be prevented from being a unique Bucklin winner by adding at most k candidates.

From left to right: Suppose G has a dominating set B' of size k. Add the corresponding candidates to the election. Now candidate c gets pushed at least one position to the right in each of the n votes in the first voter group. Thus, on level $n-1$ none of the candidates pass the SMT, and $score^n_{((C-B)\cup B',V)}(c)=n+1=score^n_{((C-B)\cup B',V)}(w)$, i.e., both candidates c and w reach the SMT exactly on level n, and since their level n score is equal, c is not a unique Bucklin winner of the election anymore.

From right to left: Suppose c can be prevented from being a unique Bucklin winner by adding at most k candidates denoted by B'. By a similar argument as in the constructive case, this is possible only if G has a dominating set of size k. ❑

Theorem 3.2. *Both constructive and destructive control by deleting candidates in BV are* W[2]*-hard.*

Proof. We will start with the W[2]-hardness proof in the constructive case. Let $(G=(B,E),k)$ be a given instance of Dominating Set. Define the election (C,V), where $C=\{c,w\}\cup B\cup X\cup Y\cup Z$ with $X=\{x_1,x_2,\ldots,x_{n^2-\sum_{i=1}^{n}||N_c[b_i]||}\}$, $Y=\{y_1,y_2,\ldots,y_{n-1}\}$, $Z=\{z_1,z_2,\ldots,z_{n-2}\}$ is the set of candidates, w is the distinguished candidate, and V is the following collection of $2n+1$ voters:

1. For each i, $1\leq i\leq n$, there is one voter of the form:

$$N_c[b_i]\ \ X_i\ \ w\ \ ((B-N_c[b_i])\cup(X-X_i)\cup Y\cup Z\cup\{c\}),$$

   where $X_i=\{x_{1+(i-1)n-\sum_{j=1}^{i-1}||N_c[b_j]||},\ldots,x_{in-\sum_{j=1}^{i}||N_c[b_j]||}\}$.

2. There are $n-1$ voters of the form:

$$Y\ \ c\ \ (B\cup X\cup Z\cup\{w\}).$$

[4]Here, changing the roles of c and w means simply that now not candidate w but c is the distinguished candidate.

3. There is one voter of the form:

$$(Y - \{y_1\}) \;\; c \;\; w \;\; (B \cup X \cup Z \cup \{y_1\}).$$

4. There is one voter of the form:

$$Z \;\; w \;\; c \;\; (B \cup X \cup Y).$$

Note that candidate c is the unique level n Bucklin winner of the election (C,V), since only c passes the SMT on level n among all candidates.

We claim that G has a dominating set of size k if and only if w can be made the unique Bucklin winner by deleting at most k candidates.

From left to right: Suppose G has a dominating set $B' \subseteq B$ of size k. Delete the corresponding candidates. Now candidate w gets pushed at least one position to the left in each of the n votes in the first voter group. Since candidate c reaches the SMT on level n and $score^n_{(C-B',V)}(w) = n+2 > n+1 = score^n_{(C-B',V)}(c)$, and no other candidate passes the SMT until level n, candidate w is the unique Bucklin winner of the resulting election.

From right to left: Suppose w can be made the unique Bucklin winner of the election by deleting at most k candidates. Since candidate c already passes the SMT on level n, w has to beat c no later than on level n. This is possible only if candidate w is pushed in all votes in voter group 1 at least one position to the left. This, however, is possible only if G has a dominating set of size k.

For the W[2]-hardness proof in the destructive case in Bucklin, let $(G = (B,E),k)$ be a given instance of Dominating Set. Define the election (C,V), where $C = \{c,w\} \cup B \cup M \cup X \cup Y_1 \cup Y_2 \cup Z_1 \cup Z_2$ with $M = \{m_1, m_2 \ldots, m_k\}$, $X = \{x_1, x_2, \ldots, x_{n^2 - \sum_{i=1}^{n} ||N_c[b_i]||}\}$, $Y_1 = \{y_{1,1}, y_{1,2}, \ldots, y_{1,n-1}\}$, $Y_2 = \{y_{2,1}, y_{2,2}, \ldots, y_{2,k}\}$, $Z_1 = \{z_{1,1}, z_{1,2}, \ldots, z_{1,n-2}\}$, $Z_2 = \{z_{2,1}, z_{2,2}, \ldots, z_{2,n-2}\}$ is the set of candidates, c is the distinguished candidate, and V is the following collection of $2n+1$ voters:

1. For each i, $1 \leq i \leq n$, there is one voter of the form:

   $$N_c[b_i] \;\; X_i \;\; w \;\; M \;\; ((B - N_c[b_i]) \cup (X - X_i) \cup Y_1 \cup Y_2 \cup Z_1 \cup Z_2 \cup \{c\}),$$

   where $X_i = \{x_{1+(i-1)n - \sum_{j=1}^{i-1} ||N_c[b_j]||}, \ldots, x_{in - \sum_{j=1}^{i} ||N_c[b_j]||}\}$.

2. There are n voters of the form:

   $$Y_1 \;\; c \;\; Y_2 \;\; (B \cup M \cup X \cup Z_1 \cup Z_2 \cup \{w\}).$$

3. There is one voter of the form:

   $$Z_1 \;\; w \;\; c \;\; Z_2 \;\; (B \cup M \cup X_1 \cup X_2 \cup Y_1 \cup Y_2).$$

Note that candidate c is the unique level n Bucklin winner of the election (C,V), since only c passes the SMT on level n among all candidates.

We claim that G has a dominating set of size k if and only if c can be prevented from being a unique Bucklin winner by deleting at most k candidates.

From left to right: Suppose G has a dominating set $B' \subseteq B$ of size k. Delete the corresponding candidates. Now candidate w gets pushed at least one position to the left in each of the n votes in the first voter group. Since candidate c passes the SMT no earlier than on level n and $score^n_{(C-B',V)}(w) = n+1 = score^n_{(C-B',V)}(c)$, candidate c is not a unique Bucklin winner of the resulting election anymore.

From right to left: Suppose c can be prevented from being a unique Bucklin winner of the election by deleting at most k candidates. Note that deleting one candidate from an election can move the strict majority level of another candidate at most one level to the left. Observe that only candidate w can prevent c from winning the election, since w is the only candidate other than c who passes the SMT until level $n+k$. In election (C,V), candidate w passes the SMT no earlier than on level $n+1$, candidate c not before level n. Candidate w could only prevent c from winning by reaching the SMT no later than on level n. This is possible only if candidate w is pushed in all votes in voter group 1 at least one position to the left. This, however, is possible only if G has a dominating set of size k. ❑

Theorem 3.3. *Both constructive and destructive control by adding and deleting candidates in FV are* W[2]*-hard.*

3.2 Voter Control

Theorem 3.4. *Constructive control by adding voters in BV is* W[2]*-hard.*

Proof. Let $(G=(B,E),k)$ be a given instance of Dominating Set. Define the election $(C,V\cup W)$, where $C=B\cup\{w,x\}\cup Y\cup Z$, with $Y=\{y_1,y_2,\ldots,y_{\sum_{i=1}^{n}\|N_c[b_i]\|}\}$, $Z=\{z_1,z_2,\ldots,z_{n-1}\}$ is the set of candidates, w is the distinguished candidate, and $V\cup W$ is the following collection of $n+k-1$ voters:

1. V is the collection of $k-1$ registered voters of the form:

$$x\ \ Z\ \ B\ \ w\ \ Y.$$

2. W is the collection of unregistered voters, where for each i, $1\leq i\leq n$, there is one voter w_i of the form:

$$(B-N_c[b_i])\ \ Y_i\ \ w\ \ x\ \ (N_c[b_i]\cup(Y-Y_i)\cup Z),$$

where $Y_i=\{y_{(\sum_{j=1}^{i-1}\|N_c[b_j]\|)+1},\ldots,y_{\sum_{j=1}^{i}\|N_c[b_j]\|}\}$.

Clearly, x is the level 1 Bucklin winner of the election (C,V).

We claim that G has a dominating set of size k if and only if w can be made the unique Bucklin winner by adding at most k voters from W.

From left to right: Suppose G has a dominating set B' of size k. Add the corresponding voters from set W to the election (i.e., each voter w_i if $b_i\in B'$). Now there are $2k-1$ registered voters, thus the SMT is $M_t=k$. Since until level n only candidate w passes the SMT, namely on level n, w is the unique Bucklin winner of the resulting election.

From right to left: Suppose w can be made the unique Bucklin winner by adding at most k voters (denote these voters by W'). Note that $score^1_{(C,V\cup W')}(x)=k-1$. Since if a candidate passes the SMT on level 1, he or she is the unique winner of the election, $k-1$ cannot be the SMT. This is only possible, if $||W'||\geq k-1$. If $||W'||=k-1$ then $score^{n+1}_{(C,V\cup W')}(w)=k-1<M_t=k<score^{n+1}_{(C,V\cup W')}(x)=2k-1$. In this case candidate w couldn't be made the unique Bucklin winner of the election. Thus, $||W'||=k$. Note that $score^{n}_{(C,V\cup W')}(w)=k>k-1=score^{n}_{(C,V\cup W')}(x)$ and k is also a strict majority. Since we could make w the unique Bucklin winner of the election, none of the candidates in B can be ranked on the first n positions by each voter in W', otherwise there would exist a candidate $b\in B$ with $score^{n}_{(C,V\cup W')}(b)=k$ and b would reach the SMT on a higher level than w. This is only possible if G has a dominating set of size k. ❑

Theorem 3.5. *Constructive control by deleting voters in BV is* W[2]*-hard.*

Proof. To prove W[2]-hardness, we provide again a reduction from Dominating Set. Let $(G = (B,E),k)$ be a given instance of Dominating Set. Define the election (C,V), where $C = \{c,w\} \cup B \cup X \cup Y \cup Z$ with $X = \{x_1,\ldots,x_{\sum_{i=1}^{n} \|(B-N_c[b_i])\|}\}$, $Y = \{y_1,\ldots,y_{\sum_{i=1}^{n} \|N_c[b_i]\|}\}$, $Z = \{z_1,\ldots,z_{(k-1)(n+1)}\}$ is the set of candidates, w is the distinguished candidate, and V is the following collection of $2n+k-1$ voters:

1. For each i, $1 \le i \le n$, there is one voter v_i of the form:

$$N_c[b_i] \;\; c \;\; X_i \;\; ((B - N_c[b_i]) \cup (X - X_i) \cup Y \cup Z) \;\; w,$$

   where $X_i = \{x_{1+\sum_{j=1}^{i-1} \|(B-N_c[b_j])\|},\ldots,x_{\sum_{j=1}^{i} \|(B-N_c[b_j])\|}\}$.

2. For each i, $1 \le i \le n$, there is one voter of the form:

$$(B - N_c[b_i]) \;\; Y_i \;\; w \;\; (N_c[b_i] \cup X \cup (Y - Y_i) \cup Z \cup \{c\},$$

   where $Y_i = \{y_{1+\sum_{j=1}^{i-1} \|N_c[b_j]\|},\ldots,y_{\sum_{j=1}^{i} \|N_c[b_j]\|}\}$.

3. There are $k-1$ voters of the form:

$$c \;\; Z_i \;\; (B \cup X \cup Y \cup (Z - Z_i)) \;\; w,$$

   where $Z_i = \{z_{(i-1)(n+1)+1},\ldots,z_{i(n+1)}\}$.

Note that since candidate w reaches the SMT only on the last level, he or she is not the unique Bucklin winner of the election.

We claim that G has a dominating set of size k if and only if w can be made the unique Bucklin winner by deleting at most k voters.

From left to right: Suppose G has a dominating set B' of size k. Delete the corresponding voters from the first voter group (i.e., each voter v_i if $b_i \in B'$). Let V' denote the new set of voters. Now on level $n+1$ only candidate w passes the SMT, namely with $score^{n+1}_{(C,V')}(w) = n = M_t$. Thus, w is the unique Bucklin winner of the resulting election.

From right to left: Suppose w can be made the unique Bucklin winner by deleting at most k voters. Observe that deleting less than k voters would make it impossible for candidate w to be the unique winner of the election. In that case the SMT $M_t > n$ and since w is ranked last place in all votes except of n votes, he would reach the SMT on the last level thus, would not be the unique Bucklin winner of the election. Clearly, w has to win the election on level $n+1$. Now, since for all i with $1 \le i \le n$ $score^{n+1}_{(C,V)}(b_i) = n = score^{n+1}_{(C,V)}(w)$, each b_i had to loose at least one point on the first $n+1$ levels. Obviously, we cannot delete voters from the second voter group, else candidate w wouldn't reach the SMT on level $n+1$. So the k voters were deleted from the first voter group. Since each candidate b_i has lost at least one point, this is only possible if G has a dominating set of size k. ❑

Theorem 3.6. *Both constructive control by adding and deleting voters in FV are* W[2]*-hard.*

4 Conclusions and Open Questions

In this paper we have studied the parameterized complexity of the control problems for the recently proposed system of *fallback voting* and of *Bucklin voting*, parameterized by the amount of action taken by the chair.

In the case of constructive control, all of the problems are W[2]-hard. A natural question to investigate is whether these problems remain intractable when parameterized by both the amount of action and some other measure. We have shown that all four problems of constructive and destructive control by adding or deleting candidates are hard for W[2]. What is the complexity when the parameter is both the amount of action and the number of voters? We have also shown that both constructive control by adding and deleting voters are hard for W[2] in both fallback voting and Bucklin voting, and that both destructive control by adding and deleting voters are in FPT in both fallback voting and Bucklin voting. What is the complexity of constructive control parameterized by both the amount of action and the number of candidates?

Acknowledgments: We thank the anonymous COMSOC-2010 referees for their helpful comments on the preliminary version of this paper.

References

[BEH+09] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. Technical Report TR-944, Department of Computer Science, University of Rochester, Rochester, NY, May 2009. To appear in *Handbook of Approval Voting*, J. Laslier and R. Sanver, editors, Springer.

[BF78] S. Brams and P. Fishburn. Approval voting. *American Political Science Review*, 72(3):831–847, 1978.

[BF83] S. Brams and P. Fishburn. *Approval Voting*. Birkhäuser, Boston, 1983.

[BO91] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[BS09] S. Brams and R. Sanver. Voting systems that combine approval and preference. In S. Brams, W. Gehrlein, and F. Roberts, editors, *The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn*, pages 215–237. Springer, 2009.

[BTT89] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT92] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Comput. Modelling*, 16(8/9):27–40, 1992.

[BU08] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. In *Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications*, pages 43–53. Springer-Verlag *Lecture Notes in Computer Science #5165*, August 2008.

[CSL07] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[DF99] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[ENR09] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, 55(4):425–443, 2009.

[EPR10] G. Erdélyi, L. Piras, and J. Rothe. Control complexity in fallback voting. Technical Report arXiv:1004.3398v1 [cs.GT], ACM Computing Research Repository (CoRR), April 2010. A precursor appeared as [ER10].

[ER10] G. Erdélyi and J. Rothe. Control complexity in fallback voting. In *Proceedings of Computing: the 16th Australasian Theory Symposium*, pages 39–48. Australian Computer Society *Conferences in Research and Practice in Information Technology Series*, January 2010.

[FHH09] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[FHHR09a] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[FHHR09b] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 118–127. ACM Press, July 2009.

[HH07] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

[HHR07] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[HHR09] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[LFZL09] H. Liu, H. Feng, D. Zhu, and J. Luan. Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, 410(27-29):2746–2753, 2009.

Gábor Erdélyi
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1
40225 Düsseldorf, Germany
Email: `erdelyi@cs.uni-duesseldorf.de`

Michael R. Fellows
PC Research Unit, Office of DVC (Research)
University of Newcastle
Callaghan, NSW 2308, Australia
Email: `michael.fellows@newcastle.edu.au`

Cloning in Elections[1]

Edith Elkind, Piotr Faliszewski, and Arkadii Slinko

Abstract

We consider the problem of manipulating elections via cloning candidates. In our model, a manipulator can replace each candidate c by one or more *clones*, i.e., new candidates that are so similar to c that each voter simply replaces c in his vote with the block of c's clones. The outcome of the resulting election may then depend on how each voter orders the clones within the block. We formalize what it means for a cloning manipulation to be successful (which turns out to be a surprisingly delicate issue), and, for a number of prominent voting rules, characterize the preference profiles for which a successful cloning manipulation exists. We also consider the model where there is a cost associated with producing each clone, and study the complexity of finding a minimum-cost cloning manipulation. Finally, we compare cloning with the related problem of control via adding candidates.

1 Introduction

In real-life elections with more than two candidates, the winner does not always have broad political support. This is possible, for example, when the opposing views are represented by several relatively similar candidates, and therefore the vote in favor of the opposition gets "split". For example, it is widely believed that in the 2000 U.S. Presidential election spoiler candidate Ralph Nader have split votes away from Democratic candidate Al Gore allowing Republican candidate George W. Bush to win.

One can also imagine scenarios where having several similar candidates may bias the outcome in their favor. For example, suppose that an electronics website runs a competition for the best digital camera by asking consumers to vote for their two favorite models from a given list. If the list contains one model of each brand, and half of the consumers prefer Sony to Nikon to Kodak, while the remaining consumers prefer Kodak to Nikon to Sony, then Nikon will win the competition. On the other hand, if each brand is represented by several similar models, then the "Sony" customers are likely to vote for two models of Sony, the "Kodak" customers are likely to vote for two models of Kodak, and Nikon receives no votes.

The above-described scenarios present an opportunity for a party that is interested in manipulating the outcome of an election. Such a party—most likely, a campaign manager for one of the candidates—may invest in creating "clones" of one or more candidates in order to make its most preferred candidate (or one of its "clones") win the election. A natural question, then, is which voting rules are resistant to such manipulation, and whether the manipulator can compute the optimal cloning strategy for a given election.

The first study of cloning was undertaken by Tideman [18], who introduced the concept of "independence of clones" as a criterion for voting rules. He considered a number of well-known voting rules, and discovered that among these rules, STV was the only one that satisfied this criterion. However, STV does not satisfy many other important criteria for voting rules, e.g., Condorcet consistency. Thus, Tideman [18] proposed a voting rule, the "ranked pairs rule," that was both Condorcet-consistent and independent of clones in all but a small fraction of settings. Subsequently, Zavist and Tideman [19] proposed a modification of this rule that is completely independent of clones. Later it was shown that some other voting rules, such as Schulze's rule [17], are also resistant to cloning.

[1] This paper in its preliminary form will be presented at AAAI-2010.

A related concept of composition consistency as well as its weaker version, cloning consistency, was considered by Laffond at al. [11] and by Laslier [12]. They proved that a number of tournament solutions such as the Banks Set, the Uncovered Set, the Tournament Equilibrium Set (TEQ), and the Minimal Covering Set are composition-consistent. They also demonstrated that various tournament solution concepts and voting rules such as the Top Cycle, the Slater rule, the Copeland rule, and all scoring rules are not composition-consistent.

In this paper we take a rather different perspective on cloning: Instead of looking at cloning as a manipulative action that should be prevented, we view cloning as a campaign management tool. This point of view raises a number of questions that have not been considered before(or, have not been considered from this perspective):

What does it mean for cloning to be successful? The campaign manager can produce clones of existing candidates, but the voters rank them in response. We assume that clones are similar enough to be ranked as a group by each voter; however, the order of clones in such groups is specific to a particular voter. Since the campaign manager cannot control or predict the order of clones in each voter's ranking, we assume that this order is random (that is, each voter assigns equal probability to each possible order of the cloned candidates). Thus, the success of a cloning manipulation is a random event, and we can measure it probability. Let q be some real number between 0 and 1. We say that manipulation by cloning is *q-successful* if the probability of electing the desired candidate is at least q. We focus on two extreme cases: one where no matter what the voters do, the campaign manager's preferred candidate p wins (cloning is 1-successful), and one where there is a non-zero chance that p wins (by a slight abuse of notation, we will call such cloning is 0-successful).

In which instances of elections can cloning be successful? While previous work demonstrates that many well-known voting rules are susceptible to cloning, no attempt has been made to characterize the elections in which a specific candidate can be made a winner with respect to a given voting rule by means of cloning. However, from the point of view of a campaign manager who considers cloning as one of the ways to run the campaign, such characterizations are crucial. Thus, in this paper we characterize cloning-manipulable elections for several prominent voting rules. Often, manipulable elections can be characterized in terms of well-known notions of social choice such as Pareto optimality, Condorcet loser, or Uncovered Set.

Which candidates can be cloned and to what extent? The existing work on cloning does not place any restrictions on the number of clones that can be introduced, or on which candidates can be cloned at all. On the other hand, it is clear that in practical campaign management scenarios these issues cannot be ignored: not all candidates can be cloned, and creating a clone of a given candidate may be costly. Thus, we consider settings in which each clone of each candidate comes at some cost, and we seek a least expensive successful cloning strategy. However, mostly we focus on the standard model where clones come at zero cost, and on the *unit cost* model, where all clones have the same cost.

What is the computational complexity of finding cloning strategies? Finally, we consider the computational complexity of finding successful cloning strategies. In practice, it is not sufficient to know that cloning *might* work: We need to know exactly which strategy to use. We believe that our paper is the first to consider the computational aspect of cloning. Following the line of work initiated by the seminal papers of Bartholdi, Tovey, and Trick [1, 2], we seek to establish which cloning problems are NP-hard for a given voting rule, and which are solvable in polynomial time.

One might argue that in real-life elections cloning isn't really a practical campaign management tool. After all, creating even a single clone may well be too difficult or too costly. Nonetheless, below we provide two natural examples where our model of cloning is practical and well-motivated.

First, let us consider an election in which parties nominate candidates for some position, and each party can nominate several candidates. From the point of view of the voters, especially those not following the political scene closely, candidates from the same party are perceived as clones. A party's campaign manager might attempt to strategically choose the number of candidates her party should nominate, and, in fact, she might even be able to affect the number of candidates nominated by other parties (e.g., by accusing them of not giving the voters enough choice).

Second, let us consider an environment where, as suggested by Ephrati and Rosenschein in their classic paper [6], software agents vote to choose a joint plan (that is, the candidates are possible joint plans or steps of possible joint plans). In such a system, the agents can easily come up with minor variations of the (steps of the) plan, effectively creating clones of the candidates. (Laslier [12] has given a very similar example regarding a society of agents choosing a project to implement.) In both cases, the assumption that all clones are ranked contiguously and the requirement that finding a successful cloning strategy should be computationally easy are particularly relevant and realistic.

2 Preliminaries

Given a set A of *alternatives* (also called *candidates*), a voter's *preference* R is a *linear order* over A, i.e., a total transitive antisymmetric binary relation over A. An *election* E with n voters is given by its set of alternatives A and a *preference profile* $\mathcal{R} = (R_1, \dots, R_n)$, where R_i is the preference of voter i; we write $E = (A, \mathcal{R})$. For readability, we sometimes write $\succ_i$ in place of R_i. Also, we denote by $|\mathcal{R}|$ the number of voters in the election.

A voting rule $\mathcal{F}$ is often defined as a mapping from elections with a fixed set of alternatives A to the set 2^A of all subsets of A. However, in this work, we are interested in situations where the number of alternatives may change. Thus, we require voting rules to be defined for arbitrary finite sets of alternatives and preference profiles over those alternatives. Most well-known voting rules (see below) fit this more demanding definition; for ones that do not (e.g., scoring rules), we explain how to adapt their standard definition to our setting. Thus, we say that a *voting rule* $\mathcal{F}$ is a mapping from pairs of the form $E = (A, \mathcal{R})$, where A is some finite set and $\mathcal{R}$ is a preference profile over A, to subsets of A. The elements of $\mathcal{F}(E)$ are called the *winners* of the election E. Thus, we allow an election to have more than one winner, i.e., we work with social choice correspondences (also called *non-unique winner* model.)

In this paper we consider the following voting rules (for all rules described in terms of scores the winners are the alternatives with the maximum score):

Plurality. The *Plurality score* $Sc_P(c)$ of a candidate $c \in A$ is the number of voters that rank c first.

Veto. The *Veto score* $Sc_V(c)$ of a candidate $c \in A$ is the number of voters that do not rank c last.

Borda. Given an election $(A, \mathcal{R})$ with $|\mathcal{R}| = n$, the *Borda score* $Sc_B(c)$ of a candidate $c \in A$ is given by $Sc_B(c) = \sum_{i=1}^{n} |\{a \in A \mid c \succ_i a\}|$.

k-Approval. For any $k \geq 1$, the *k-Approval score* $Sc_k(c)$ of a candidate $c \in A$ is the number of voters that rank c in the top k positions. Plurality is simply 1-Approval.

Plurality with Runoff. In the first stage, all but two candidates with the top two Plurality scores are eliminated. Then the winner is the one of the survivors that is preferred to the other one by at least half of the voters. We may need to break a tie after the first round, if more than one candidate has the best or the second best score; to this end we use the *parallel universes* tie-breaking rule [4].

Maximin. Given an election $(A, \mathcal{R})$ with $|\mathcal{R}| = n$, for any $a, c \in A$, let $W(c, a) = |\{i \mid c \succ_i a\}|$. The *Maximin score* $Sc_M(c)$ of a candidate $c \in A$ is given by $Sc_M(c) = \min_{a \in A} W(c, a)$, i.e., it is the number of votes c gets in his worst pairwise contest.

Copeland. The *Copeland score* $Sc_C(c)$ of a candidate $c \in A$ is $|\{a \mid W(c, a) > W(a, c)\}| - |\{a \mid W(a, c) > W(c, a)\}|$. This is equivalent to saying that for each candidate a, c gets 1 point if she wins the pairwise contest against a, 0.5 point if there is a tie, and 0 if she loses the contest.[2]

Many results of this paper are computational and thus we assume the reader is somewhat familiar with standard notions of computational complexity such as classes P and NP, many-one reductions, NP-hardness and NP-completeness. Our NP-hardness results typically follow by reductions from EXACT COVER BY 3-SETS problem, defined below.

Definition 2.1 ([9])**.** *An instance $(G, \mathcal{S})$ of* EXACT COVER BY 3-SETS (X3C) *is given by a ground set $G = \{g_1, \ldots, g_{3K}\}$, and a family $\mathcal{S} = \{S_1, \ldots, S_M\}$ of subsets of G, where $|S_i| = 3$ for each $i = 1, \ldots, M$. It is a "yes"-instance if there is a subfamily $\mathcal{S}' \subseteq \mathcal{S}$, $|\mathcal{S}'| = K$, such that for each $g_i \in G$ there is an $S_j \in \mathcal{S}'$ such that $g_i \in S_j$, and a "no"-instance otherwise.*

3 Our Framework

Cloning and independence of clones were previously defined in [14, 18, 19]. However, we need to modify the definition given in these papers in order to model the manipulator's intentions and the budget constraints. We will now describe our model formally.

Definition 3.1. *Let $E = (A, (R_1, \ldots, R_n))$ be an election with a set of candidates $A = \{c_1, \ldots, c_m\}$. We say that an election $E' = (A', (R'_1, \ldots, R'_n))$ is obtained from E by replacing a candidate $c_j \in A$ with k* clones *for some $k > 0$ if $A' = A \setminus \{c_j\} \cup \{c_j^{(1)}, \ldots, c_j^{(k)}\}$ and for each $i \in [n]$, R'_i is a total order over A' such that:*

(i) for any $a \in A \setminus \{c_j\}$ and any $s \in [k]$ it holds that $c_j^{(s)} \succ'_i a$ if and only if $c_j \succ_i a$;
(ii) for any $a, b \in A \setminus \{c_j\}$ it holds that $a \succ'_i b$ if and only if $a \succ_i b$.

We say that an election $E^* = (A^*, \mathcal{R}^*)$ is cloned *from an election $E = (A, \mathcal{R})$ if there is a vector of non-negative integers $(k_1, \ldots, k_m)$ such that E^* is derived from E by replacing each c_j, $j = 1, \ldots, m$, with k_j clones.*

Thus, when we clone a candidate c, we replace her with a group of new candidates that are ranked together in all voters' preferences. Observe that according to the definition above, cloning a candidate c_j once means simply changing his name to $c_j^{(1)}$ rather than producing an additional copy of c_j. While not completely intuitive, this choice of terminology simplifies some of the arguments in the rest of the paper.

The definition above is essentially equivalent to the one given in [19]; the main difference is that we explicitly model cloning of more than one candidate. However, we still need to

[2] The original Copeland rule [5] was applied to tournaments and the score was the number of wins.

introduce the two other components of our model: a definition of what it means for a cloning to be successful, and the budget.

We start with the former assuming throughout this discussion that the voting rule is fixed. We observe that the final outcome of cloning depends on the relative ranking of the clones chosen by each voter, which is not under the manipulator's control. Thus, a cloning may succeed for some orderings of the clones, but not for others. The election authorities may approach this issue from the worst-case perspective, and consider it unacceptable when a given cloning succeeds for *at least one* ordering of clones by voters. Alternatively, they can take an average-case perspective, i.e.., assume that the voters rank the clones randomly and independently, with each ordering of the clones being equally likely (due to the similarities among the clones), and consider it acceptable for a cloning manipulation to succeed with probability that does not exceed a certain threshold. On the other hand, a (cautious) manipulator would view cloning as successful only if it succeeds for *all* orderings.

Definition 3.2. *Given a positive real $0 < q \leq 1$, we say that a manipulation by cloning (or simply cloning) is q-*successful *if (a) the manipulator's preferred candidate is not a winner of the original election, and (b) a clone of the manipulator's preferred candidate is a winner of the cloned election with probability at least q.*

The two approaches discussed above are special cases of this framework. Indeed, a cloning succeeds for all orderings if and only if it is 1-successful, and it succeeds for some ordering if and only if it is q-successful for some $q > 0$ no matter how small it is; we abuse notation by referring to such cloning as 0-successful. Saying that cloning is 0-successful is equivalent to saying that the cloning would be successful if the manipulator could dictate each voter how to order the clones. We will use this observation very often as it simplifies proofs.

Observe that, according to our definition, the manipulator succeeds as long as *any one* of the clones of the preferred candidate wins. This assumption is natural if the clones represent the same company (e.g., Coke Light and Coke Zero) or political party. However, if a campaign manager has created a clone of his candidate simply by recruiting an independent candidate to run on a similar platform, he may find the outcome in which this new candidate wins less than optimal. We could instead define success as a victory by the original candidate (i.e., the clone $c^{(1)}$), but, at least for neutral voting rules, this is essentially equivalent to the previous definition. Indeed, any preference profile in which the original candidate wins can be transformed into one in which some clone wins, by switching their order in each voter's preferences so $c^{(1)}$ wins with the same probability as any other clone.

Note that our definition of q-successful cloning is similar in spirit to that of [7], where voters are bribed to increase their probabilities of voting as the briber wants.

Another issue that we need to address is that of the costs associated with cloning. Indeed, the costs are an important aspect of realistic campaign management, as the manager is always restricted by the budget of the campaign. The most general way to model the cloning costs for an election with the initial set of candidates $A = \{c_1, \dots, c_m\}$ is via a *price function* $p\colon [m] \times \mathbb{Z}^+ \to \mathbb{Z}^+ \cup \{0\} \cup \{\infty\}$, where $p(i, j)$ denotes the cost of producing the j-th copy of candidate c_i. Note that $p(i, 1)$ corresponds to not producing additional copies of i, so we require $p(i, 1) = 0$ for all $i \in [m]$. We remark that it is natural to assume that all costs are non-negative (though some of them may equal zero); the assumption that all costs are integer-valued is made for computational reasons. This is not a real restriction as monetary values are discrete.

We assume that for some positive integer t the marginal cost of introducing an additional cloned candidate becomes constant, that is, $p(i, j) = p(i, t)$ for $j > t$. This ensures that the price function is succinctly representable. Thus our cost function is in fact a mapping

$p\colon [m] \times [t] \to \mathbb{Z}^+ \cup \{0\} \cup \{\infty\}$. The two natural special cases of our model defined below—Zero Cost and Unit Cost models—satisfy this condition.

Definition 3.3. *An instance of the q-*CLONING *problem for $q \in [0, 1]$ is given by the initial set of candidates $A = \{c_1, \dots, c_m\}$, a preference profile $\mathcal{R}$ over A, a manipulator's preferred candidate $c \in A$, a parameter $t > 1$, a price function $p\colon [m] \times [t] \to \mathbb{Z}^+ \cup \{0\} \cup \{\infty\}$, a budget B, and a voting rule $\mathcal{F}$. We ask if there exists a q-successful cloning with respect to $\mathcal{F}$ that costs at most B.*

For most voting rules that we consider, it is easy to bound the number of clones needed for 0-successful or 1-successful cloning (if one exists); moreover, this bound is usually polynomial in n and m. We focus on two natural special cases of q-CLONING:

1. ZERO COST (ZC): $p(i, j) = 0$ for all $i \in [m]$, $j \in \mathbb{Z}^+$. In this case we would like to decide whether an election is manipulable at all.
2. UNIT COST (UC): $p(i, j) = 1$ for all $i \in [m]$, $j \geq 2$. This model assumes that creating each new clone has a fixed cost equal for all candidates.

We say that an election E is *q-manipulable by cloning with respect to a voting rule $\mathcal{F}$* if there is a q-successful manipulation by cloning with respect to $\mathcal{F}$ in the ZC model. Further, we say that E is *manipulable by cloning with respect to $\mathcal{F}$* if it is 0-manipulable with respect to $\mathcal{F}$, and *strongly manipulable by cloning with respect to $\mathcal{F}$* if it is 1-manipulable with respect to $\mathcal{F}$.

In the rest of the paper, we discuss the complexity of the q-CLONING problem for a number of well-known voting rules, focusing on the ZC and UC models. Clearly, hardness results for these special cases also imply hardness results for the general model. Somewhat less obviously, hardness results for the ZC q-CLONING imply hardness results for UC q-CLONING: it suffices to set $B = \infty$.

Note that for polynomial-time computable voting rules 0-CLONING is clearly in NP. After a moment's thought, we can also see that q-CLONING for such rules is in Σ_2^{p}, the second level of the polynomial hierarchy, for $q = 1$, and is in $\mathrm{NP}^{\mathrm{PP}}$ for $q \in (0, 1)$. However, in this paper we are interested in P-membership and NP-hardness results only.

4 Plurality and Similar Rules

In this section we focus on q-CLONING for Plurality, Plurality with Runoff, Veto, and Maximin. Surprisingly, these four rules exhibit very similar behavior with respect to cloning.

4.1 Plurality

We start by considering Plurality, which is arguably the simplest voting rule.

Theorem 4.1. *An election is manipulable with respect to Plurality if and only if the manipulator's preferred candidate c does not win, but is ranked first by at least one voter. Moreover, for Plurality* 0-CLONING *can be solved in linear time.*

It is not too hard to strengthen Theorem 4.1 from 0-manipulability to q-manipulability for any $q < 1$.

Theorem 4.2. *For any $q < 1$, a Plurality election is q-manipulable if and only if the manipulator's preferred candidate c does not win, but is ranked first by at least one voter. However, no election is strongly manipulable.*

4.2 Veto and Plurality with Runoff

The Veto rule exhibits extreme vulnerability to cloning.

Theorem 4.3. *Any election is strongly manipulable with respect to Veto. Moreover, for Veto both* 0-CLONING *and* 1-CLONING *can be solved in linear time.*

We now consider Plurality with Runoff. Observe first that cloning any alternative cannot change what happens in the runoff: indeed, if a beats c in their pairwise contest, a would also beat any clone of c in the runoff, and if a loses to c in their pairwise contest, a would also lose in the runoff to any clone of c. Thus, if an alternative c is a *Condorcet loser*, i.e., for any $a \in A \setminus \{c\}$ a strict majority of voters prefers a to c, then c cannot be made a winner by cloning. If it is not a Condorcet loser, then it wins at least one pairwise contest, say against w. Then, if c and w get to the runoff, c would win the election. Further, c and w have a non-zero probability to reach the runoff if both are ranked first at least once. Taken together, these two considerations lead to the following criterion.

Theorem 4.4. *An election is manipulable with respect to Plurality with Runoff if and only if*

(1) the manipulator's preferred candidate c is not a current winner, and

(2) c is not a Condorcet loser and both c and some alternative w that does not beat c in their pairwise election are ranked first by at least one voter each.

Moreover, for Plurality with Runoff 0-CLONING *can be solved in polynomial time.*

As for Plurality, we can characterize q-manipulability for $q \in [0, 1]$. The following theorem can be proved similarly to Theorem 4.2.

Theorem 4.5. *For any $q < 1$, an election is q-manipulable with respect to Plurality with Runoff if and only if it is manipulable with respect to it. However, no election is strongly manipulable.*

4.3 Maximin

Consider the following election that will be used in this section. Let $E = (A, \mathcal{R})$ with $A = \{a_1, \dots, a_k\}$, $\mathcal{R} = (R_1, \dots, R_k)$, where for $i \in [k]$ the preferences of the i-th voter are given by $a_i \succ_i a_{i+1} \succ_i \dots \succ_i a_k \succ_i a_1 \succ_i \dots \succ_i a_{i-1}$. We will refer to any election that can be obtained from E by renaming the candidates as a *k-cyclic election.* In this election, for any $i = 1, \dots, k$, there are $k-1$ voters that prefer a_{i-1} to a_i (where we assume $a_{k+1} = a_1$). Thus, the Maximin score of each candidate in A is 1. Further, this remains true if we add arbitrary candidates to the election, no matter how the voters rank the additional candidates. This means that, given a candidate $a \in A$, by cloning a and telling the voters to order the clones as in a cyclic election, we can ensure that the Maximin score of any clone of a is 1: in an election with n voters, we create n clones of a and consider the situation where the voters' preferences over those clones form an n-cyclic election. This construction enables us to prove the following result.

Theorem 4.6. *An election is manipulable by cloning with respect to Maximin if and only if the manipulator's preferred candidate c does not win, but is Pareto-optimal. Further, for Maximin* 0-CLONING *can be solved in linear time. No election is strongly manipulable*

It is not clear if one can strengthen the result of Theorem 4.6 to q-manipulability for $0 < q < 1$. This amounts to the following question: suppose that for a fixed n we randomly draw n permutations of $\{1, \dots, k\}$. Let $P(n, k)$ be the probability that for each $i \in [k]$ there is a $j \in [k]$ such that j precedes i in at least $n-1$ permutations. Is it the case that the probability $P(n, k)$ approaches 1 as $k \to \infty$? Our computations show[3] that this is unlikely to be the case. For $(n, k) = (5, 20)$ there was only one success out of 10^6 random trials and only three for $(n, k) = (5, 50)$. For both $(n, k) = (7, 20)$ and $(n, k) = (7, 50)$ not a single random trial out of 10^6 trials was successful. This means that, even if Maximin is q-manipulable for $q > 0$, the number of clones needed would be astronomical.

5 Borda, *k*-Approval, and Copeland

We now consider Borda, k-Approval, and Copeland rules, for which cloning issues get significantly more involved.

5.1 Borda Rule

For Borda rule, just as for Maximin, Pareto-optimality of the manipulator's favorite alternative is necessary and sufficient for the existence of successful manipulation by cloning. However, Borda and Maximin exhibit different behavior with respect to strong manipulability. Moreover, from the point of view of finding an optimal-cost cloning, Borda appears to be harder to deal with than Maximin.

Theorem 5.1. *An election is manipulable by cloning with respect to Borda if and only if the manipulator's preferred candidate c does not win, but is Pareto-optimal. Moreover,* UC 0-CLONING *for Borda can be solved in linear time.*

Briefly, an optimal cloning manipulation for Borda in the UC model is to clone c sufficiently many times and ask all voters to order the clones in the same way. However, for $q > 0$, cloning c is not necessarily optimal.

Strengthening Theorem 5.1 to q-manipulability for some constant q, or to strong manipulability appears to be difficult. We will first characterize the elections that can be strongly manipulated with respect to Borda by cloning the manipulator's favorite candidate.

Proposition 5.2. *An election is strongly manipulable with respect to Borda by cloning the manipulator's preferred candidate c if and only if any candidate whose Borda score is higher than that of c loses to c in a pairwise contest.*

The proof of Proposition 5.2 indicates which orderings of the clones are the most problematic for the manipulator: these are the orderings that, roughly speaking, grant each clone the same number of points. But this is exactly the expected outcome if the orderings are generated uniformly at random! Thus, our proof shows that for Borda, Pareto optimality of the manipulator's most preferred candidate c is insufficient for q-manipulability with $q > 0$ by cloning c only. However, cloning a different candidate may be a better strategy: Suppose that c is Pareto-optimal, and, moreover, the original preference profile contains a candidate c' that is ranked right under c by all voters (one can think of this candidate as an "inferior clone" of c; however, we emphasize that it is present in the original profile). Then one can show that by cloning c' sufficiently many times we can make c a winner with probability 1. However, cloning c itself does not have the same effect if the voters order the clones randomly or adversarially to the manipulator. This is illustrated by the following example.

Example 5.3. Let us consider the following Borda election: $C = \{a, b, c, d\}$, there are four voters v_1, v_2, v_3, v_4, and the preference orders of the voters are:

[3]We are grateful to Danny Chang for his help with these.

$$
\begin{array}{ll}
v_1 : a \succ c \succ b \succ d & Sc_B(a) = 9 \\
v_2 : a \succ c \succ b \succ d & Sc_B(b) = 4 \\
v_3 : a \succ c \succ b \succ d & Sc_B(c) = 8 \\
v_4 : d \succ c \succ b \succ a & Sc_B(d) = 3
\end{array}
$$

The winner here is a with 9 points. However, cloning b into three clones b_1, b_2, b_3 is a 1-manipulation in favor of c since the new score of a is 15 while the new score of c is 16, no matter how clones are ordered. At the same time, no amount of cloning of c can have the same effect. Indeed, after splitting c into $k+1$ clones, the expected score of each clone of c is $4(2 + k/2) = 8 + 2k$, whereas a's score is $9 + 3k$.

This shows that in general, we may need to clone several candidates that are placed between c and its "competitors" in a large number of votes, and determining the right candidates to clone might be difficult. Indeed, it is not clear if a 1-successful manipulation can be found in polynomial time. We thus propose determining the complexity of identifying strongly manipulable profiles with respect to Borda as an open problem.

A related question that is not answered by Theorem 5.1 is the complexity of 0-Cloning in the general cost model. Note that there is a certain similarity between this problem and that of strong manipulability: in both cases, it may be suboptimal to clone c. Indeed, for general costs, we can prove that q-Cloning is NP-hard for any rational q.

Theorem 5.4. *For Borda, q-*Cloning *in the general cost model is* NP*-hard for any $q \in [0, 1]$. Moreover, this is the case even if $p(i, j) \in \{0, 1, \infty\}$ for all $i \in [m], j \in \mathbb{Z}^+$.*

The cost function used in the proof of Theorem 5.4 is very similar to the UC model, except that we are not allowed to clone some of the alternatives.

5.2 k-Approval

Plurality, k-approval and Borda are perhaps the best-known representatives of a large family of voting rules known as *scoring rules*, i.e., rules in which each voter grants each candidate a certain number of points that depends on that candidate's position in the voter's preference order. (Formally, Plurality, k-Approval, and Borda are *families* of scoring rules.) It would be interesting to characterize scoring rules vulnerable to manipulation by cloning. Recall that one can define a scoring rule $\mathcal{F}_\mathbf{w}$ for any vector $\mathbf{w} = (w(1), \dots, w(m))$ with $w(i) \in \mathbb{R}^+ \cup \{0\}$ for $i \in [m]$ (usually, though not always, it is also required that $w(1) \geq \dots \geq w(m)$) as follows: given a preference profile $(R_1, \dots, R_n)$ over a set of alternatives A of size m, the $\mathcal{F}_\mathbf{w}$-score of each alternative $c \in A$ is given by

$$
Sc_\mathbf{w}(c) = \sum_{i=1}^{n} w(\mathrm{pos}(c, i)),
$$

where $\mathrm{pos}(c, i)$ is the *position* of c in R_i, i.e., $\mathrm{pos}(c, i) = |\{a \in A \mid a \succ_i c\}| + 1$. As usual, the winners are the alternatives with the maximum score. Note, however, that this description does not fit our definition of a voting rule, as it only works for a fixed number of alternatives. To fix this, we will now define scoring rules for infinite rather than finite vectors.

Definition 5.5. *Given a profile $(R_1, \dots, R_n)$ over a set of alternatives A and a monotone sequence $\mathbf{w} = (w(1), \dots,)$, i.e., one that satisfies either (i) $w(1) \leq w(2) \leq \dots$ or (ii) $w(1) \geq w(2) \geq \dots$, we define the $\mathcal{F}_\mathbf{w}$-score of $c \in A$ as $Sc_\mathbf{w}(c) = \sum_{i=1}^{n} w(|A| - \mathrm{pos}(c, i) + 1)$ if $\mathbf{w}$ is non-decreasing and $Sc_\mathbf{w}(c) = \sum_{i=1}^{n} w(\mathrm{pos}(c, i))$ if $\mathbf{w}$ is non-increasing. The winners under $\mathcal{F}_\mathbf{w}$ are the alternatives with the maximum $\mathcal{F}_\mathbf{w}$-score.*

Observe that the Borda rule corresponds to the non-decreasing sequence $(0, 1, 2, 3, \dots)$ and Plurality corresponds to the non-increasing sequence $(1, 0, \dots)$, i.e., we need to consider both non-increasing and non-decreasing sequences to capture well-known scoring rules.

Now, we have observed that even though both Borda and Plurality are susceptible to manipulation by cloning, they exhibit very different behavior with respect to the cloning procedure. Indeed, under Plurality the winner will suffer from cloning, while under Borda her position will usually strengthen (at least as long as we are focusing on manipulability rather than strong manipulability). Further, while no election is strongly manipulable with respect to Plurality, there is a large category of elections that are strongly manipulable with respect to Borda. Thus, an interesting research direction is to determine the relationship between the properties of the sequence $\mathbf{w}$ and the manipulability of the corresponding scoring rule (compare with the work of Hemaspaandra and Hemaspaandra [10] on voter manipulation of scoring rules).

However, this problem is far from being trivial. Indeed, we will now demonstrate that there is a family of scoring rules for which deciding whether a given election is susceptible to cloning is computationally hard. Specifically, this is the case for k-Approval for any $k \geq 2$. We start by showing this for $k = 2$; subsequently, we will generalize our result to the case $k > 2$. Our proof gives a reduction from the problem Dominating Set, defined below.

Definition 5.6. *An instance of the* Dominating Set *problem is a triple (V, E, s), where (V, E) is an undirected graph and s is an integer. We ask if there is a set $W \subseteq V$ such that (a) $|W| \leq s$ and (b) for each $v \in V$ we have $v \in W$ or $(v, w) \in E$ for some $w \in W$.*

Lemma 5.7. *For* 2*-Approval, it is* NP*-hard to decide whether a given election is manipulable by cloning.*

It is not hard to modify the construction in the proof of Lemma 5.7 for the case $k > 2$.

Theorem 5.8. *For any given $k \geq 2$, it is* NP*-hard to decide whether a given election is manipulable by cloning with respect to k-Approval.*

One can also use ideas in the proof of Theorem 5.8 to show that it is NP-hard to decide whether an election is strongly manipulable with respect to k-Approval.

Theorem 5.9. *For any given $k \geq 2$, it is* NP*-hard to decide whether a given election is strongly manipulable by cloning with respect to k-Approval.*

5.3 Copeland

For an election E with a set of candidates A, its *pairwise majority graph* is a directed graph (A, X), where X contains an edge from a to b if more than half of the voters prefer a to b; we say that a *beats* b if $(a, b) \in X$. If exactly half of the voters prefer a to b, we say that a and b are *tied* (this does not mean that their Copeland scores are equal).

For an odd number of voters, the graph (A, X) is a *tournament*, i.e., for each pair $(a, b) \in A^2$, $a \neq b$, we have either $(a, b) \in X$ or $(b, a) \in X$. In this case, we can make use of a well-known tournament solution concept of Uncovered Set [16, 8, 13], defined as follows. Given a tournament (A, X), a candidate a is said to *cover* another candidate b if a beats b as well as every other candidate beaten by b. The *Uncovered Set* of (A, X) is the set of all candidates not covered by other candidates.

It turns out that if the number of voters is odd, the Uncovered Set coincides with the set of candidates that can be made Copeland winners by cloning.

Theorem 5.10. *For any $q \in [0, 1]$, an election E with an odd number of voters is q-manipulable with respect to cloning if and only if the manipulator's preferred candidate c does not win, but is in the Uncovered Set of the pairwise majority graph of E.*

For elections with an even number of voters, the situation is significantly more complicated. The notion of Uncovered Set can be extended to pairwise majority graphs of arbitrary elections in a natural way (see, e.g. [3]): we say that u covers c if u beats c and all alternatives beaten by c, and, in addition, c loses to all alternatives that beat u. In particular, this means that u *does not* cover c if it is beaten by some alternative that is tied with c. This definition generalizes the one for the odd number of voters, However, for an even number of voters, the condition that c is in the Uncovered Set turns out to be necessary, but not sufficient for manipulability by cloning.

Example 5.11. Consider an election with $A = \{a, b, c, u, w\}$. Suppose that a beats u, u beats b, b beats w, w beats a, u and w beat c, and any other pair of candidates is tied. Note that by McGarvey theorem [15] there are voters' preferences that produce this pairwise majority graph. It is easy to see that in this election c cannot be made a winner by cloning even though it is not covered.

Instead, we can characterize cloning-manipulable profiles in terms of the properties of the induced (bipartite) subgraph of (A, X) whose vertices are, on the one hand, the candidates that are tied with c, and, on the other hand, the candidates that beat c as well as all candidates beaten by c. However, it is not clear if this characterization leads to a polynomial-time algorithm. We omit the details due to space constraints.

On the other hand, finding an optimal-cost cloning manipulation is hard even in the UC model.

Theorem 5.12. *For Copeland,* UC q-CLONING *is* NP*-hard for each* $q \in [0, 1]$.

6 Conclusions

We have provided a formal model of manipulating elections by cloning, characterized manipulable and strongly manipulable profiles for many well-known voting rules, and explored the complexity of finding a minimum-cost cloning manipulation. The grouping of voting rules according to their susceptibility to manipulation differs from most standard classifications of voting rules: e.g., scoring rules behave very differently from each other, and Maximin is more similar to Plurality than to Copeland. Future research directions include designing approximation algorithms for the minimum-cost cloning under voting rules for which this problem is known to be NP-hard, and extending our results to other voting rules.

Acknowledgements. Edith Elkind is supported by NRF Research Fellowship (NRF-RF2009-08). Piotr Faliszewski is supported by AGH University of Technology Grant no. 11.11.120.865, by Polish Ministry of Science and Higher Education grant N-N206-378637, and by Foundation for Polish Science's program Homing/Powroty. Arkadii Slinko is supported by the Faculty of Science Research and Development Fund grant 3624495/9844.

References

[1] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[2] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[3] F. Brandt and F. Fischer. Computational aspects of covering in dominance graphs. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 694–699. AAAI Press, July 2007.

[4] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. AAAI Press, July 2009.

[5] A. Copeland. A "reasonable" social welfare function. Seminar on Mathematics in Social Sciences, University of Michigan, 1951.

[6] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

[7] G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, D. Raible, and J. Rothe. The complexity of probabilistic lobbying. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 86–97. Springer-Verlag *Lecture Notes in Computer Science #5783*, October 2009.

[8] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[9] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[10] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

[11] G. Laffond, J. Laine, and J. Laslier. Composition consistent tournament solutions and social choice functions. *Social Choice and Welfare*, 13(1):75–93, 1996.

[12] J. Laslier. Rank-based choice correspondencies. *Economics Letters*, 52(3):279–286, 1996.

[13] J. Laslier. *Tournament Solutions and Majority Voting*. Springer-Verlag, 1997.

[14] J. Laslier. Aggregation of preferences with a variable set of alternatives. *Social Choice and Welfare*, 17(2):269–282, 2000.

[15] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[16] N. Miller. Graph theoretical approaches to the theory of voting. *American Journal of Political Science*, 21(4):769–803, 1977.

[17] M. Schulze. A new monotonic and clone-independent single-winner election method. *Voting Matters*, 17:9–19, 2003.

[18] T. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[19] T. Zavist and T. Tideman. Complete independence of clones in the ranked pairs rule. *Social Choice and Welfare*, 64(2):167–173, 1989.

Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
Email: `eelkind@ntu.edu.sg`

Piotr Faliszewski
Department of Computer Science
AGH University of Science and Technology
Kraków, Poland
Email: `faliszew@agh.edu.pl`

Arkadii Slinko
Department of Mathematics
University of Auckland
Auckland, New Zealand
Email: `slinko@math.auckland.ac.nz`

On Problem Kernels for Possible Winner Determination Under the *k*-Approval Protocol[1]

Nadja Betzler[2]

Abstract

The POSSIBLE WINNER problem asks whether some distinguished candidate may become the winner of an election when the given incomplete votes (partial orders) are extended into complete ones (linear orders) in a favorable way. Under the k-approval protocol, for every voter, the best k candidates of his or her preference order get one point. A candidate with maximum total number of points wins. The POSSIBLE WINNER problem for k-approval is NP-complete even if there are only two votes (and k is part of the input). In addition, it is NP-complete for every fixed $k \in \{2, \ldots, m-2\}$ with m denoting the number of candidates if the number of votes is unbounded. We investigate the parameterized complexity with respect to the combined parameter k and "number of incomplete votes" t, and with respect to the combined parameter $k' := m - k$ and t. For both cases, we use kernelization to show fixed-parameter tractability. However, we show that whereas there is a polynomial-size problem kernel with respect to (t, k'), it is very unlikely that there is a polynomial-size kernel for (t, k). We provide additional fixed-parameter algorithms for some special cases.

1 Introduction

Voting situations arise in political elections, multi-agent systems, human resource departments, etc. This includes scenarios in which one is interested in finding a small group of winners (or losers), such as awarding a small number of grants, picking out a limited number of students for a graduate school, or voting for a committee with few members. Such situations are naturally reflected by a variant of approval voting, the *k-approval* voting system, where every voter gives one point to each of the k alternatives/candidates which he or she likes best and the candidates having the most points in total win. On the one side, k-approval extends *plurality* where a voter gives one point to one candidate, that is $k = 1$, and, on the other side, it extends *veto* where a voter gives one point to all but one candidate, that is, $k' = 1$ for $k' := m - k$ and m candidates.

At a certain point in the decision making process one might face the situation that the voters have made up their minds "partially". For example, for the decision about the Nobel prize for peace in 2009, a committee member might have already known that he (or she) prefers Obama and Bono to Berlusconi, but might have not decided on the order of Obama and Bono yet. This immediately leads to the question whether, given a set of "partial preferences", a certain candidate may still win. The formalization of this question leads to the POSSIBLE WINNER problem.

The POSSIBLE WINNER problem has been introduced by Konczak and Lang [16] and since then its computational complexity has been studied for several voting systems [2, 3, 5, 18, 19]. Even for the comparatively simple k-approval voting, it turned out that POSSIBLE WINNER is NP-complete except for the special cases of plurality and veto [3], that is, for any k greater than one and smaller than the number of candidates minus one. A multivariate complexity study showed that it is NP-complete if there are only two voters when k is part of the input but fixed-parameter tractable with respect to the "number of candidates" [5]. In contrast, for the approval voting variant where each voter can assign a point to *up to k* candidates, it can easily be seen that POSSIBLE WINNER can be solved in polynomial-time. A prominent special case of POSSIBLE WINNER is the MANIPULATION

[1] To appear in *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science (MFCS 2010)*, Brno, Czech Republic, August 2010.

[2] Supported by the DFG, research project PAWS, NI 369/10.

problem, where the input consists of a set of linear orders and a set of completely unspecified votes. For k-approval, it is easy to see that MANIPULATION is solvable in polynomial time for unweighted votes but for weighted votes it is NP-complete for all fixed $k \neq 1$ [15].

The above described hardness results motivate a multivariate analysis with respect to the combined parameter "number of voters" *and* "number of candidates to which a voter gives one/zero points" for k-approval. Can we efficiently solve POSSIBLE WINNER in the case that these parameters are both small? Directly related questions are whether we can ignore or delete candidates which are not relevant for the decision process and how to identify such candidates. In this context, parameterized algorithmics [11, 17] provides the concept of kernelization by means of polynomial-time data reduction rules that "preprocess" an instance such that the size of the "reduced" instance only depends on the parameters [6, 14].

In this work, we use kernelization to show the fixed-parameter tractability of POSSIBLE WINNER for k-approval in two "symmetric" scenarios. First, we consider the combined parameter "number of incomplete votes" t *and* "number of candidates to which every voter gives zero points" $k' := m - k$ for m candidates (directly extending the veto voting system with $k' = 1$). Making use of a simple observation we show that POSSIBLE WINNER admits a polynomial-size problem kernel with respect to (t, k') and provide two algorithms: one with exponential running time factor $2^{O(k')}$ in case of constant t and one with exponential running time factor $2^{O(t)}$ in case of constant k'. Second, we consider the combined parameter t and k, where k denotes the "number of candidates to which a voter gives a point". We observe that here one cannot argue symmetrically to the first scenario. Using other arguments, we give a superexponential-size problem kernel showing the fixed-parameter tractability of POSSIBLE WINNER with respect to (t, k). For the special case of 2-approval, we give an improved polynomial-size kernel with $O(t^2)$ candidates. Using a methodology due to Bodlaender et al. [7], our main technical result shows that POSSIBLE WINNER is very unlikely to admit a polynomial-size problem kernel with respect to (t, k).

2 Preliminaries

A *linear vote* is a transitive, antisymmetric, and total relation on a set C of candidates and *partial vote* a transitive and antisymmetric relation on a set C of candidates. We use $>$ to denote the relation between candidates in a linear vote and $\succ$ to denote the relation between candidates in a partial vote. We often specify a subset $D \subseteq C$ of candidates instead of single candidates in a partial vote; for a candidate $e \in C \setminus D$ and $D = \{d_1, \ldots, d_s\}$, the meaning of "$e \succ D$" is "$\{e \succ d_1, e \succ d_2, \ldots, e \succ d_s\}$". A linear vote v^l *extends* a partial vote v^p if $v^p \subseteq v^l$, that is, for every $i, j \leq m$, from $c_i \succ c_j$ in v^p it follows that $c_i > \cdots > c_j$ in v^l. An extension E of a set of partial votes $V^p = \{v_1^p, \ldots, v_n^p\}$ is a mapping from V^p to a set of linear votes $V^l := \{v_1^l, \ldots, v_n^l\}$ such that v_i^l extends v_i^p for every i. Given a set of partial votes V^p on C, a candidate $c \in C$ is a *possible winner* if there exists a *winning extension* E, that is, c wins in E with respect to a considered voting system. For any voting system R, the underlying decision problem is defined as follows.

POSSIBLE WINNER
Given: A set of candidates C, a set of partial votes V on C, and a distinguished candidate $c \in C$.
Question: Is there an extension E of V such that c wins with respect to R in E?

We focus on the voting system *k-approval* where, given a set V of linear votes on a set C of candidates, the first k candidates within a vote get one point and the remaining candidates get zero points. For every candidate $c' \in C$, one sums up the points over all votes from V to obtain its *score* $s(c')$ and the candidates with maximum score win. We call the first k positions of a vote *one-positions* and the remaining positions *zero-positions*. All results are given for the *unique winner* case, that is, looking for a single candidate with maximum score, but can be adapted easily to hold for the "co-winner" case where several candidates may get the maximum score and all of them win.

A parameterized problem L is a subset of $\Sigma^* \times \Sigma^*$ for some finite alphabet Σ [11, 17]. An instance of a parameterized problem consists of (x, p) where p is called the parameter. We mainly consider "combined" parameters which are tuples of positive integers. A parameterized problem is *fixed-parameter tractable* if it can be solved in time $f(|p|) \cdot \mathrm{poly}(|x|)$ for a computable function f. A kernelization algorithm consists of a set of *(data) reduction rules* working as follows [6, 14, 17]. Given an instance $(x, p) \in \Sigma^* \times \Sigma^*$, they output in time polynomial in $|x| + |p|$ an instance $(x', p') \in \Sigma^* \times \Sigma^*$ such that the following two conditions hold. First, (x, p) is a yes-instance if and only if (x', p') is a yes-instance (termed *soundness*). Second, $|x'| + |p'| \leq g(|p|)$ where g is a computable function. If g is a polynomial function, then we say that the parameterized problem admits a *polynomial kernel*.

Some of the reduction rules given in this work will not directly decrease the instance size by removing candidates or votes but instead only decrease the number of possible extensions of a vote, for example, by "fixing" candidates. To *fix* a candidate at a certain position means to specify its relation to all other candidates. Clearly, a candidate may not be fixed at every position in a specific partial vote. To take this into account, an important concept is the notation of *shifting* a candidate. More precisely, we say a candidate c' can shift a candidate c'' to the left (right) in a partial vote v if $c'' \succ c'$ ($c' \succ c''$) in v, that is, setting c' to a one-position (zero-position) implies setting c'' to a one-position (zero-position) as well. For every candidate $c' \in C$ and a partial vote $v \in V$, let $L(v, c') := \{c'' \in C \mid c'' \succ c' \text{ in } v\}$ and $R(v, c') := \{c'' \in C \mid c' \succ c'' \text{ in } v\}$. Then, fixing a candidate $c' \in C$ *as good as possible* means to add $L(v, c') \succ c' \succ C \setminus (L(v, c') \cup \{c'\})$ to v. Analogously, fixing a candidate *as bad as possible* is realized by adding $C \setminus (R(v, c') \cup \{c'\}) \succ c' \succ R(v, c')$ to v. If a candidate $c' \in C$ is fixed in all partial votes, this implies that also its score $s(c')$ is fixed.

The votes of an input instance of POSSIBLE WINNER can be partitioned into a (possibly empty) set of linear votes, called V^l, and a set of proper (non-linear) partial votes, called V^p. We state all our results for the parameter $t := |V^p|$. All positive results also hold for the parameter number of total votes $n := |V^l| + |V^p|$. Due to the space restrictions, several (parts of) proofs are deferred to a full version of this work.

3 Fixed number of zero-positions

For $(m - k')$-approval with $k' < m$, k' denotes the number of zero-positions. We give a polynomial kernel with respect to (t, k') for POSSIBLE WINNER where t is the number of partial votes. In addition, we provide two parameterized algorithms for special cases.

3.1 Problem kernel

Consider a POSSIBLE WINNER instance with candidate set C, vote set $V = V^l \cup V^p$, and distinguished candidate $c \in C$ for $(m - k')$-approval. We start with a simple reduction rule that is a crucial first step for all kernelization results in this work.

Rule 1. *For every vote $v_i \in V^p$, if $|L(v_i, c)| < m - k'$, fix c as good as possible in v_i.*

The soundness and polynomial-time running time of Rule 1 is easy to verify. The condition $|L(v_i, c)| < m - k'$ is crucial since otherwise c might shift a candidate c' to a one-position whereas c is assigned to a zero position and this could cause c' to beat c. After applying Rule 1, the score of c is fixed at the maximum possible value since it makes one point in all votes in which this is possible. Now, for every candidate $c' \in C \setminus \{c\}$, by counting the points that c' makes within the linear votes V^l, compute the number of zero positions that c' must assume within the partial votes V^p such that it is beaten by c. Let this number be $z(c')$ and $Z_+ := \{c' \in C \setminus \{c\} \mid z(c') > 0\}$. Since there are only tk' zero positions in V^p, one can observe the following.

Observation 1. *In a yes-instance, $\sum_{c' \in C \setminus \{c\}} z(c') \leq tk'$ and $|Z_+| \leq tk'$.*

Initialization:
For every $D' \in \mathcal{D} \setminus \{(d_1, \ldots, d_p)\}$, set $T(0, D') = 0$.
Set $T(0, (d_1, \ldots, d_p)) = 1$.

Update:
For $0 \leq i \leq t-1$,
for every $D' = (d'_1, \ldots, d'_p) \in \mathcal{D}$,
$T(i+1, D') = 1$ if there are two candidates z_g, z_h that can take the zero-positions in v_{i+1}
and $T(i, D'') = 1$ with $D'' := \{d''_1, \ldots, d''_p\}$ and
$d''_j = d'_j$ for $j \in \{1, \ldots, q\} \setminus \{g, h\}$, $d''_g \leq d'_g + 1$, and $d''_h \leq d'_h + 1$.

Output:
"yes" if $T(t, (0, \ldots, 0)) = 1$, "no" otherwise

Figure 1: Dynamic programming algorithm for $(m-2)$-approval.

Observation 1 provides a simple upper bound for the number of candidates in Z_+. By formulating a data reduction rule bounding the number of remaining candidates and replacing the linear votes V^l by a bounded number of "equivalent votes" we can show the following theorem. The basic idea is that since a remaining candidate from $C \setminus (Z_+ \cup \{c\})$ can be set arbitrarily in every vote without beating c, it is possible to replace the set of all remaining candidates by tk'^2 "representative candidates".

Theorem 1. *For $(m-k')$-approval,* POSSIBLE WINNER *with t partial votes admits a polynomial kernel with at most $tk'^2 + tk' + 1$ candidates.*

3.2 Parameterized algorithms

We give algorithms running in $2^{O(p)} \cdot \mathrm{poly}(n, m)$ time with p denoting either k' or t where the other parameter is of constant value. Note that the kernelization from the previous subsection does not imply such running times.

Constant number of partial votes. For two partial votes, there can be at most $2k'$ candidates that must take a zero-position in a yes-instance (see Observation 1). Branching into the two possibilities of taking the zero-position in the first or in the second vote for every such candidate, results in a search tree of size $2^{2k'} = 4^{k'}$. For every "leaf" of the search tree it is easy to check if there is a corresponding extension. Using similar arguments, one arrives at the following.

Proposition 1. *For a constant number t of partial votes,* POSSIBLE WINNER *for $(m-k')$-approval can be solved in $2^{t^2k'} \cdot \mathrm{poly}(n, m)$ time.*

Constant number of zero-positions. For constant k' the existence of an algorithm with running time $2^{O(t)} \cdot \mathrm{poly}(n, m)$ seems to be less obvious than for the case of constant t. We start by giving a dynamic programming algorithm for $(m-2)$-approval. Employing an idea used in [4, Lemma 2], we show that it runs in $4^t \cdot \mathrm{poly}(n, m)$ time and space.

As in the previous subsection, fix c according to Rule 1 such that it makes the maximum possible score and let $Z_+ := \{z_1, \ldots, z_p\}$ denote the set of candidates that take at least one zero-position in a winning extension. Let $d_1, \ldots, d_p$ denote the corresponding number of zero-positions that must be assumed and let $\mathcal{D} := \{(d'_1, \ldots, d'_p) \mid 0 \leq d'_j \leq d_j \text{ for } 0 \leq j \leq p\}$. Then, the dynamic programming table T is defined by $T(i, D')$ for $1 \leq i \leq t$ and $D' = (d'_1, \ldots, d'_p) \in \mathcal{D}$. Herein, $T(i, D') = 1$ if the partial votes from $\{v_1, \ldots, v_i\}$ can be extended such that candidate z_j takes at least $d_j - d'_j$ zero-positions for $1 \leq j \leq p$; otherwise $T(i, D') = 0$. Intuitively, d'_j stands for the number of zero-positions which z_j must still take in the remaining votes $\{v_{i+1}, \ldots, v_t\}$. Clearly, if $T(t, (0, \ldots, 0)) = 1$ for an instance, then it is a yes-instance. The dynamic programming algorithm is given in Figure 1. By further extending it to work for any constant k' we can show the following.

Theorem 2. *For $(m-2)$-approval with t partial votes,* Possible Winner *can be solved in $4^t \cdot \mathrm{poly}(n,m)$ time and $O(t\cdot 4^t)$ space. For $(m-k')$-approval with t partial votes,* Possible Winner *can be solved in $2^{O(t)} \cdot \mathrm{poly}(n,m)$ time for constant k'.*

4 Fixed number of one-positions

We study Possible Winner for k-approval with respect to the combined parameter k and number t of partial votes. The problem can be considered as "filling" tk one-positions such that no candidate beats c. In the previous section, we exploited that the number of candidates that must take a zero-position is already bounded by the combined parameter t and "number of zero-positions" in a yes-instance (Observation 1). Here, we cannot argue analogously: Our combined parameter (t,k) only bounds the number of one-positions but there can be an unbounded number of candidates that may take a one-position in different winning extensions of the partial votes. Hence, we argue that if there are too many candidates that can take a one-position, then there must be several choices that lead to a valid extension. We show that it is sufficient to keep a set of "representative candidates" that can take the required one-positions if and only if this is possible for the whole set of candidates. This results in a problem kernel of super-exponential size showing fixed-parameter tractability with respect to (t,k). We complement this result by showing that it is very unlikely that there is a kernel of polynomial size. In addition, we give a polynomial kernel with $O(t^2)$ candidates for 2-approval.

4.1 Problem kernels

We first describe a kernelization approach for Possible Winner for k-approval in general and then show how to obtain a better bound on the kernel size for 2-approval.

Problem kernel for k-approval. In order to describe more complicated reduction rules, we assume that a considered instance is exhaustively reduced with respect to some simple rules. To this end, we fix the distinguished candidate c as good as possible by Rule 1 (using that $m-k'=k$). Afterwards, we apply a simple reduction rule to get rid of "irrelevant" candidates and check whether an instance is a trivial no-instance:

Rule 2. *First, for every candidate $c' \in C \setminus \{c\}$, if making one point in the partial votes causes c' not to be beaten by c, then fix c' as bad as possible in every vote. Second, compute the set D of candidates that can be deleted: For every candidate $c' \in C \setminus \{c\}$ with $|L(v,c')| > k$ for all $v \in V^p$, if the score $s(c')$ is at least $s(c)$, then output "no solution", otherwise add c' to D. Delete D and replace V^l by an equivalent set.*

The soundness of Rule 2 is easy to see: Every candidate fixed by the first part cannot be assigned to a one-position in any winning extension. For the second part, every winning extension of an unreduced instance can easily be transformed into a winning extension for the reduced one by deleting the candidates specified by Rule 2 and *vice versa*. A set of equivalent linear votes can be found according to [3, Lemma 1][3].

In the following, we assume that Rule 2 has been applied, that is, all remaining candidates can make at least one point in an extension without beating c. To state further reduction rules, a partial vote v is represented as a digraph with vertex set $\{c' \mid c' \in C \setminus \{c\}$ and $|L(v,c')| < k\}$. All other candidates are considered as "irrelevant" for this vote since they cannot take a one-position. The vertices are organized into k levels. For $0 \le j \le k-1$, let $L_j(v) := \{c' \mid c' \in C \setminus \{c\}$ and $|L(v,c')| = j\}$ containing all candidates that shift exactly j candidates to a one-position if they are assigned to the best possible position. There is a directed arc from c' to c'' if and only if $c'' \in L(v,c')$. Figure 2 displays an example for the representation of a partial vote for 3-approval.

[3]Herein, it might be necessary to add one new candidate. However, this will not affect the following analysis and will be discussed in more detail in the full version of this work.

$$
\begin{aligned}
v: \quad & a \succ b \succ d \succ x, \\
& e \succ f, \\
& g \succ f \succ y \succ c, \\
& a \succ h
\end{aligned}
$$

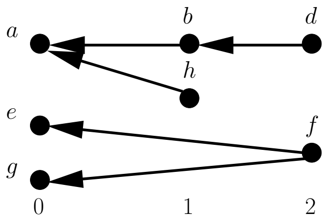


Figure 2: Example for 3-approval: Partial vote v (left-hand side) and corresponding digraph with levels 0, 1, and 2. Arcs following by transitivity are omitted. Note that x, y, and c do not appear in the digraph since they are irrelevant for v.

In general, the number of candidates per level is unbounded. However, for some cases it is easy to see that one can "delete" all but some representative candidates. The following reduction rule provides such an example using the fact that in any vote a candidate from the first level can be set to an arbitrary one-position without shifting any other candidate.

Rule 3. *For $v \in V^p$ with $|L_0(v)| \geq tk$, consider any subset $L' \subseteq L_0(v)$ with $|L'| = tk$. Add $L' \succ C \setminus L'$ to v.*

To see the soundness of Rule 3 consider a winning extension E for a non-reduced instance and a vote $v \in V^p$ with $|L_0(v)| \geq tk$. Since there are tk one-positions in the partial votes, there must be at least k candidates from L' not having assumed a one-position within the other $t-1$ votes. Setting these k candidates to the one-positions in v leads to a winning extension of the reduced instance. The other direction is obvious.

If Rule 3 applies to all partial votes, then in a reduced instance at most t^2k candidates are not fixed at zero-positions in V^p and the remaining candidates can be deleted by Rule 2. Hence, we consider the situation that there is a partial vote v with $|L_0(v)| < tk$. Then, we cannot ignore the candidates from the other levels but replace them by a bounded number of representatives. We first discuss how to find a set of representatives for 2-approval and then extend the underlying idea to work for general k.

For 2-approval, for a vote v with $|L_0(v)| < 2t$, it remains to bound the size of $L_1(v)$. This is achieved by the following reduction rule: Fix all but $2t$ in-neighbors of every candidate from $L_0(v)$ at zero-positions. To see the soundness, we show, given a winning extension E for the non-reduced instance, how to obtain a winning extension E' for v after the reduction (the other direction is obvious). Clearly, in $E(v)$ the first position must be assigned to a candidate c' from $L_0(v)$ and c' can also be assigned to the first position in $E'(v)$. If there is another candidate from $L_0(v)$ that takes the second position in $E(v)$, one can do the same in $E'(v)$. Otherwise, distinguish two cases. First, c' has less than $2t$ in-neighbors, then the reduction rule has not fixed any candidate that shifts c' to the first position and thus v can be extended in the same way as in E. Second, c' has at least $2t$ in-neighbors. Since there are only $2t$ one-positions and $2t$ non-fixed in-neighbors, the second position of v can be assigned to a candidate that does not take a one-position in any other vote of E.

Altogether, for 2-approval, one ends up with up to $4t^2$ non-fixed candidates per vote and hence with $O(t^3)$ non-reduced candidates in total. For general k, extend this approach iteratively by bounding the number of candidates for every level:

Rule 4. *Consider a partial vote $v \in V^p$ with $|L_0(v)| < tk$. Start with $i = 1$ and repeat until $i = k$.*
- For every candidate $c' \in L_i(v)$, if there are more than tk candidates in $L_i(v)$ which have the same neighborhood as c' in $L_0(v) \cup L_1(v) \cup \cdots \cup L_{i-1}(v)$, fix all but tk of them as bad as possible.
- Set $i := i + 1$.

Using Rule 4 one can show the following.

Theorem 3. *For k-approval,* Possible Winner *admits a problem kernel with size bounded by a computable function in k and the number of partial votes t.*

Improved problem kernel for 2-approval. As discussed above, the kernelization as stated for k-approval in general leads to a polynomial kernel with $O(t^3)$ candidates for 2-approval. To give a kernel with $O(t^2)$ candidates, we use some properties of bipartite graphs. For a bipartite graph $(G \cup H, E)$ with vertex set $G \cup H$ and edge set $E \subseteq \{\{g, h\} \mid g \in G$ and $h \in H\}$, a *matching* denotes a subset $M \subseteq E$ such that for all $e, e' \in M$, $e \cap e' = \emptyset$. A vertex contained in e for an $e \in M$ is called *matching vertex* and, for $\{g, h\} \in M$, g and h are *matching neighbors*. A *maximum matching* is a matching with maximum cardinality. The *open neighborhood* of a vertex $g \in G$ is denoted by $N(g) := \{h \mid \{g, h\} \in E\}$ and, for $G' \subseteq G$, $N(G') := \bigcup_{g \in G'} N(g)$.

Lemma 1. *For a bipartite graph $(G \cup H, E)$ with maximum matching M, there is a partition of G into $G_1 \uplus G_2$, such that the following holds. First, all neighbors of G_1 are part of M. Second, every vertex from G_2 has a matching neighbor outside $N(G_1)$.*

Now, we employ Lemma 1 to design a reduction rule. Note that similar arguments are used in several works, see [9, 17]. In the following, we assume that Rule 1 and Rule 2 have been applied. We define a bipartite graph $(G \cup H, E)$ as follows. For a partial profile with partial votes V^p and candidate set C, let $V' := \{v' \in V^p \mid |L_0(v')| < 2t\}$. For every $v'_i \in V'$, for $1 \leq j \leq |L_0(v'_i)|$, add a vertex g_i^j to G. Intuitively, for every candidate that can take a first position in v'_i there is a corresponding vertex in G. If a candidate can take the first position in several votes, then there are several vertices corresponding to this candidate. The vertex set H contains one vertex for every candidate from $(\bigcup_{v' \in V'} L_1(v')) \setminus (\bigcup_{v' \in V'} L_0(v'))$. There is an edge between $g_i^j \in G$ and $h \in H$ if setting the candidate corresponding to h to the second position in v'_i shifts the candidate corresponding to g_i^j to the first position. Now, we can state the following.

Rule 5. *Compute a maximum matching M in $(G \cup H, E)$. Fix every candidate corresponding to a non-matched vertex in H as bad as possible in every vote from V'.*

Lemma 2. *Rule 5 is sound and can be carried out in $O(|E| \cdot |G \cup H| + |V| \cdot |C|)$ time.*

Proof. A winning extension for an instance reduced with respect to Rule 5 is also a winning extension for an unreduced instance. Now, we show the other direction. Given a winning extension E for an unreduced instance, we construct a winning extension E_r for a reduced instance. Since Rule 5 does not fix any candidate which can take the first position in at least one vote, the first positions in E_r can be assumed by the same candidates as in E. It remains to fix the second positions without beating c. For every vote v_i, let g_i^e denote the candidate that takes the first position in v_i in E. For the corresponding vertex g_i^e one can distinguish two cases: First, $g_i^e \in G_1$. In this case, none of the neighbors of g_i^e have been fixed and, thus, the candidate which takes the second position in v_i in E can also take the second position E_r. Second, $g_i^e \in G_2$. In this case, set the candidate corresponding to the matching neighbor from g_i^e to the second position. Now, it is not to hard to see that c wins in E_r: The only candidates that possibly make more points in E_r than in E are the candidates corresponding to the matching neighbors of vertices from G_2. Due to the matching property, every such candidate makes at most one point in V'. By definition, G only contains vertices that can make at least one point and for all votes from $V^p \setminus V'$ one can easily find a winning extension which does not assign the "matching-candidates" to one-positions (see Rule 2). It follows that c also wins in the extension E_r. The claimed running time follows since a maximum bipartite matching can be found in $O(|E| \cdot |G \cup H|)$ time. ☐

Bounding the size of candidates in level 0 by Rule 3 and the (remaining) candidates in level 1 by Rule 5 one arrives at the following.

Theorem 4. *For 2-approval with t partial votes,* Possible Winner *admits a polynomial kernel with less than $4t^2$ candidates.*

4.2 Kernel lower bound

In the previous subsection, we provided a kernel of super-exponential size with respect to (t, k) for POSSIBLE WINNER under k-approval. Here, we complement this result by showing that for k-approval, POSSIBLE WINNER cannot have a polynomial kernel with respect to (t, k) under some reasonable assumptions from classical complexity theory. To this end, we apply a method introduced by Bodlaender et al. [7] and Fortnow and Santhanam [13] which is briefly described in the following.

Definition 1. *[7] A* composition algorithm *for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that receives as input a sequence $((x_1, p), \ldots, (x_q, p))$ with $(x_i, p) \in \Sigma^* \times \mathbb{N}$ for each $1 \leq i \leq q$, uses time polynomial in $\sum_{i=1}^{q} |x_i| + p$, and outputs $(y, p') \in \Sigma^* \times \mathbb{N}$ with*

- *$(y, p') \in L \Leftrightarrow (x_i, p) \in L$ for some $1 \leq i \leq q$ and*
- *p' is polynomial in p.*

A parameterized problem is *compositional* if there is a composition algorithm for it. Note that this definition directly extends to parameters that are constant-size tuples of integers. For a parameterized problem L, the *unparameterized version* L^u is the language $\{x\#1^k \mid (x, k) \in L\}$ where 1 is an arbitrary fixed letter in Σ and $\# \notin \Sigma$.

Theorem 5. *[7, 13] Let L be a compositional parameterized problem whose unparameterized version is NP-complete. Then, unless $\mathrm{coNP} \subseteq \mathrm{NP} / \mathrm{poly}$, there is no polynomial kernel for L.*

For POSSIBLE WINNER parameterized with respect to (t, k), it is easy to see that the unparameterized version is NP-complete as well. Hence, the main work to apply Theorem 5 is to achieve a composition algorithm. Composition algorithms have been provided for several fundamental combinatorial problems, see for example [8, 10]. In particular, Dom et al. [10] introduced a general framework to build composition algorithms employing so-called "identifiers". One of the necessary conditions to apply this framework, is the existence of an algorithm running in $2^{p^\gamma} \cdot \mathrm{poly}$ time for the considered parameter p and a fixed constant γ. Considering the combined parameter "number of ones" k and "number of partial votes" t for POSSIBLE WINNER under k-approval, there is no known algorithm running in $2^{(tk)^\gamma} \cdot \mathrm{poly}$ time. Hence, we apply the following overall strategy (which might be also useful for other problems).

Overall strategy. We employ a proof by contradiction. Assume that there is a polynomial kernel with respect to (t, k). Then, since for POSSIBLE WINNER there is an obvious brute-force algorithm running in $m^{tk} \cdot \mathrm{poly}(n, m)$ time for m candidates and n votes, there must be an **Algorithm S** with running time $\mathrm{poly}(t, k)^{tk} \cdot \mathrm{poly}(n, m) < 2^{(tk)^\gamma} \cdot \mathrm{poly}(n, m)$ for an appropriate constant γ. In the next paragraph, we use the existence of Algorithm S to design a composition algorithm for the combined parameter (t, k). Since it is easy to verify that the unparameterized version of POSSIBLE WINNER is NP-complete, it follows from Theorem 5 that unless $\mathrm{coNP} \subseteq \mathrm{NP} / \mathrm{poly}$ there is no polynomial kernel with respect to (t, k), a contradiction under the assumption that $\mathrm{coNP} \nsubseteq \mathrm{NP} / \mathrm{poly}$. Altogether, it remains to give a composition algorithm.

Composition algorithm. Consider a sequence $((x_1, (t, k)), \ldots, (x_q, (t, k)))$ of q POSSIBLE WINNER instances for k-approval. To simplify the construction, we make two assumptions. First, we assume that there is no "obvious no-instance", that is, an instance in which a candidate c' is not beaten by c even if c' makes zero points in all of the partial votes. This does not constitute any restriction since such instances can be found and removed in time polynomial in $\sum_{i=1}^{q} |x_i|$. Second, we assume that for $x_j, 1 \leq j \leq q$, within the partial votes the distinguished candidate makes zero points in every extension. Since it follows from known constructions [3, 5] that the unparameterized version of the problem remains NP-complete for this case, this assumption leads to a non-existence result for this special case and thus also for the general case.

The overall structure of the composition algorithm is described as follows. If $q > 2^{(tk)^\gamma}$ for γ as specified for Algorithm S, the composition algorithm applies S to every instance. This can be done within the running time bound required by Definition 1. Hence, in the following, we assume that the number of instances is at most $2^{(tk)^\gamma}$. As suggested by Dom et al. [10], this can be used to assign an "identifier" of sufficiently small size to every instance. Basically, the identifiers, which will be realized by specific sets of candidates, rely on the binary representation of the numbers from $\{1, \ldots, q\}$. The size of an identifier will be linear in $s := \lceil \log q \rceil$ which is polynomial in the combined parameter (t, k) since $q \le 2^{(tk)^\gamma}$.

Now, we provide a composition algorithm for the case that $q \le 2^{(tk)^\gamma}$. Compose the sequence of instances to one big instance

$$(X, (3s + 4, 2t)) \text{ with } X = (C, V^l \cup V^p, c)$$

as follows. For $1 \le i \le q$, let x_i be $(C_i, V_i^l \cup V_i^p, c_i)$. Then,

$$C := \biguplus_{1 \le i \le q} (C_i \setminus \{c_i\}) \uplus \{c\} \uplus D \uplus Z \uplus A \uplus B$$

with

- $D := \{d_0^0, \ldots, d_s^0\} \cup \{d_0^1, \ldots, d_s^1\}$,
- $Z := \bigcup_{1 \le j \le t} Z_j$ with $Z_j := \{z_{h,j}^0 \mid 0 \le h \le s\} \cup \{z_{h,j}^1 \mid 0 \le h \le s\}$,
- $A := \{a_1, \ldots, a_q\}$, and
- a set B with $|B| := 2s + 3 - k$.

The candidates from D and Z will be used as identifiers for the different instances. More specifically, every instance x_i is uniquely identified by the binary code of the integer $i = b_0 \cdot 2^0 + b_1 \cdot 2^1 + \cdots + b_s \cdot 2^s$ with $b_h \in \{0, 1\}$ leading to the following definition.

Definition 2. *A subset* $D_i \subset D$ identifies x_i *when* $d_h^1 \in D_i$ *if and only if* $b_h = 1$ *and* $d_h^0 \in D_i$ *if and only if* $b_h = 0$.

Let $\overline{D_i} := D \setminus D_i$. Similarly, for every $1 \le j \le t$, the set $Z_{i,j}$ denotes the candidates from Z_j that identify i, that is,

$$Z_{i,j} := \{z_{h,j}^0 \mid h \in \{0, \ldots, s\} \text{ and } b_h = 0\} \cup \{z_{h,j}^1 \mid h \in \{0, \ldots, s\} \text{ and } b_h = 1\}.$$

Let $\overline{Z_{i,j}} := Z_j \setminus Z_{i,j}$ denote the remaining candidates from Z_j.

The set of partial votes V^p consists of two subsets V_1^p and V_2^p, both containing t partial votes. The basic idea is that a winning extension of V_1^p "selects" an instance x_i and there is a winning extension for x_i if and only if V_2^p can be extended such that c wins. The set V_1^p contains the vote

$$\{Z_{i,1} \cup \overline{D_i} \cup \overline{Z_{i,t}} \succ a_i \mid 1 \le i \le q\},\ D \cup Z \cup A \succ C \setminus (D \cup Z \cup A),$$

meaning that the vote contains the constraints $Z_{i,1} \cup \overline{D_i} \cup \overline{Z_{i,t}} \succ a_i$ for every i. Furthermore, for every $j \in \{2, \ldots, t\}$, the set V_1^p contains the vote

$$\{Z_{i,j} \cup \overline{D_i} \cup \overline{Z_{i,j-1}} \succ a_i \mid 1 \le i \le q\},\ D \cup Z \cup A \succ C \setminus (D \cup Z \cup A).$$

The set V_2^p consists of the partial votes $v_1, \ldots, v_t$. Every vote $v_j \in V_2^p$ "composes" the votes v_i^j for $i \in \{1, \ldots, q\}$ where v_i^j denotes the jth vote from instance x_i after deleting c_i. Then, for $j \in \{1, \ldots, t\}$, the vote v_j is

$$B \succ (C \setminus B),\ \{v_i^j \mid 1 \le i \le q\},\ \{D_i \succ C_i \setminus \{c_i\} \mid 1 \le i \le q\},\ C \setminus (A \cup Z \cup \{c\}) \succ A \cup Z \cup \{c\}.$$

One can construct a set V_l of linear votes polynomial in $|C|$ and $|V^p|$ such that the following hold [3, Lemma 1].

$$
\begin{array}{lllllll}
V_1^p: & Z_{w,1} & > \overline{D_w} & > \overline{Z_{w,t}} & > a_w & > C \setminus (Z_{w,1} \cup \overline{D_w} \cup \overline{Z_{w,t}}) & \\
 & Z_{w,j} & > \overline{D_w} & > \overline{Z_{w,j-1}} & > a_w & > C \setminus (Z_{w,j} \cup \overline{D_w} \cup \overline{Z_{w,j-1}}) & \text{for } 2 \leq j \leq t \\
V_2^p: & B & > D_w & > w_j & & > C \setminus (B \cup D_w \cup (C_w \setminus \{c_w\})) & \text{for } 1 \leq j \leq t
\end{array}
$$

Figure 3: Extension for X in which c wins. For a winning extension $E(x_w) = w'_1, \ldots, w'_t$ of x_w, let w_j denotes the linear order given by w'_j restricted to the candidates from $C_w \setminus \{c\}$. The remaining subsets of candidates are fixed in any transitivity preserving order.

- For $i \in \{1, \ldots, q\}$, the maximum partial score of every candidate $c' \in C_i \setminus \{c_i\}$ equals the maximum partial score of c' in x_i.
- For every candidate from $A \cup D \cup B$, the maximum partial score is t.
- For every candidate from Z, the maximum partial score is one.

Lemma 3. *The constructed instance X is a yes-instance for $(3s+4)$-approval if and only if there is an $i \in \{1, \ldots, q\}$ such that x_i is a yes-instance for k-approval.*

Proof. "$\Leftarrow$": Assume there is an instance x_w for which c is a possible winner. Let $E(x_w) = w_1, \ldots, w_t$ denote a winning extension for x_w and recall that C_w denotes the set of candidates from x_w. Then, extend the partial votes from X as indicated in Figure 3. Since there are $3s+4$ one-positions per vote, $|D_i| = s+1$, and $|B| = 2s+3-k$, in every extended vote from V_2^p, there are k one-positions that are assumed by candidates from $C_w \setminus \{c_w\}$. Because of this and due to the equivalence of the partial orders in the corresponding votes, the candidates from $C_w \setminus \{c_w\}$ make exactly the same number of points in the extension for X as in $E(x_w)$ and are beaten by c. The remaining "instance candidates", namely, $\bigcup_{i \neq w} C_i \setminus \{c_i\}$ do not make any points in the given extension and thus are beaten by c. The candidates from D can be partitioned into the two disjoint subsets D_w and $\overline{D_w}$. The candidates from D_w make t points in V_2^p and zero points in V_1^p whereas the candidates from $\overline{D_w}$ make zero points in V_2^p and t points in V_1^p. Thus, all candidates from D are beaten by c. Regarding the candidates from Z_j, every candidate appears either in $Z_{w,j}$ or in $\overline{Z_{w,j}}$ and thus makes exactly one point and is beaten by c. Clearly, all candidates from $A \cup B$ are also beaten by c. Hence, c is a possible winner for X.

Finally, we briefly discuss that fixing the order within the given subsets of candidates in Figure 3 can be done without violating the restriction provided by the partial orders. For v_j in V_2^p such an extension is

$$B > D_w > w_j > \overline{D_w} > \bigcup_{i \neq j} C_i \setminus \{c_i\} > A > Z > \{c\}$$

where, the candidates from $B, D_w, \overline{D_w}, A$, and Z can be fixed in an arbitrary order since there are not any internal constraints in v_j. The remaining candidates from $\bigcup_{i \neq w} C_i \setminus \{c_i\}$ can be ordered such that $C_i \setminus \{c_i\} > C_s \setminus \{c_s\}$ for $i > s$, $i \neq w$, and $s \neq w$ and within $C_i \setminus \{c_i\}$, for every $i \neq w$, the candidates can be ordered according to any extension of v_i^j. A "complete" extension for the votes from V_1^p can be obtained similarly.

"$\Rightarrow$": Consider an extension of X in which c wins. First, by proving the following claim, we show that within V_1^p one instance x_w must be "selected".

Claim: There must be a $w \in \{1, \ldots, q\}$ such that every candidate from $\overline{D_w}$ is assigned to a one-position in every extended vote from V_1^p whereas every candidate from D_w makes zero points in V_1^p.

Proof of Claim: Since there are $3s+4$ one-positions per vote, in V_1^p there are altogether $3st+4t$ one-positions that must be filled. The candidates from Z can take at most $2st+2t$ of them since $|Z| = 2t(s+1)$ and each candidate from Z can make at most one point without beating c. By using some argumentation including the votes from V_2^p, we can show that the candidates from D can take at most $st + t$ of the one-positions in V_1^p in a winning extension: In every vote from V_2^p, by construction, the first $2s + 3 - k$ positions are assumed by candidates from B and the remaining $s+k+1$ one-positions can only be assigned to candidates from $\bigcup_{1\leq i\leq q} C_i \setminus \{c_i\} \cup D$. Since every candidate from $\bigcup_{1\leq i\leq q} C_i \setminus \{c_i\}$ shifts $s + 1$ candidates from $\overline{D}$ to the left by assuming a one-position, it directly follows that the total number of one-positions assumed by candidates from D within V_2^p is at least $t(s + 1)$. Since $|D| = 2s + 2$ and every candidate from D can make at most t points, the candidates from D can take at most $t(2s + 2) - t(s + 1) = st + t$ of the one-positions in V_1^p in a winning extension.

Summarizing, in a winning extension, in V_1^p at most $3st + 3t$ one-positions can be assigned to candidates from $D \cup Z$. Hence, at least t one-positions must be assigned to candidates from A. Furthermore, a candidate a_i from A shifts $3s+3$ candidates from $D\cup Z$ to one-positions if a_i takes a one-position. Thus, at most one candidate from A can take a one-position in an extended vote. It follows that in every vote $v_j \in V_2^p$ exactly one candidate a_i from A must take a one position thereby shifting the candidates from $Z_{i,j} \cup \overline{D_i} \cup \overline{Z_{i,j-1}}$ (or $Z_{i,1} \cup \overline{D_i} \cup \overline{Z_{i,t}}$ for $j = 1$) to one-positions.

Now, we show for $1 \leq j \leq t-1$ that if the candidate $a_w \in A$ takes a one-position in v_j, then a_w also takes a one-position in v_{j+1}. Assume that in v_j, a_w and thus also the candidates from $Z_{w,j}$ take a one-position. As discussed above, in v_{j+1} a candidate from A must shift $s + 1$ further candidates from Z_j. Since every candidate from Z can make at most one point, the set of these candidates must be disjoint from $Z_{w,j}$. The only set of candidates fulfilling this is $\overline{Z_{w,j}}$ and is shifted only by a_w. Analogously, if a_w takes a one-position in v_t, then it also must take a one-position in v_1 because of the candidates from Z_t. This finishes the proof of the Claim.

Now, as direct consequence of the Claim, within V_2^p each candidate from D_w can still make t points whereas the candidates from $\overline{D_w}$ cannot make any points without beating c. Hence, in every vote from V_2^p, we can only set candidates from C_w to the one-positions since setting any other candidates would shift a candidate from $\overline{D_w}$. This means that one can extend V_2^p such that, in every vote, k one-positions are assigned to candidates from $C_w \setminus \{c_w\}$ without beating c. Since the partial relations between the candidates in $C_w \setminus \{c_w\}$ are the same in the ith vote of x_w and X and c makes zero points in both cases, a winning extension for X directly gives a winning extension for x_w. ☐

By using Lemma 3 it is easy to verify that the given composition algorithm fulfills all requirements of Definition 1. Hence, Theorem 6 follows from our overall strategy.

Theorem 6. *For k-approval,* POSSIBLE WINNER *parameterized by the combined parameter k and "number of partial votes" does not admit a polynomial problem kernel unless* $\mathrm{NP} \subseteq \mathrm{coNP} / \mathrm{poly}$.

5 Outlook

We provided fixed-parameter tractability results based on kernelization. It seems interesting whether similar results can be obtained for "more general" problems such as SWAP BRIBERY [12] or the counting version of POSSIBLE WINNER[1]. Another interesting scenario might be as follows. Given a number s of winners in the input, for example, the size of a committee, one is interested in the s candidates such that each of them has more points than the remaining candidates. For this scenario, the negative results for POSSIBLE WINNER for k-approval as given in this work and related work [3, 5] can be adapted by adding $s - 1$ fixed candidates that always win, but as to the algorithmic results, it is open whether they extend to this scenario.

References

[1] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *Proc. of 24th AAAI*, 2010. To appear.

[2] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the Possible Winner problem in pure scoring rules. In *Proc. of 19th ECAI*, 2010. Short paper.

[3] N. Betzler and B. Dorn. Towards a complexity dichotomy of finding possible winners in elections based on scoring rules. In *Proc. of 34th MFCS*, volume 5734 of *LNCS*, pages 124–136. Springer, 2009. Longversion to appear in *J. Comput. Syst. Sci.*

[4] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. *Inform. Comput.*, 208(2):165–177, 2010.

[5] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. of 21st IJCAI*, pages 53–58, 2009.

[6] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proc. of 4th IWPEC*, volume 5917 of *LNCS*, pages 17–37. Springer, 2009.

[7] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[8] H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *Proc. of 17th ESA*, volume 5757 of *LNCS*, pages 635–646. Springer, 2009.

[9] B. Chor, M. Fellows, and D. W. Juedes. Linear kernels in linear time, or how to save k colors in $o(n^2)$ steps. In *Proc. of 30th WG*, volume 3353 of *LNCS*, pages 257–269. Springer, 2004.

[10] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In *Proc. of 36th ICALP*, volume 5555 of *LNCS*, pages 378–389. Springer, 2009.

[11] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[12] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proc. of 2nd SAGT*, volume 5814 of *LNCS*, pages 299–310. Springer, 2009.

[13] L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *Proc. of 40th STOC*, pages 133–142. ACM, 2008.

[14] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007.

[15] E. Hemaspaandra and L. A. Hemaspaandra. Dichotomy for voting systems. *J. Comput. Syst. Sci.*, 73(1):73–83, 2007.

[16] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. of IJCAI-2005 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[17] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[18] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *Proc. of 20th IJCAI*, pages 1464–1469, 2007.

[19] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proc. of 23rd AAAI*, pages 196–201. AAAI Press, 2008.

Nadja Betzler
Institut für Informatik
Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2
D-07743 Jena, Germany
Email: `nadja.betzler@uni-jena.de`

Possible Winners When New Alternatives Join: New Results Coming Up!

Lirong Xia, Jérôme Lang, and Jérôme Monnot

Abstract

In a voting system, sometimes multiple new alternatives will join the election after the voters' preferences over the initial alternatives have been revealed. Computing whether a given alternative can be a co-winner when multiple new alternatives join the election is called the *possible co-winner with new alternatives (PcWNA)* problem, introduced by Chevaleyre et al. [5, 6]. In this paper, we show that the PcWNA problems are NP-complete for the Bucklin, Copeland$_0$, and Simpson (a.k.a. maximin) rule, even when the number of new alternatives is no more than a constant. We also show that the PcWNA problem can be solved in polynomial time for plurality with runoff. For the approval rule, we define three different ways to extend a linear order with new alternatives, and characterize the computational complexity of the PcWNA problem for each of them.

1 Introduction

In many real-life situations, a set of voters have to choose a common alternative out of a set that can grow during the process. For instance, when a committee want to decide which proposal should be granted, some applications might arrive late (due to unexpected delay in mailing system, etc). Suppose that we have already elicited the preference of the voters (members in the committee) on the initial proposals. It is important for the applicants to know whether they are already out (so that they can submit the same proposal to other founding sources right away without waiting for the committee members to make the final decision). A recent paper by Chevaleyre et al. [5] considers the following problem: *suppose that the voters' preferences about a set of initial alternatives have already been elicited, and we know that a given number k of new alternatives will join the election; we ask who among the initial alternatives can possibly win the election in the end*. This problem is a special case of the *possible winner problem* [9, 12, 11, 3, 4, 2], restricted to the case where the incomplete profile consists of a collection of full rankings over the initial alternatives (nothing being known about the voters' preferences about the new alternatives), somehow dual of another special case of the problem where the incomplete profile consists of a collection of full rankings over all alternatives for a subset of voters (nothing being known about the remaining voters' preferences), which itself is equivalent to the coalitional manipulation problem. The problem is also related to control by adding candidates [1], as discussed in [5].

Chevaleyre et al. [5, 6] investigated the complexity of computing possible winners with new alternatives, and laid the focus on scoring rules, obtaining both polynomiality and NP-completeness results, depending on the scoring rule used and the number of new alternatives. Their results, however, did not go beyond scoring rules. Here we go further and give results for several other common rules, especially some common rules that are based on *pairwise elections*. After giving some background in Section 2, each of the following sections is devoted to the PcWNA problem for a specific voting rule. In Section 3, we focus on approval voting. Since the notion of a complete profile (including the new alternatives) extending a partial profile over the initial alternatives is not straightforward, we propose three possible definitions, which we think are the three most reasonable definitions. We show that PcWNA problems are trivial for two of these definitions, and NP-complete for the third one. In Sections 4, 5 and 6 we show that the problem is NP-complete for, respectively, the Bucklin rule, the Copeland rule, and the Simpson (a.k.a. maximin) rule, and finally in Section 7 we focus on plurality with runoff, for which the problem is in P (due to the space constraint, the proof of this result is omitted).

2 Preliminaries

Let $\mathcal{C}$ be the set of *alternatives* (or candidates), with $|\mathcal{C}| = m$. Let $\mathcal{I}(\mathcal{C})$ denote the set of votes. Most often, the set of votes is the set of all linear orders over $\mathcal{C}$. An *n-profile* P is a collection of n votes for some $n \in \mathbb{N}$, that is, $P \in \mathcal{I}(\mathcal{C})^n$. A *voting rule* r is a mapping that assigns to each profile a set of winning alternatives, that is, r is a mapping from $\{\emptyset\} \cup \mathcal{I}(\mathcal{C}) \cup \mathcal{I}(\mathcal{C})^2 \cup \ldots$ to $2^{\mathcal{C}}$. Some common voting rules are listed below. For all of them (except the approval rule), $\mathcal{I}(\mathcal{C})$ is the set of all linear orders over $\mathcal{C}$; for the approval rule, the set of votes is the set of all subsets of $\mathcal{C}$, that is, $\mathcal{I}(\mathcal{C}) = \{S : S \subseteq \mathcal{C}\}$.

(Positional) scoring rules: Given a *scoring vector* $\vec{v} = (v(1), \ldots, v(m))$, for any vote $V \in L(\mathcal{C})$ and any $c \in \mathcal{C}$, let $s(V, c) = v(j)$, where j is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $s(P, c) = \sum_{i=1}^{n} s(V_i, c)$. The rule will select $c \in \mathcal{C}$ so that $s(P, c)$ is maximized. Some examples of positional scoring rules are *Borda*, for which the scoring vector is $(m-1, m-2, \ldots, 0)$; *l-approval* ($l \leq m$), for which the scoring vector is $v(1) = \ldots = v(l) = 1$ and $v_{l+1} = \ldots = v_m = 0$; and *plurality*, for which the scoring vector is $(1, 0, \ldots, 0)$.

Approval: Each voter submits a set of alternatives (that is, the alternatives that are "approved" by the voter). The winner is the alternative approved by the largest number of voters. Note that the approval rule is different from the l-approval rule, in that for the l-approval rule, a voter must approve l alternatives, whereas for the approval rule, a voter can approve an arbitrary number of alternatives.

Bucklin: The Bucklin score of an alternative c is the smallest number t such that more than half of the votes rank c among top t positions. The alternatives that have the lowest Bucklin score win. (We do not consider any further tie-breaking for Bucklin.)

Copeland$_\alpha$ ($0 \leq \alpha \leq 1$): For any two alternatives c_i and c_j, we can simulate a *pairwise election* between them, by seeing how many votes prefer c_i to c_j, and how many prefer c_j to c_i; the winner of the pairwise election is the one preferred more often. Then, an alternative receives one point for each win in a pairwise election, α points for each tie, and zero point for each loss. The alternatives that have the highest scores win.

Simpson (a.k.a. maximin): Let $N_P(c_i, c_j)$ denote the number of votes that rank c_i ahead of c_j in P. The Simpson score of alternative $c \in C$ in profile P is defined as $Sim_P(c) = \min\{N_P(c, c') : c' \in C \setminus \{c\}\}$. A Simpson winner for P is an alternative $c_0 \in C$ such that $Sim_P(c_0) = \max\{Sim_P(c) : c \in C\}$.

Plurality with runoff: The election has two rounds. In the first round, all alternatives are eliminated except the two with the highest plurality scores. In the second round (runoff), the winner is the alternative that wins the pairwise election between them.

Let $\mathcal{C}$ denote the set of original alternatives, let Y denote the set of new alternatives. For any linear order V over $\mathcal{C}$, a linear order V' over $\mathcal{C} \cup \{V\}$ *extend* V, if in V', the pairwise comparison between any pair of alternatives in $\mathcal{C}$ is the same as in V. That is, for any $c, d \in \mathcal{C}$, $c \succ_V d$ if and only if $c \succ_{V'} d$.

Given a voting rule r, an alternative c, and a profile P over $\mathcal{C}$, we are asked whether there exists a profile P' over $\mathcal{C} \cup Y$ such that P' is an extension of P and $c \in r(P')$. This problem is called the *possible co-winner with new alternatives (PcWNA)* problem [5, 6].

Similarly, we let *PWNA* denote the problem in which we are asked whether c is a possible (unique) winner, that is, $r(P') = \{c\}$. Up to now, the PcWNA and PWNA problems are well-defined for all voting rules studied in this paper (except the approval rule). For the approval rule, we will introduce three types of extension, and discuss the computational complexity of the PcWNA and PWNA problems under these extensions.

In this paper, all NP-hardness results are proved by reductions from the Exact Cover by 3-Sets problem (denoted by X3C) or the 3-DIMENSIONAL MATCHING problem (denoted by 3DM). An instance $I = (\mathcal{S}, \mathcal{V})$ of X3C consists of a set $\mathcal{V} = \{v_1, \ldots, v_{3q}\}$ of $3q$ elements and $t \geq q$ 3-sets $\mathcal{S} = \{S_1, \ldots, S_t\}$ of $\mathcal{V}$, i.e., for any $i \leq t$, $S_i \subseteq \mathcal{V}$ and $|S_i| = 3$. For any $v \in \mathcal{V}$, let $d_I(v)$ denote the number of 3-sets containing element v in instance I. Let $\Delta(I) = \max_{v\in\mathcal{V}} d_I(v)$. We are asked whether there exists a subset $J \subseteq \{1, \ldots, t\}$ such that $|J| = q$ and $\bigcup_{j\in J} S_j = \mathcal{V}$ (indeed, the sets S_j for $j \in J$ form a partition of $\mathcal{V}$). This problem is known to be NP-complete, even if $\Delta(I) \leq 3$ (problem [SP2] page 221 in [8]). In this paper, we will use a special case of 3DM that is also a special case of X3C, defined as follows.[1] Given A, B, X, where $A = \{a_1, \ldots, a_q\}$, $B = \{b_1, \ldots, b_q\}$, $X = \{x_1, \ldots, x_q\}$, $T \subseteq A \times B \times X$, $T = \{S_1, \ldots, S_t\}$ with $t \geq q$. We are asked whether there exists $M \subseteq T$ such that $|M| = q$ and for any $(a_1, b_1, x_1), (a_2, b_2, x_2) \in M$, we have $a_1 \neq a_2$, $b_1 \neq b_2$, and $x_1 \neq x_2$. That is, M corresponds to an exact cover of $\mathcal{V} = A \cup B \cup X$. This problem with the restriction where no element of $A \cup B \cup X$ occurs in more than 3 triples (i.e, $\Delta(I) \leq 3$) is known to be NP-complete (problem [SP1] page 221 in [8]).

It is straightforward to check that the PcWNA (respectively, PWNA) problems for all voting rules studied in this paper are in NP, because given an extension of a profile P, it is polynomial to verify if the given alternative c is a co-winner (respectively, unique winner) for all rules studied in this paper (again, we discuss the approval rule separately). Therefore, in this paper we only show NP-hardness proofs.

To prove that the PcWNA and PWNA problems are NP-hard, we first prove that another useful special case of 3DM (as well as X3C) remains NP-complete.

Proposition 1 3DM *is* NP-*complete, even if* q *is even,* $t = 3q/2$, *and* $\Delta(I) \leq 6$.

Proof of Proposition 1: Let $I = (T, A \times B \times X)$ be an instance of 3DM with $A = \{a_1, \ldots, a_q\}$, $B = \{b_1, \ldots, b_q\}$, $X = \{x_1, \ldots, x_q\}$, $T \subseteq A \times B \times X$, $T = \{S_1, \ldots, S_t\}$ and $\Delta(I) \leq 3$. We next show how to build an instance $I' = (T', A' \times B' \times X')$ of 3DM in polynomial time, with $|A'| = |B'| = |X'| = q'$, $T' \subseteq A' \times B' \times X'$ and $|T'| = t'$ such that q' is even, $t' = 3q'/2$, and $\Delta(I') \leq 6$.

• If q is odd, then we add to the instance 3 new elements $\{a'_1, b'_1, x'_1\}$ with $A' = A \cup \{a'_1\}$, $B' = B \cup \{b'_1\}$, $X' = X \cup \{x'_1\}$ and one new triplet (a'_1, b'_1, x'_1).

• Suppose that q is even. If $t > 3q/2$, then we add $6(t - 3q/2)$ new elements $\{a'_1, \ldots, a'_{2(t-3q/2)}\}$ to A, $\{b'_1, \ldots, b'_{2(t-3q/2)}\}$ to B, $\{x'_1, \ldots, x'_{2(t-3q/2)}\}$ to X and $2(t - 3q/2)$ new triples $\{S'_1, \ldots, S'_{2(t-3q/2)}\}$, where for any $i \leq 2(t - 3q/2)$, $S'_i = (a'_i, b'_i, x'_i)$. If $t < 3q/2$, then we add $3q/2 - t$ dummy triples to T by duplicating $3q/2 - t$ triples of T once each. We note that $t \geq q$ implies that $t \geq 3q/2 - t$.

It is easy to check in I', q' is even, $t' = 3q'/2$, and $\Delta(I') \leq 6$. The size of the input of the new instance is polynomial in the size of the input of the old instance. Moreover, I is a yes-instance if and only if I' is also a yes-instance. □

3 Approval

Since the input of the approval rule is different from the input of other voting rules studied in this paper, we have to define the set of possible extensions of an approval profile over $\mathcal{C}$. Let $P_{\mathcal{C}} = (V_1, \ldots, V_n)$ be an approval profile over $\mathcal{C}$, where each V_i is a subset of $\mathcal{C}$. An extension of $P_{\mathcal{C}}$ over $\mathcal{C} \cup Y$ is a collection $(V'_1, \ldots, V'_n)$ where $V'_i \subseteq \mathcal{C} \cup Y$ is an extension of V_i. Now, we have to define what it means to say that $V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq \mathcal{C}$. We can think of three natural definitions:

[1] Generally, 3DM is not a special case of X3C.

Definition 1 (extension of an approval vote, definition 1) *$V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq \mathcal{C}$ if $V' \cap \mathcal{C} = V$.*

In other words, under this definition, V' is an extension of V if $V' = V \cup Y'$, where $Y' \subseteq Y$. This definition coincides with the definition used in [10] (namely, Definition 4.3) for the control of approval voting by adding candidates. The problem with Definition 1 is that it assumes that any alternative approved in V is still approved in V'. However, in some contexts, extending the choice with alternatives of Y may change the "approval threshold". Moreover, since we have more alternatives, this threshold should either stay the same or move upwards: some alternatives that were approved initially may become disapproved. This leads to the following definition of extension:

Definition 2 (extension of an approval vote, definition 2) *$V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq X$ if one of the following conditions holds: (1) $V = V'$; (2) $V' \cap Y \neq \emptyset$ and $V' \cap \mathcal{C} \subseteq V$.*

Lastly, we may also allow the acceptance threshold to move downwards, even though the set of alternatives grows, especially in the case where the new alternatives are particularly bad, thus rendering some alternatives in $\mathcal{C}$ acceptable after all. This leads to the third definition of extension:

Definition 3 (extension of an approval vote, definition 3) *$V' \subseteq \mathcal{C} \cup Y$ is an extension of $V \subseteq \mathcal{C}$ if one of the following conditions holds: (1) $V' \cap \mathcal{C} \subset V$ and $V' \cap Y \neq \emptyset$; (2) $V \subset V' \cap \mathcal{C}$, and $Y \setminus V' \neq \emptyset$; (3) $V' \cap \mathcal{C} = V$.*

Under Definition 3, either the threshold moves upward, in which case all alternatives which were disapproved in$V$ are still disapproved in V', and obviously, at least one alternative in Y is approved; or the threshold moves downward, in which case all alternatives that were approved in V are still approved in V', and obviously not all alternatives in Y are approved. Note that in the case where $V' \cap \mathcal{C} = V$, the threshold can have moved upward, or downward, or remained the same[2].

Let us give a brief summary of the three definitions of extension. Definition 1 assumes that the threshold cannot move; Definition 2 assumes that the threshold can stay the same or move upward (because the set of alternatives grows); and Definition 3 assumes that the threshold can stay the same, move upward, or move downward. Next, we show an example that illustrates these definitions. Let $\mathcal{C} = \{a, b, c, d\}$, $Y = \{y_1, y_2\}$, and $V = \{a, b\}$.

- $V'_1 = \{a, b\}$ and $V'_2 = \{a, b, y_1\}$ are extensions of V under any definition;
- $V' = \{a, y_1\}$ is an extension of V under definitions 2 and 3 but not under definition 1 (the threshold has moved upward, since b was approved in V and is no longer approved in V');
- $V' = \{a, b, c, y_1\}$ is an extension of V under definition 3 but neither under definitions 1 nor 2 (the threshold has moved downward, since c was not approved in V and becomes approved in V' – note that, intuitively, y_2 must be a very unfavored alternative for this to happen);
- $V' = \{a, b, c\}$ is an extension of V under definitions 3 but neither under definitions 1 nor 2, for the same reason as above;
- $V' = \{a\}$ is not an extension of V under any of the definitions: to have b disapproved in V' and approved in V, the threshold has to move upward, which cannot be the case if no alternative of Y is approved;
- $V' = \{a, b, c, y_1, y_2\}$ is not an extension of V under any of the definitions: to have c disapproved in V and approved in V', the threshold has to move downward, which cannot be the case of all alternatives of Y are disapproved;

[2] The rationale behind Definition 3 is that the threshold may depend on the average quality of candidates, and therefore may go down after some bas new candidates have been added. For instance, suppose a voter hates red meat, and has the preference relation `tofu` $\succ$ `fish` $\succ$ `chicken` $\succ$ `beef` $\succ$ `mutton`; if the initial set of candidates is {`tofu, fish, chicken`}, it is perfectly reasonable that he should approve {`tofu, fish` }, while he would approve {`tofu, fish, chicken`} after `beef` and `mutton` have been added in the set of candidates. This is perfectly in agreement with the notion of sincere ballot in approval voting (see, *e.g.*, [7] and references therein).

• $V' = \{a, c, y_1\}$ is not an extension of V under any of the definitions: the threshold cannot simultaneously move upward and downward.

It is straightforward to check that the PcWNA and PWNA problems are in P for approval under definition 1: an alternative $c \in \mathcal{C}$ is a possible (co-)winner in P if and only if it is a (co-)winner for approval in P (this is because for any $V \in P$, the scores of alternatives in $\mathcal{C}$ will not change from V to its extension V'). However, when we adopt definition 2 of extension, the problems become NP-complete.

Theorem 1 *Under Definition 2, PcWNA and PWNA problems are* NP*-complete for the approval rule.*

Proof of Theorem 1: We first prove the hardness of the PcWNA problem by a reduction from X3C. For any X3C instance $I = (\mathcal{S}, \mathcal{V})$, we construct the following PcWNA instance.

Alternatives: $\mathcal{V} \cup \{c\} \cup Y$, where $Y = \{y_1, \ldots, y_{t-q}\}$.

Votes: for any $i \leq t$, we have a vote $V_i = S_i$; and we have an additional vote $V_{t+1} = \{c\}$. That is, $P = (V_1, \ldots, V_t, V_{t+1})$.

Suppose the X3C instance has a solution, denoted by $\{S_{i_1}, \ldots, S_{i_q}\}$. Then, take the following extension P' of P: for any $j \leq q$, let $V'_{i_j} = V_{i_j}$. For any $i \leq t$ such that $i \neq i_j$ for any $j \leq q$, we let V'_i be a singleton containing exactly one of the new alternatives. Let $V'_{p+1} = \{c\}$. For any $v \in \mathcal{V}$, because v appears exactly in one S_{i_j}, v is approved by exactly one voter. So is c. Now, there are exactly $t - q$ votes V_i where i is not equal to one of the i_j's. Therefore, the total approval score of the new alternatives is $t - q$, and it suffices to approve every new alternative exactly once. Therefore c is a co-winner in P', and thus a possible co-winner in P.

Conversely, suppose c is a possible co-winner for P and let P' be an extension of P for which c is a co-winner. We note that c is approved at most once in P'. Therefore, every alternative in $\mathcal{V} \cup Y$ must be approved at most once. Without loss of generality, assume that every vote V'_i in P' is either of the form V_i or of the form $\{y_j\}$ (if not, remove every alternative (except one y_j) from V'_i; c will still be a co-winner in the resulting profile). Since we have $t - q$ new alternatives, each being approved at most once in P', we have at least q votes V'_i in P' such that $V'_i = V_i$. If we had more than q votes V'_i such that $V'_i = V_i$, then more than $3q$ points would be distributed to $3q$ alternatives and one of them would get at least 2, which means that c would not be a co-winner in P'. Therefore we have exactly q votes V'_i such that $V'_i = V_i$, and $3q$ points distributed to $3q$ alternatives; since none of them gets more than one point, they get one point each, which implies that the collection of all S_i such that $V_i = V'_i$ forms an exact cover of C.

For the PWNA problem, we add one more vote $V_{t+2} = \{c\}$ to the profile P. □

Now, let us consider Definition 3. Notice that the profile P' where every voter adds c to her vote (if she was not already voting for c) is an extension of P, and obviously c is a co-winner in P', therefore every alternative of $\mathcal{C}$ is a possible co-winner for P, which means that the problem is trivial.

4 Bucklin

Theorem 2 *The PWNA and PcWNA problems are* NP*-complete for Bucklin, even when there are three new alternatives.*

Proof of Theorem 2: We prove the NP-hardness of the PcWNA problem by a reduction from the special case of 3DM mentioned in Proposition 1. Given any 3DM instance where $|A| = |B| = |X| = q$, q is even, $t = 3q/2$, and no element in $A \cup B \cup X$ appears in more than 6 elements in T, we construct a PcWNA instance as follows. Without loss of generality, assume $q \geq 5$; otherwise the instance 3DM can be solved in linear time.

Alternatives: $A \cup B \cup X \cup Y \cup D \cup \{c\}$, where $Y = \{y_1, y_2, y_3\}$ is the set of new alternatives, and

$D = \{d_1, \ldots, d_{9q^2}\}$ is the set of auxiliary alternatives.

Votes: For any $i \leq 2q+1$, we define a vote V_i. Let $P = (V_1, \ldots, V_{2q+1})$. Instead of defining these votes explicitly, below we give the properties that P satisfies. The votes can be constructed in polynomial time.

(i) For any $i \leq q$, c is ranked in the first position. Suppose $S_i = (a, b, x)$. Then, let a, b, x be ranked in the $(3q+1)$th, $(3q+2)$th, and $(3q+3)$th positions in V_i, respectively.

(ii) For any i such that $q < i \leq 3q/2 = t$, c is ranked in the $(3q+4)$th position. Suppose $S_i = (a, b, x)$. Then, let a, b, x be ranked in the $(3q+1)$th ,$(3q+2)$th, and $(3q+3)$th positions in V_i, respectively.

(iii) For any i such that $3q/2 < i \leq 2q+1$, let c be ranked in the $(3q+4)$th position, and no alternative in $A \cup B \cup X$ is ranked in the $(3q+1)$th, $(3q+2)$th, or $(3q+3)$th position in V_i.

(iv) For any $j \leq 3q$, v_j is ranked within top $3q+3$ positions for exactly $q+1$ times in P.

(v) For any $d \in D$, d is ranked within top $3q+4$ positions at most once.

The existence of a profile P that satisfies (iv) is guaranteed by the assumption that in the 3DM instance, $q \geq 5$, no element is covered more than 6 times, and there are enough positions within top $3q+3$ positions in all votes to fit in all alternatives in $\mathcal{C}$, with each alternative appears $q+1$ times. We note that there are in total $9q^2$ auxiliary alternatives, and the total number of top $3q+4$ positions in all votes is $(3q+4)(2q+1) < 9q^2$. Therefore, (v) can be satisfied. It follows that there exists a profile P that satisfies (i), (ii), (iii), (iv), and (v), and such a profile can be constructed in polynomial time (by first putting the alternatives to their positions defined in (i), (ii), and (iii), then filling out the positions using remaining alternatives to meet conditions (iv) and (v)). The Bucklin score of c is $3q+4$ in P. For any $j \leq q$, the Bucklin score of a_j (resp., b_j, x_j) is at most $3q+3$ in P, and for any $j \leq 9q^2$, the Bucklin score of $d_j \in D$ is at least $3q+4$ in P. Observe that the Bucklin score of any alternative cannot be decreased in any extension of P.

Suppose that the 3DM instance has a solution, denoted by $\{S_j : j \in J\}$, where $J \subseteq \{1, \ldots, t\}$. For any $j \in J$, we let V'_j be the extension of V_j in which y_1, y_2, y_3 are ranked in the $(3q+1)$th, $(3q+2)$th, and $(3q+3)$th positions, respectively. For any $j \in \{1, \ldots, 2q+1\} \setminus J$, we let V'_j be the extension of V_j where $\{y_1, y_2, y_3\}$ are ranked in the bottom positions. Let $P' = (V'_1, \ldots, V'_{2q+1})$. It follows that in P', the Bucklin score of c is $3q+4$, and the Bucklin score of any other alternative is at least $3q+4$. Therefore, c is a co-winner for Bucklin for P', which means that there is a solution to the PcWNA instance.

Conversely, suppose that there is a solution to the PcWNA instance, denoted by $P' = (V'_1, \ldots, V'_{2q+1})$. We recall that in order for c to be a co-winner, the Bucklin score of any alternative in $A \cup B \cup X$ must be at least $3q+4$ (since the Bucklin score of c cannot decrease in P'). We note that there are only three new alternatives, and the $(3q+1)$th, $(3q+2)$th, and $(3q+3)$th positions in V_i are occupied by some alternatives in D. It follows that for every $a \in A$ and every i such that $t < i \leq 2q+1$, it cannot be the case that a is ranked within top $3q+3$ positions in V_i, and a is ranked lower than the $(3q+3)$th position in V'_i. Therefore, for every $a \in A$, there exists $i \leq t$ such that a is ranked within top $3q+3$ positions in V_i, and is ranked lower than the $(3q+3)$th position in V'_i. It follows that in each of such V'_i where a is ranked lower than the $(3q+3)$th position, the new alternatives must be ranked within top $3q+3$ positions. Therefore, each new alternative must be ranked within top $3q+3$ positions in $V_1, \ldots, V_t$ for q times (one for each $a \in A$). Because c is a co-winner, no alternative in Y is ranked within top $3q+3$ positions in P' for more than q times. Therefore, in exactly q votes in P', the alternatives in Y are ranked within top $3q+3$ positions. We let $\{V'_{i_1}, \ldots, V'_{i_q}\}$ denote these votes.

We claim that $\{S_{i_1}, \ldots, S_{i_q}\}$ is a solution to the 3DM instance. If not, then there exists $e \in B \cup X$ that does not appear in any S_{i_j}. However, it follows that e is ranked within top $3q+3$ positions for exactly q times, which means that the Bucklin score of e is at most $3q+3$. Therefore, the Bucklin score of e is lower than the Bucklin score of c. This contradicts the assumption that c is a co-winner for P'. Therefore, the PcWNA problem is NP-hard for Bucklin.

For PWNA, we make the following changes. In conditions (i) and (ii) that P should satisfy, we require that a, b, x are in the $(3q+2)$th, $(3q+3)$th, and $(3q+4)$th positions, respectively. $\square$

5 Copeland$_0$

For any profile P, the Copeland score of an alternative $c \in \mathcal{C}$ in profile P is denoted by $\text{CS}_P(c) = |\{c' \in \mathcal{C} : N_P(c, c') > n/2\}|$ (recall that we focus on Copeland$_0$, which means that the tie in a pairwise election gives 0 point to both participating alternatives). We have the following straightforward observation.

Property 1 *For any profile P' over $\mathcal{C} \cup \{y\}$ that is an extension of profile P, the following inequalities hold:*

$$\forall c \in \mathcal{C},\ CS_P(c) \leq CS_{P'}(c) \leq CS_P(c) + 1 \tag{1}$$

We prove that a useful restriction of X3C remains NP-complete.

Proposition 2 X3C *is* NP*-complete, even if $t = 2q - 2$ and $\Delta(I) \leq 6$.*

Proof of Proposition 2: The proof is similar to the proof for Proposition 1. Let $I = (\mathcal{S}, \mathcal{V})$ be an instance of X3C, where $\mathcal{V} = \{v_1, \ldots, v_{3q}\}$ and $\mathcal{S} = \{S_1, \ldots, S_t\}$. We next show how to build an instance $I' = (\mathcal{S}', \mathcal{V}')$ of X3C in polynomial time, with $|\mathcal{V}'| = 3q'$ and $|\mathcal{S}'| \leq 6$ such that $t' = 2q' - 2$ and $\Delta(I') \leq 6$.

• If $t < 2q - 2$, then we add $2q - 2 - t$ dummy 3-sets to $\mathcal{S}$ by duplicating $2q - 2 - t$ sets of $\mathcal{S}$ once each. It follows from $t \geq q$ that $2q - 2 - t \leq q - 2 < t$.

• If $t > 2q - 2$, then we add $3(t - 2q + 2)$ new elements $v'_1, \ldots, v'_{3(t-2q+2)}$ and $t - 2q + 2$ 3-sets $\{v'_1, v'_2, v'_3\}, \ldots, \{v'_{3(t-2q+2)-2}, v'_{3(t-2q+2)-1}, v'_{3(t-2q+2)}\}$.

The size of the input of the new instance is polynomial in the size of the input of the old instance. Moreover, I is a yes-instance if and only if I' is also a yes-instance. Finally, in the new instance I', we have: $|\mathcal{V}'| = |\mathcal{V}| = 3q$ and $t' = |\mathcal{S}'| = t + (2q - 2 - t) = 2q - 2 = 2q' - 2$ in the first case, while $3q' = |X'| = 3q + 3(t - 2q + 2) = 3(t - q + 2)$ and $t' = |\mathcal{S}'| = t + (t - 2q + 2) = 2(t - q + 1) = 2(q' - 1)$ in the second case. Moreover, $d_{I'}(v) \leq 2d_I(v) \leq 6$ if $v \in \mathcal{V}$, and $d_{I'}(v) = 1$ if $v \in \mathcal{V}' \setminus \mathcal{V}$. $\square$

Theorem 3 *The PcWNA problem is* NP*-complete for Copeland$_0$, even when there is one new alternative.*

Proof of Theorem 3: The proof is by a reduction from X3C. Let $I = (\mathcal{S}, \mathcal{V})$, where $t = 2q - 2$ and $\Delta(I) \leq 6$ be an instance of X3C as described in Proposition 2. As previously, assume $q \geq 8$; hence $\Delta(I) \leq q - 2$. For any X3C instance, we construct the following PcWNA instance for Copeland$_0$.
Alternatives: $\mathcal{V} \cup D \cup Y \cup \{c\}$, where $D = \{d_1, \ldots, d_t\}$ and $Y = \{y\}$ is the set of the new alternative.
Votes: For any $i \leq t$, we define the following $2t$ votes.

$$V_i = [d_i \succ (D \setminus \{d_i\}) \succ (\mathcal{V} \setminus S_i) \succ c \succ S_i]$$

$$V'_i = [\text{rev}(S_i) \succ \text{rev}(\mathcal{V} \setminus S_i) \succ \text{rev}(D \setminus \{d_i\}) \succ c \succ d_i]$$

Here the elements in a set are ranked according to the order of their subscripts, i.e., if $S_i = \{v_2, v_5, v_7\}$, then the elements are ranked as $v_2 \succ v_5 \succ v_7$. For any set X such that $X \subset \mathcal{V}$ or $X \subset D$, let $\text{rev}(X)$ denote the linear order where the elements in X are ranked according to the reversed order of their subscripts. For example, $\text{rev}(\{v_2, v_5, v_7\}) = v_7 \succ v_5 \succ v_2$.

We also define the following $t = 2q - 2$ votes.

$$W_1 = \ldots = W_{q-1} = [\mathcal{V} \succ D \succ c]$$

$$W'_1 = \ldots = W'_{q-1} = [\text{rev}(D) \succ \text{rev}(\mathcal{V}) \succ c]$$

Let $P = (V_1, V'_1, \ldots, V_t, V'_t, W_1, W'_1, \ldots, W_{q-1}, W'_{q-1})$.

We note that there are $3t$ votes in the instance. We recall that by assumption, $3t/2 = 3q-3$. We make the following observations on the function N_P.

• *For any $d \in D$, d beats c*: this holds because $N_P(c,d) = 1$.

• *For any $v \in \mathcal{V}$, v beats c*: this holds because $N_P(c,v) = d_I(v) \le q-2 < 3q-3$.

• *For any $d \in D$ and $v \in \mathcal{V}$, d and v are tied*: this holds because $N_P(v,d) = t+q-1 = 3q-3$.

• *For any $v, v' \in \mathcal{V}$ ($v' \neq v$), v and v' are tied*: this holds because $N_P(v,v') = t+q-1 = 3q-3$, because for any $i \le q$, $v \succ v'$ either in V_i or in V'_i.

• *For any $d, d' \in D$ ($d' \neq d$), d and d' are tied*: this holds because $N_P(d,d') = 3q-3$.

From these observations we have the following calculation on the Copeland scores:

• $\text{CS}_P(c) = 0$.

• For any $v \in \mathcal{V}$, $\text{CS}_P(v) = 1$.

• For any $d \in D$, $\text{CS}_P(d) = 1$.

Now, assume that $I = (\mathcal{S}, \mathcal{V})$ is a yes-instance of X3C; hence, there exists $J \subset \{1, \ldots, t\}$ with $|J| = q$ and $\bigcup_{j \in J} S_j = \mathcal{V}$. Next, we show how to make c a co-winner by introducing one new alternative y.

• For any $j \in J$, we let $\widetilde{V}_j = [d_j \succ D \setminus \{d_j\} \succ \mathcal{V} \setminus S_j \succ c \succ y \succ S_j]$ be the completion of V_j.

• For any $i \le t$, we let $\widetilde{V}'_i = [\text{rev}(S_i) \succ \text{rev}(\mathcal{V} \setminus S_i) \succ \text{rev}(D \setminus \{d_i\}) \succ c \succ y \succ d_i]$ be the completion of V'_i.

• For any vote not mentioned above, we put y in the top position.

• Finally, let P' denote the profile obtained in the above way.

It follows that y loses to c in their pairwise election, and for any other alternative $c' \in \mathcal{C}$ ($c' \neq y$ and $c' \neq c$), c' and y are tied in their pairwise election. Therefore, the Copeland score is 1 for c, any alternative in $\mathcal{V}$, and any alternative in D; the Copeland score of y is 0. It follows that c is a co-winner.

Next, we show how to convert a solution to the PcWNA instance to a solution to the X3C instance. Let $P' = (\widetilde{V}_1, \ldots, \widetilde{V}_t, \widetilde{V}'_1, \ldots, \widetilde{V}'_t, \widetilde{W}_1, \widetilde{W}'_1, \ldots, \widetilde{W}_{q-1}, \widetilde{W}'_{q-1})$ be a profile with the new alternative, such that c becomes a co-winner according to the Copeland$_0$ rule. We denote $P'_1 = (\widetilde{V}_1, \ldots, \widetilde{V}_t)$, $P'_2 = (\widetilde{V}'_1, \ldots, \widetilde{V}'_t)$ and $P'_3 = (\widetilde{W}_1, \widetilde{W}'_1, \ldots, \widetilde{W}_{q-1}, \widetilde{W}'_{q-1})$. It follows from the above observations on Copeland scores of alternatives in profile P and inequalities (1) of Property 1, that $\text{CS}_{P'}(c) = 1$, $\forall c' \in D \cup \mathcal{V}$, $\text{CS}_{P'}(c) = 1$ and $\text{CS}_{P'}(y) \le 1$.

We now claim the following.

(a) $\forall v \in \mathcal{V}$, $N_{P'}(v,y) \le 3q-3$, $N_{P'}(y,c) = 3q-2$ and $\forall d \in D$, $N_{P'}(d,y) = 3q-3$. $N_{P'_2}(c,y) = t = 2q-2$. Moreover, for any $i \le q$, $c \succ y \succ d_i$ in $\widetilde{V}'_i$.

(b) $\forall v \in \mathcal{V}$, $N_{P'_2 \cup P'_3}(v,y) \ge N_{P'_2 \cup P'_3}(c,y)$.

For (a). Since c is a co-winner for P', c must beat y in their pairwise election. Meanwhile, any $c' \in \mathcal{V} \cup D$ cannot beat y in their pairwise elections. Therefore, we must have that $N_{P'}(c,y) \ge 3q-2$, and for any $c' \in \mathcal{V} \cup D$, $N_{P'}(c',y) \le 3q-3$. For any $d_i \in D$, in profile P', we have that $d_i \succ c$ except in $\widetilde{V}'_i$, which means that $N_{P'}(d_i,y) \ge N_{P'}(c,y) - 1$ by transitivity in each vote. Hence, $3q-3 \ge N_{P'}(d_i,y) \ge N_{P'}(c,y) - 1 \ge 3q-3$, which means that $N_{P'}(d_i,y) = 3q-3$ and $N_{P'}(c,y) = 3q-2$. From these equalities, we deduce that $\forall d \in D$, $N_{P'}(d,y) = N_{P'}(c,y) - 1$ and then, for any $i \le t$, we have that $c \succ y \succ d_i$ in $\widetilde{V}'_i$.

For (b). Since in P', $v \succ c$ except for some votes in P'_1, we have that for all $v \in \mathcal{V}$, $N_{P'_2 \cup P'_3}(v,y) \ge N_{P'_2 \cup P'_3}(c,y)$.

Let $J = \{j \le t : c \succ y \text{ in } \widetilde{V}_j\}$. We will prove that $|J| = q$ and $\cup_{j \in J} S_j = \mathcal{V}$. First, note that $|J| \le q$ because $|J| = N_{P'_1}(c,y) \le N_{P'}(c,y) - N_{P'_2}(c,y) = q$ from item (a).

Now, for any $v \in \mathcal{V}$ let $J_v = \{j \le t : y \succ v \text{ in } \widetilde{V}_j\}$. We claim: $\forall v \in \mathcal{V}$, $J \cap J_v \neq \emptyset$. Otherwise, there exists $v^* \in \mathcal{V}$ with $J \cap J_{v^*} = \emptyset$. This means that $c \succ y$ implies $v^* \succ y$ in votes in

P'_1. Hence, $N_{P'_1}(v^*, y) \geq N_{P'_1}(c, y)$. By adding this inequality with the inequality in item (b) (let $v = v^*$), we obtain that $N_{P'}(v^*, y) \geq N_{P'}(c, y)$. Now, combining the inequalities in item (a), we have that $3q - 3 \geq N_{P'}(v^*, y) \geq N_{P'}(c, y) = 3q - 2$, which is a contradiction. Therefore, for all $v \in \mathcal{V}$, $J \cap J_v \neq \emptyset$. Finally, since $|\mathcal{V}| = 3q$, $|S_i| = 3$ and $|J| \leq q$, we deduce that $|J| = q$ and $J = \{j \leq t : c \succ y \succ S_j \text{ in } \widetilde{V}_j\}$. Also, because for all $v \in \mathcal{V}$, $J \cap J_v \neq \emptyset$, we have $\bigcup_{j \in J} S_j = \mathcal{V}$. In conclusion, $I = (\mathcal{S}, \mathcal{V})$ is a yes-instance of X3C. This completes the NP-hardness proof for the PcWNA problem for Copeland$_0$. □

6 Simpson

To prove the NP-hardness of the PcWNA problem for Simpson, we first make the following observation, whose proof is straightforward.

Property 2 *Let P be a profile over $\mathcal{C}$, P' be a profile over $\mathcal{C} \cup \{y\}$, P' is an extension P. The following (in)equalities hold:*
(i) $\forall c \in \mathcal{C}$, $Sim_{P'}(c) = \min\{Sim_P(c), N_{P'}(c, y)\}$.
(ii) $\forall c \in \mathcal{C}$, $Sim_{P'}(c) \leq Sim_P(c)$.

Theorem 4 *PcWNA and PWNA problems are* NP*-complete for Simpson, even when there is one new alternative.*

Proof of Theorem 4: We first prove the NP-hardness for the PcWNA problem by a reduction from X3C. Let $I = (\mathcal{S}, \mathcal{V})$ with $t = 2q - 2$ and $\Delta(I) \leq 6$ be an instance of X3C as described in Proposition 2. Without loss of generality, assume $q \geq 8$; in particular, we deduce $\Delta(I) \leq q - 2$. We define a PcWNA instance for Simpson as follows:
Alternatives: $\mathcal{V} \cup \{c, d\} \cup \{y\}$, where y is the new alternative.
Votes: For any $i \leq t$, we define the following vote. $V_i = [(\mathcal{V} \setminus S_i) \succ d \succ c \succ S_i]$. For any $j \leq q - 1$, we define the following vote. $W_1 = \cdots = W_{q-1} = [c \succ \text{rev}(\mathcal{V}) \succ d]$. We also let $W_q = [\text{rev}(\mathcal{V}) \succ d \succ c]$. Let $P_1 = (V_1, \ldots, V_t)$, $P_2 = (W_1, \ldots, W_q)$, and $P = P_1 \cup P_2$.

We make the following observation on the Simpson scores of the alternatives before y is added.

- $Sim_P(c) = q - 1$. Indeed, $N_P(c, d) = q - 1$ and $\forall v \in \mathcal{V}$, $N_P(c, v) = q - 1 + d_I(v) \geq q$.
- $Sim_P(d) \leq 6 \leq q - 2$. This is because for any $v \in \mathcal{V}$, v is covered by the 3-sets for no more than $q - 2$ times (the assumption of the input X3C instance), which means that in P_1, $d \succ v$ for at most $q - 2$ times, i.e., $N_P(d, v) = d_I(v) \leq 6 \leq q - 2$.
- For any $v \in \mathcal{V}$, $Sim_P(v) \geq q$. Actually, $N_P(v, d) = N_P(v, c) = t - d_I(v) + q \geq 3q - 2 - (q - 2) \geq q$. Now, assume $v = v_i$. If $i < j$, then $N_P(v, v_j) = N_{P_1}(v, v_j) \geq t - d_I(v) \geq 2q - 2 - (q - 2) = q$ and if $j > i$, $N_P(v, v_j) = N_{P_2}(v, v_j) = q$.

Now, assume that $I = (\mathcal{S}, \mathcal{V})$ is a yes-instance of X3C; hence, there is a $J \subset \{1, \ldots, t\}$ with $|J| = q$ and $\bigcup_{j \in J} S_j = \mathcal{V}$. We show how to make c a co-winner by introducing one new alternative y.

- For any $j \in J$, we let $V'_j = [(\mathcal{V} \setminus S_j) \succ d \succ c \succ y \succ S_j]$.
- For any $j \in \{1, \ldots, t\} \setminus J$, we let $V'_j = [y \succ (\mathcal{V} \setminus S_j) \succ d \succ c \succ S_j]$.
- For any $j \leq q - 1$, we let $W'_j = [c \succ y \succ \text{rev}(\mathcal{V}) \succ d]$.
- Let $W'_q = [y \succ \text{rev}(\mathcal{V}) \succ d \succ c]$.
- Finally, let $P' = (V'_1, \ldots, V'_t, W'_1, \ldots, W'_q)$.

In P', the Simpson score of y is $q - 1$ (via c), because $t = 2q - 2$, which means that $t - q + 1 = q - 1$; the Simpson score of c is $q - 1$ (via d); the Simpson score of d is no more than $q - 1$ (via any of $v \in \mathcal{V}$); and the Simpson score of any $v \in \mathcal{V}$ is $q - 1$ (via y). Therefore, c is a co-winner for the Simpson rule.

Next, we show how to convert a solution P' to the above PcWNA instance for the Simpson rule to a solution to the X3C instance. Let $P' = (V'_1, \ldots, V'_t, W'_1, \ldots, W'_q)$ with $P'_1 = (V'_1, \ldots, V'_t)$ and

$P_2' = (W_1', \ldots, W_q')$ be a profile such that c becomes a co-winner according to the Simpson rule when alternative y is introduced.

We make the following observations.

(a) $\forall v \in \mathcal{V}$, $N_{P'}(v, y) \leq q - 1$,

(b) $N_{P'}(y, c) \leq q - 1$ and $N_{P'}(y, d) \geq q$,

(c) $y \succ c$ in W_q'.

For item (a): Since c is a winner, we have that for any $v \in \mathcal{V}$, $Sim_{P'}(v) \leq Sim_{P'}(c)$. Thus, using Property 2, $Sim_P(c) = q - 1$ and $Sim_P(v) \geq q$. We have the following calculation.

$$\min\{N_{P'}(v, y), q\} = Sim_{P'}(v) \leq Sim_{P'}(c) \leq Sim_P(c) = q - 1$$

For item (b): First from (a), we deduce that for any $v \in \mathcal{V}$, $N_{P'}(y, v) \geq t + q - N_{P'}(v, y) > q$. Thus, we obtain:

$$Sim_{P'}(y) = \min\{N_{P'}(y, c), N_{P'}(y, d)\} \leq Sim_{P'}(c) \leq Sim_P(c) = q - 1 \tag{2}$$

Now, assume $N_{P'}(y, d) \leq q - 1$. Then, $N_{P_2'}(d, y) = q - N_{P_2'}(y, d) \geq q - N_{P'}(y, d) \geq 1$. Hence, there exists $i \leq q$ such that in W_i', we have that for any $v \in \mathcal{V}$, $v \succ d \succ y$. Moreover, $N_{P_1'}(d, y) = t - N_{P_1'}(y, d) \geq 2q - 2 - (q - 1) = q - 1$. Let $J_0 \subseteq \{1, \ldots, t\}$ (with $|J_0| = q - 1$) be the subscripts of arbitrary $q - 1$ votes in P_1', where $d \succ y$. Because $|\mathcal{V}| = 3q$ and $|S_j| = 3$, there exists $v^* \in \mathcal{V} \setminus \bigcup_{j \in J_0} S_j$. We deduce that for all $j \in J_0$, $v^* \succ y$ in V_j'. In conclusion, $N_{P'}(v^*, y) \geq |J_0| + 1 = q$, which contradicts item (a). Using inequality (2), item (b) follows.

For item (c): Otherwise, by the definition of W_q, we deduce:

$$\forall v \in \mathcal{V},\ N_{P_2'}(v, y) \geq 1 \tag{3}$$

On the other hand, using $N_{P_1'}(y, c) \leq N_{P'}(y, c)$ and item (b), we have $N_{P_1'}(c, y) = t - N_{P_1'}(y, c) \geq t - N_{P'}(y, c) \geq t - (q - 1) = q - 1$. Let $J_0 \subseteq \{1, \ldots, t\}$ (with $|J_0| = q - 1$) be the subscripts of arbitrary $q - 1$ votes in P_1', where $c \succ y$. We have $\mathcal{V} \setminus \bigcup_{j \in J_0} S_j \neq \emptyset$ since $|\mathcal{V}| = 3q$ and $|S_i| = 3$. Hence, there exists $v^* \in \mathcal{V} \setminus \bigcup_{j \in J_0} S_j$ such that:

$$N_{P_1'}(v^*, y) \geq |J_0| = q - 1 \tag{4}$$

Summing up inequalities (3) (let $v = v^*$) and (4), we get obtain a contradiction with item (a).

From items (b) and (c), we get $N_{P_1'}(y, c) = N_{P'}(y, c) - N_{P_2'}(y, c) \leq q - 1 - 1 = q - 2$. Thus, $N_{P_1'}(c, y) = t - N_{P_1'}(y, c) \geq t - (q - 2) = q$. Let J denote the subscripts of arbitrary q votes in P_1' where $c \succ y$. We claim $\bigcup_{j \in J} S_j = \mathcal{V}$. Otherwise, there exists $v^* \in \mathcal{V} \setminus \bigcup_{j \in J} S_j$. It follows that for any $j \in J$, $v^* \in (\mathcal{V} \setminus \bigcup_{j \in J} S_j) \subseteq \mathcal{V} \setminus S_j$, which means that $v^* \succ c \succ y$ in V_j. Hence, $N_{P'}(v^*, y) \geq N_{P_1'}(v^*, y) \geq |J| = q$, which contradicts item (a). In conclusion, $I = (\mathcal{S}, \mathcal{V})$ is a yes-instance of X3C. Therefore, PcWNA is NP-complete for Simpson.

For the PWNA problem, we make the following change. Let $W_q = [\text{rev}(\mathcal{V}) \succ c \succ d]$. Then, before the new alternative is introduced, the Simpson score of c is q. Then, similarly we can prove the NP-hardness of the PWNA problem. □

7 Plurality with runoff

In this section, we focus on possible co-winners, which means that ties are never broken, neither in the first round nor in the second round. If a tie occurs in the first round, then all possible compatible second rounds are considered: for instance, if the plurality scores, ranked in decreasing order, are $x_1 \mapsto 8, x_2 \mapsto 6, x_3 \mapsto 6, x_4 \mapsto 5 \ldots$, then the set of co-winners contains the majority winner between x_1 and x_2 and the majority winner between x_1 and x_3.

Proposition 3 *Determining whether $c \in \mathcal{C}$ is a possible (co-)winner for plurality with runoff is in* P.

The proof does not present any particular difficulty, and due to the lack of space, we only give a very brief sketch for the PcWNA problem. It proceeds in two steps as follows. Let $\succeq_M^P$ be the weak majority relation induced by a profile P. Let P be a profile over $\mathcal{C}$. c is a possible co-winner in P if and only if one of the following two conditions hold:

1. There exists a completion P' of P such that c and some $d \in \mathcal{C} \setminus \{c\}$ are possible second round competitors, and $c \succeq_M^{P'} d$.

2. There exists a completion P' of P such that c and some $y \in Y$ are possible second round competitors, and $c \succeq_M^{P'} y$.

For each of these two conditions we can find equivalent, polynomial-time computable characterizations.

For the PWNA problem, the algorithm is similar: we need to make sure that the pairs of alternatives that enter the second round must be (c, d), where $c \succ_M^P d$.

8 Conclusion

In this paper we have gone much beyond existing results on the complexity of the possible (co-)winner problem with new alternatives. While [5, 6] focused on scoring rules, we have identified three new rules for which the PcWNA problem is NP-complete (Bucklin, Copeland, and Simpson). We also showed that the PcWNA problem has a polynomial time algorithm for plurality with runoff, and as far as approval voting is concerned, we have given three definitions of the extension of a profile to new alternatives and shown that depending on the chosen definition, the problem can be trivial or NP-complete. Our NP-completeness proofs and algorithms for the PcWNA problems can also be extended to the PWNA problems for approval, Bucklin, Simpson, and plurality with runoff. The results are summarized in the following table.

<table>
<tr><th>Voting rule</th><th>PcWNA</th><th>PWNA</th></tr>
<tr><td>Borda</td><td colspan="2">P [6]</td></tr>
<tr><td>2-approval</td><td colspan="2">P [6]</td></tr>
<tr><td>l-approval ($l \geq 3$)</td><td colspan="2">NP-complete [2] [6]</td></tr>
<tr><td>Approval</td><td colspan="2">P (Definition 1)
NP-complete (Definition 2)
Trivial (Definition 3)</td></tr>
<tr><td>Bucklin</td><td colspan="2">NP-complete [2]</td></tr>
<tr><td>Copeland$_0$</td><td>NP-complete [3]</td><td>?</td></tr>
<tr><td>Simpson</td><td colspan="2">NP-complete [3]</td></tr>
<tr><td>Plurality with runoff</td><td colspan="2">P</td></tr>
</table>

Table 1: Complexity of PcWNA and PWNA problems for some common voting rules.

An obvious and interesting direction for future research is studying the computational complexity of the PcWNA (PWNA) problems for more common voting rules, including Copeland$_\alpha$ (for some $\alpha \neq 0$), ranked pairs, and voting trees. Even for Copeland$_0$, the complexity of the PWNA problem still remains open.

[2] Even with 3 new alternatives.
[3] Even with 1 new alternative.

Acknowledgements

Lirong Xia is supported by a James B. Duke Fellowship and NSF under award number IIS-0812113.

References

[1] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Social Choice and Welfare*, 16(8-9):27–40, 1992.

[2] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, 2010. To appear.

[3] N. Betzler and B. Dorn. Towards a dichotomy of finding possible winners in elections based on scoring rules. In *Proc. MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 2009.

[4] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. IJCAI-09*, pages 53–58, 2009.

[5] Y. Chevaleyre, J. Lang, N. Maudet, and J. Monnot. Possible winners when new candidates are added: the case of scoring rules. In *Proceedings of AAAI 2010*, 2010. To appear.

[6] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. Submitted to MSS, 2010.

[7] U. Endriss. Vote manipulation in the presence of multiple sincere ballots. In *Proceedings of TARK-07*, pages 125–134, 2007.

[8] M. Garey and D. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. Freeman, 1979.

[9] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[10] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *J. Artif. Intell. Res. (JAIR)*, 33:149–178, 2008.

[11] M.S. Pini, F. Rossi, K. Brent Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *Proceedings of IJCAI'07*, pages 1464–1469, 2007.

[12] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of AAAI-08*, pages 196–201, 2008.

Lirong Xia
Department of Computer Science
Duke University
Durham, NC 27708, USA
Email: `lxia@cs.duke.edu`

Jérôme Lang
LAMSADE
Universite Paris-Dauphine
75775 Paris Cedex, France
Email: `lang@lamsade.dauphine.fr`

Jérôme Monnot
LAMSADE
Universite Paris-Dauphine
75775 Paris Cedex, France
Email: `jerome.monnot@lamsade.dauphine.fr`

Bypassing Combinatorial Protections: Polynomial-Time Algorithms for Single-Peaked Electorates*

Felix Brandt, Markus Brill, Edith Hemaspaandra, and Lane A. Hemaspaandra

Abstract

For many election systems, bribery (and related) attacks have been shown NP-hard using constructions on combinatorially rich structures such as partitions and covers. It is important to learn how robust these hardness protection results are, in order to find whether they can be relied on in practice. This paper shows that for voters who follow the most central political-science model of electorates—single-peaked preferences—those protections vanish. By using single-peaked preferences to simplify combinatorial covering challenges, we show that NP-hard bribery problems—including those for Kemeny and Llull elections—fall to polynomial time. By using single-peaked preferences to simplify combinatorial partition challenges, we show that NP-hard partition-of-voters problems fall to polynomial time. We furthermore show that for single-peaked electorates, the winner problems for Dodgson and Kemeny elections, though Θ_2^p-complete in the general case, fall to polynomial time. And we completely classify the complexity of weighted coalition manipulation for scoring protocols in single-peaked electorates.

1 Introduction

Elections are perhaps the most important framework for preference aggregation. An election (rule) is a mapping that takes as input the preferences of the voters with respect to the set of candidates (alternatives) and returns a set of "winners," which is some subset of the candidate set. Elections are central in preference aggregation among humans—in everything from political elections to selecting good singers on popular television shows. Elections are rapidly increasing in importance in electronic settings such as multiagent systems, and have been used or proposed for such varied tasks as recommender systems and collaborative filtering [23], web spam reduction and improved web-search engines [12], and planning [13]. In electronic settings, elections may have huge numbers of voters and alternatives.

One natural worry with elections is that agents may try to slant the outcome, for example, by bribing voters. Motivated by work from economics and political science showing that reasonable election systems always allow manipulations of certain types, starting in 1989, Bartholdi, Tovey, and Trick [3, 4] made the thrilling suggestion that elections be protected via complexity theory—namely, by making the attacker's task NP-hard. This line has been active ever since, and has resulted in NP-hardness protections being proven for many election systems, against such attacks as bribery (the attacker has a budget with which to buy and alter voters' votes [16]), manipulation (a coalition of voters wishes to set its votes to make a given candidate win [3]), and control (an agent seeks to make a given candidate win by adding/deleting/partitioning voters or candidates [4]). The book chapter [18] surveys such NP-hardness results, which apply to many important election systems such as plurality, single transferable vote, and approval voting.

In the past few years, a flurry of papers have come out asking whether the NP-hardness protections are satisfying. In particular, the papers explore the possibility that heuristic algorithms may do well frequently or that approximation algorithms may exist. The present paper questions the NP-hardness results from a completely different direction. In political science, perhaps the most

*This paper appeared in the proceedings of AAAI-2010.

“canonical” model of electorates is the unidimensional single-peaked model, in which the electorate has preferences over some one-dimensional spectrum (e.g., “very liberal through very conservative”) along which the candidates are also located, and in which each voter’s preferences (loosely put) have a peak, with affinity declining as one moves away from the peak. A brilliant paper by Walsh [26] recently asked whether NP-hardness protections against manipulation fall apart if electorates are single-peaked. For the case Walsh looked at, the answer he proved is “no”; he looked at a particular NP-hardness manipulation protection and proved it holds even for single-peaked societies. Faliszewski et al. [17], inspired by Walsh’s work, looked at a range of election systems and came to the sharply differing conclusion that for many crucial cases, NP-hardness protections against manipulation and control vanish for single-peaked electorates.

Those two papers [17, 26] are the only two papers we know of that study the implications of single-peakedness on the complexity of manipulation and control. The present paper seeks to take this young line of research in new directions, and to improve one existing direction, via the following contributions:

(1) We show that checking who the winner is in Dodgson, Young, and Kemeny elections, which is Θ_2^p-complete in the general case, is in polynomial time for single-peaked electorates.

(2) We for the first time study the effect of single-peaked electorates on the complexity of *bribery*. We show that many NP-hardness protections against bribery in the general case vanish for single-peaked electorates. To show this, we give polynomial-time bribery algorithms for single-peaked electorates in many settings. Our polynomial-time algorithms apply to approval voting and to the rich range of “weak-Condorcet consistent” election systems and even to systems that are merely known to be weak-Condorcet consistent when the electorate is single-peaked, including weakBlack, weakDodgson, Fishburn, Kemeny, Llull, Maximin, Schwartz, Young, and two variants of Nanson elections.

The practical lesson is that we should be very skeptical about NP-completeness results if our electorate may have limitations (such as single-peakedness) on the ensembles of votes it produces.

(3) We for the first time study the effects of single-peaked electorates on the complexity of *control by partition of voters*, in which the voters are partitioned into two groups that vote on the candidates in “primary” elections, and only the winners of the primaries compete in the final election. This is one of the seven types of control introduced in the seminal control paper of Bartholdi et al. [4], but control by partition of voters has not been previously addressed for the single-peaked case. We show that some known NP-hardness protections against control-by-partition vanish for single-peaked electorates

The shared technical theme here and in the bribery case is that single-peakedness can be used to tame the combinatorial explosion (of partitions and covers) that in the general case protected elections from attack, and in particular single-peakedness yields polynomial-time attack algorithms.

(4) Our final contribution is a strong extension of an important result from Faliszewski et al. [17]. For the broad class of election systems known as scoring protocols, Faliszewski et al. gave a complete characterization of the computational complexity of the (weighted, coalition) manipulation problem in the case of single-peaked elections *with three candidates*. Such characterizations are important as they tell both which systems are manipulable and what it is about the systems that makes them manipulable. We extend this by providing, for single-peaked electorates, a complete characterization of easy manipulability of scoring protocols.

Proofs omitted due to space constraints can be found in the full version of this paper [6].

2 Preliminaries

Election Systems, Preferences, and weakCondorcet-Consistency An election system is a mapping from a finite set of candidates C and a finite list V of voter preferences over those candidates to a collection $W \subseteq C$ called the winner set. For all but one of the election systems we cover, each voter's preference is a linear order (by which we always mean a strict linear order: an irreflexive, antisymmetric, complete, transitive relation) over the candidates. For the election system called approval voting, each voter votes by a bit-vector, approving or disapproving of each candidate separately. Voter's preferences are input as a list of ballots (i.e., votes), so if multiple voters have the same preference, the ballot of each will appear separately in V.

We now very briefly describe the election systems considered in this paper. In *approval voting*, preferences are approval vectors, and each candidate who gets the highest number of approvals among the candidates belongs to the winner set. In all the other systems we use, voters will vote by linear orders. A candidate is said to be a *Condorcet winner* (respectively, *weak Condorcet winner*), if that candidate is preferred to each other candidate by a strict majority (respectively, by at least half) of the voters. In *Condorcet* voting the winners are precisely the set of Condorcet winners. In the election system *weakCondorcet*, the winners are precisely the set of weak Condorcet winners. It has been known for two hundred years that some election instances have neither Condorcet winners nor weak Condorcet winners [7]. And of course, no election instance can have more than one Condorcet winner, whereas there might be several weak Condorcet winners.

For a rational number $\alpha \in [0, 1]$, Copeland$^{\alpha}$ is the election system where for each pair of distinct candidates we see who is preferred between the two by a strict majority of the voters. That one gets one "Copeland point" from the pairwise contest and the other gets zero "Copeland points." If they tie in their pairwise contest (which can happen only when the number of voters is even), each gets α points. Copeland^1 is known as *Llull* elections, a system defined by the mystic Ramon Llull in the thirteenth century, and is known to be remarkably resistant, computationally, to bribery and control attacks [19].

An important class of elections is the class of *scoring protocols*. Each scoring protocol has a fixed number m of candidates and is defined by a *scoring vector* $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$, $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$. Voters' votes are linear orders, and each voter contributes α_1 points to his or her most preferred candidate, α_2 points to his or her next most preferred candidate, and so on. Each candidate whose total number of points is at least as great as the totals of each other candidate is a winner. For example, m-candidate *plurality* voting is the scoring protocol defined by the scoring vector $\alpha = (1, 0, \ldots, 0)$. And m-candidate *Borda* voting is the scoring protocol defined by the scoring vector $\alpha = (m-1, m-2, \ldots, 0)$.

In *Black* elections (respectively, *weakBlack* elections), if there is a Condorcet winner (respectively, if there are weakCondorcet winners), then that defines the winners, and otherwise Borda's method is used to select the winners. Black elections were introduced by Black [5] and weakBlack elections (somewhat confusingly called Black elections there) were introduced by Fishburn [20]. In *Dodgson* elections (respectively, *weakDodgson* elections), whichever candidates can by the fewest repeated transpositions of adjacent candidates in voters' orders become Condorcet winners (respectively, weakCondorcet winners) are the winners. Dodgson elections were introduced in the 1800s by Dodgson and weakDodgson elections (somewhat confusingly called Dodgson elections there) were introduced by Fishburn [20]. In *Young* elections (respectively, *strongYoung* elections), whichever candidates can by the deletion of the fewest voters become weakCondorcet (respectively, Condorcet) winners are the winners. Young elections were introduced by Young and strongYoung elections (somewhat confusingly called Young elections there) were introduced by Rothe et al. [25].

Nanson elections are runoff methods based on Borda's scoring protocol. In Nanson's original definition, a series of Borda elections is held and all candidates who at any stage have not more than the average Borda score are excluded unless all candidates have identical Borda scores, in which case these candidates are declared the winners of the election. There exist two variants of Nanson due to Fishburn and Schwartz, which exclude candidates with the *lowest* Borda score and candidates whose Borda score is *less than* the average score, respectively. *Maximin* (a.k.a. *Simpson*) elections choose those candidates that fare best in their worst pairwise comparison against any other candidate. The remaining three election systems are based on the pairwise majority relation. In *Schwartz* elections (sometimes also called the *top cycle*), the winners are defined as the maximal elements of the asymmetric part of the transitive closure of the majority relation. The winners in *Fishburn* elections are the maximal elements of the Fishburn relation F, which is defined by letting $a \ F \ b$ if every candidate that beats a in a pairwise comparison also beats b and there exists a candidate that beats b but not a. Finally, *Kemeny* elections are based on the smallest number of reversals in the voters' pairwise preferences such that the majority relation becomes transitive and complete. The Kemeny winners are the maximal elements of such minimally modified majority relations.

An important notion in this paper is that of being weakCondorcet-consistent. An election system is said to be *weakCondorcet-consistent* (which we earlier wrote, equivalently, as *weak-Condorcet consistent*), if on every input that has at least one weak Condorcet winner, the winners of the election system are exactly the set of weak Condorcet winners. Some of our bribery results will hold for all election systems that are weakCondorcet-consistent, and even for all election systems that when restricted to single-peaked electorates are weakCondorcet-consistent on those.

Fishburn [20] has noted that the election systems weakBlack, weakDodgson, Fishburn, Maximin, and Young are weakCondorcet-consistent. We add to that the observation that Llull elections are easily seen from their definition to be weakCondorcet-consistent. We also make the (new) observation that the election systems Kemeny, Schwartz, and the two variants of Nanson are weakCondorcet-consistent when restricted to single-peaked electorates. (By Fishburn [20] and Niou, those systems are known not to be weakCondorcet-consistent in the general case.) We also observe that Black, Dodgson, strongYoung, the original version of Nanson, and for each $\alpha \in [0, 1)$, Copeland$^\alpha$ elections are not weakCondorcet-consistent even when restricted to single-peaked electorates.

Single-Peaked Preferences This paper's theme is that combinatorial protections crumble for the case of single-peaked electorates. We now briefly define what single-peaked preferences are and what their motivation is. The single-peaked preference model was introduced over half a century ago by Black [5] and has been influential ever since. The model captures the case where the electorate is polarized by a single issue or dimension, and each voter's utility along that dimension has either one peak or just rises or just falls. Candidates have positions (locations) along that dimension. And a voter's preferences (in the linear order model) simply order the candidates by utility (except with no ties allowed). Since the utility curves are very flexible, what this amounts to is that there is an overall societal ordering L of the candidates, and each voter can be placed in some location such that for all the candidates to his or her right the preferences drop off and the same to the left, although within that framework, the right and the left candidates can be interspersed with each other. A picture will make this clearer. Figure 1 shows an electorate with four voters and five candidates, in which society's polarization is on a (liberal-to-conservative) axis. From the picture, we can see that v_1 has preferences $c_5 > c_4 > c_3 > c_2 > c_1$, v_2 has preferences $c_1 > c_2 > c_3 > c_4 > c_5$, v_3 has preferences (note the interleaving) $c_2 > c_3 > c_1 > c_4 > c_5$, and v_4 has preferences $c_4 > c_5 > c_3 > c_2 > c_1$.

Formally, there are many equivalent ways to capture this behavior, and we use the following as our definition. A collection V of votes (each a linear ordering P_i of the candidates) over candidate set C is said to be *single-peaked* exactly if there exists a linear ordering L over C such that for each triple of candidates a, b, and c, it holds that $(a \, L \, b \, L \, c \vee c \, L \, b \, L \, a) \Rightarrow (\forall i) \, [a \, P_i \, b \Rightarrow b \, P_i \, c]$.

The single-peaked model has been intensely studied, and has both strengths and limitations. On the positive side, it is an excellent rough model for a wide range of elections. Votes on everything from American presidential elections to US Supreme Court votes to hiring votes within a CS department are often shockingly close to reflecting single-peaked preferences. It certainly is a vastly more reasonable model in most settings than is assuming that all voters are random and independent, although the latter model has been receiving a huge amount of study recently. In fact, a wide range of scholarly studies have argued for the value of the single-peaked model [5, 10, 24], and the model is one of the first taught to students in positive (i.e., theoretical) political science courses. On the

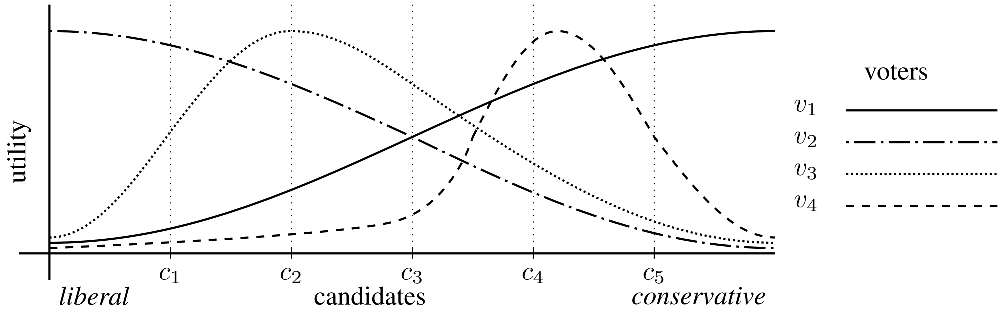


Figure 1: Example of a single-peaked electorate of four voters

other hand, some electorates certainly are driven by multidimensional concerns, and even a heavily unidimensional electorate may have a few out-of-the-box voters.

The single-peaked model also makes sense for approval voting [17]: There, a voter intuitively may be thought to have some utility threshold starting at which he or she approves of candidates. What this means is that each voter's "approved" candidates must be contiguous within society's linear order L.

Although we will assume that society's linear order is part of the input in our single-peaked winner, bribery, manipulation, and control problems, we mention in passing that given an election instance, one can in polynomial time tell whether the voters are single-peaked and when so can also in polynomial time compute a societal linear order instantiating the single-peakedness (Bartholdi and Trick [2] and Doignon and Falmagne [11] for linear-order preferences and Faliszewski et al. [17] for approval preferences).

3 Bypassing Winner-Problem Complexity

The main results sections of this paper study whether single-peakedness bypasses complexity-theoretic protections against attacks on elections. Before moving to those sections, we quickly present some results showing that single-peakedness also bypasses the complexity results some systems have for even telling who won. Unlike the "protection from attack" complexity-shield bypassings, which are in some sense bad news (for the security of the election systems), these "winner-hardness" complexity-shield bypassings are good news—taming the complexity of election systems such as Dodgson and Kemeny for the single-peaked case, despite the fact that they are known to have NP-hard winner problems in the general case.

For a given election system $\mathcal{E}$, the winner problem takes as input an election, (C, V), and a candidate $p \in C$, and asks if p is a winner in the election whose candidates are C and whose votes are V. When we speak of the single-peaked case of the winner problem, the input will also contain a linear order L relative to which the election is single-peaked. Note that the weakCondorcet winner problem is in P in the general case and thus certainly in the single-peaked case. Furthermore, something used often in our paper's proofs is the following standard fact about Condorcet voting and medians.

Fact 1. *Associate each voter with the candidate at the top of that voter's preference ordering. If we order the voters with respect to L in terms of that association, then if $\|V\|$ is odd, the weak-Condorcet and Condorcet winner set is the top preference of the median voter, and if $\|V\|$ is even, the weakCondorcet winner set is the set of all candidates who in L fall in the range, inclusively, between the top preferences of the two median voters (and if those two coincide, then that candidate is the Condorcet winner and otherwise there is no Condorcet winner).*

An immediate consequence is the well-known fact that for single-peaked elections, there is always at least one weak Condorcet winner (we are tacitly here assuming $C \neq \emptyset$). Since we earlier noted that the winner problem is in P for weakCondorcet elections, the following holds.

Theorem 1. *For each election system $\mathcal{E}$ that is weakCondorcet-consistent when restricted to single-peaked electorates, the winner problem is in* P *when restricted to single-peaked elections.*

Of course, for many such systems the winner problem is obviously in P even in general. Yet we do get some interesting consequences from Theorem 1.

Corollary 1. *When restricted to single-peaked electorates, the winner problems for Kemeny, Young, and weakDodgson elections are in* P.

In contrast, the general-case Kemeny winner problem problem was proven by Hemaspaandra et al. [22] to be Θ_2^p-complete. And we prove in the full version of this paper that the general-case winner problems for Young and weakDodgson elections are Θ_2^p-complete as well. Thus, Theorem 1 implies sharp complexity simplifications for these three election systems.

The "identify with weakCondorcet" approach that just worked on Young and weakDodgson elections does not apply to Dodgson and strongYoung elections. However, we have constructed direct algorithms that solve their winner problems in polynomial time in the single-peaked case.

Theorem 2. *When restricted to single-peaked electorates, the winner problems for Dodgson and strongYoung elections are in* P.

Our algorithm that shows this for Dodgson elections is a good example of the general technical theme of this paper: That single-peakedness often precludes combinatorial explosion. In this particular case, single-peakedness simplifies the seemingly exponential-sized search space over "series of exchanges to provide upper bounds on Dodgson scores," and will allow us to instead search over a polynomial-sized possibility space related to a particular, simple set of exchanges happening and limited to at most two voters.

Both claims in Theorem 2 contrast directly with the known Θ_2^p-completeness of the general case Dodgson [21] and strongYoung [25] winner problems, and thus reflect a substantial complexity simplification that holds when electorates are single-peaked.

4 Bribery of Single-Peaked Elections

This section shows that single-peakedness undercuts many, although not all, NP-hardness protections for bribery problems.

All bribery notions presented here, except negative approval bribery, are from the paper that started the complexity-theoretic study of bribery [16]. Given an election system $\mathcal{E}$, the $\mathcal{E}$-bribery problem takes as input C, V, $p \in C$, and $k \in \{0, 1, 2, \ldots\}$, and asks if, by changing the votes of at most k members of V, p can be made a winner of this election with respect to $\mathcal{E}$. That is the basic bribery problem. And it can be modified by any combination of the following items: "$" means each voter has a price (belonging to $\{1, 2, 3, \ldots\}$) and the question is whether there is some set of voters whose total price is at most k such that by changing their votes we can make p be a winner. The intuition for prices is that some voters can be swayed more easily than others. "Weighted" means each vote has a weight (belonging to $\{1, 2, 3, \ldots\}$), and each weight w vote is bribed as an indivisible object, but when applying $\mathcal{E}$, is viewed as w identical "regular" votes. For the case where V consists of linear orders, by "negative" we mean that if we bribe a voter then after the bribe the voter must not have p as his or her top choice unless p already was the top choice before the bribe. The intuition is that in negative bribery one is trying to stay under the radar by not directly helping one's candidate. For approval-vector votes, by "negative" we mean that when you bribe a voter, his or her after-bribe vector can approve p only if his or her before-bribe vector approved p. By “strongnegative” we mean that when you bribe a voter the voter after being bribed cannot approve p. These can occur in any combination, e.g., we can speak of Llull-negative-weighted-\$bribery.

When we speak of the single-peaked case of any of the above, we require that all bribes must result only in votes that are consistent with the input societal order L.

4.1 Approval-Bribery Results

As our main result for approval-bribery, we prove that the bribery protection that complexity gives there fails on single-peaked electorates.

Theorem 3 (Faliszewski et al. [16]). *Approval-bribery is* NP*-complete.*

Theorem 4. *Approval-bribery is in* P *for single-peaked electorates.*

The specific technical reason we can obtain polynomial-time bribery algorithms is that the NP-hardness proofs were based on the combinatorially rich structure of covering problems (whose core challenge is the “incomparability” of voters), but we use single-peakedness to create a “directional” attack on covering problems that has the effect of locally removing incomparability.

By the same general approach—using a “directional” attack to in the single-peaked setting tame the incomparability challenges of covering problems—we can establish the following two additional cases in which NP-hard bribery problems fall to P for the single-peaked case.

Theorem 5.

1. *Approval-negative-bribery and approval-strongnegative-bribery are* NP*-complete.*
2. *For single-peaked electorates, approval-negative-bribery and approval-strongnegative-bribery are in* P.

4.2 Llull-Bribery and Kemeny-Bribery Results

We now state the following eight-case result. The P cases below are proved by direct algorithmic attacks using the connection between weakCondorcet and median voters, and the NP-complete cases are shown by using the problem to capture a partition instance.

Theorem 6. *For single-peaked electorates, weakCondorcet-weighted-\$bribery, weakCondorcet-negative-weighted-bribery, and weakCondorcet-negative-weighted-\$bribery are* NP*-complete, and the remaining five weakCondorcet bribery cases are in* P.

Theorem 6 is most interesting not for what it says about weakCondorcet elections, but for its immediate consequences on other election systems, since all weakCondorcet-consistent election systems coincide for single-peaked electorates due to the nonemptiness of the set of weakCondorcet winners.

Corollary 2. *Let $\mathcal{E}$ be any election system that is weakCondorcet-consistent on single-peaked inputs. Then the three* NP*-completeness and five* P *results of Theorem 6 hold (for single-peaked electorates) for $\mathcal{E}$.*

From our discussions earlier in the paper, Corollary 2 applies to the Llull, Kemeny, Young, weakDodgson, Maximin, Schwartz, weakBlack, Fishburn, and the two variants of Nanson election systems. In light of this, Corollary 2 is quietly establishing a large number of claims about NP-hardness shields failing for single-peaked electorates. For example, we have the following claims.

Theorem 7 (Faliszewski et al. [16]). *Llull-bribery, Llull-\$bribery, Llull-weighted-bribery, and Llull-weighted-\$bribery are each* NP*-complete.*

Theorem 8 (follows from Corollary 2)**.** *For single-peaked electorates, Llull-bribery, Llull-$bribery, Llull-weighted-bribery, and Llull-weighted-$bribery are each in* P.

To the best of our knowledge, bribery of Kemeny elections has never been studied. Note, however, that the winner problem for any election system $\mathcal{E}$ many-one reduces to each of the eight types of bribery problems mentioned in Theorem 6, except with "weakCondorcet" replaced by "$\mathcal{E}$." This holds because we can ask whether the preferred candidate wins given the bribe limit of 0, and this captures the winner problem. So, from the known Θ_2^p-completeness of the winner problem for Kemeny elections [22], we have the following result, which gives us eight contrasts of hardness (three between Θ_2^p-hardness and NP membership and five between Θ_2^p-hardness and P membership).

Theorem 9 (corollary, in light of the comments just made, to Hemaspaandra et al. [22])**.** *For Kemeny elections, all eight types of bribery mentioned in Theorem 6 are Θ_2^p-hard.*

Theorem 10 (follows from Corollary 2)**.** *For single-peaked electorates, Kemeny-weighted-$bribery, Kemeny-negative-weighted-bribery, Kemeny-negative-weighted-$bribery are* NP*-complete, and the remaining five types of bribery of Kemeny elections are in* P.

5 Control of Single-Peaked Electorates

The control problems for elections ask whether by various types of changes in an election's structure a given candidate can be made a winner. The types of control that were introduced by Bartholdi et al. [4], and that (give or take some slight refinements) have been studied in subsequent papers, are addition/deletion/partition of voters/candidates. However, the only previous paper that studied the complexity of control for single-peaked electorates, Faliszewski et al. [17], focused exclusively on additions and deletions of candidates and voters.

We for the first time study the complexity of partition problems for the case of single-peaked electorates. And we show that for a broad range of election systems the control by partition of voters problem is in P for single-peaked electorates. Among the systems we do this for are Llull and Condorcet elections, whose control by partition of voters problem is known to be NP-complete for general electorates. Our proofs again work by using single-peakedness to tame combinatorial explosion—in this case, the number of partitions that must be examined is reduced from an exponential number of partitions to a polynomial number of classes of partitions each of which can be checked as a block.

The control by partition of voters problem for an election system $\mathcal{E}$ takes as input an election instance (C, V) and a candidate $c \in C$ and asks whether there is a partition of votes (V_1, V_2) such that if the "appropriate candidates" move forward from the preliminary elections (C, V_1) and (C, V_2) to a final election in which those candidates are voted on by V, then c "wins." How one clarifies the quoted strings determines the precised type of voter control one studies. In particular, one can study the nonunique-winner model or the unique-winner model. And as to the "appropriate candidates" move forward means, one can study the Ties Promote (TP) model (all winners of the preliminary elections move forward) or the Ties Eliminate (TE) model (only unique winners move forward). Our results hold for all four combinations of these models.

We will briefly mention control results about adding and deleting voters and candidates. The definitions of those are just what one would expect, and we refer the reader to Faliszewski et al. [19] for those definitions. The following is our main result for this section.

Theorem 11. *For weakCondorcet elections, (constructive) control by partition of voters is in* P *for single-peaked electorates.*

The technical challenge here is the exponential number of partitions, and our algorithm circumvents this by using single-peakedness to allow us to in effect structure that huge number of partitions into a polynomial number of classes of partitions such that for each class we can look just at the class rather than having to explore each of its member partitions. Let us note some consequences of this theorem.

Corollary 3. *Let $\mathcal{E}$ be any election system that is weakCondorcet-consistent on single-peaked inputs. Then for election system $\mathcal{E}$, (constructive) control by partition of voters is in* P *for single-peaked electorates. In particular, this holds for the election systems Llull, Kemeny, weakDodgson, Maximin, Schwartz, weakBlack, Fishburn, and the two variants of Nanson.*

For Llull elections, this provides a clear contrast with the known NP-completeness for that same control type in the general case. We now state a result that will quickly give us a number of additional contrasts between general-case control complexity and single-peaked control complexity.

Theorem 12. *For weakCondorcet elections, (constructive) control by adding voters and (constructive) control by deleting voters are each in* P *for single-peaked electorates.*

The full version of this paper contains similar results for Condorcet elections.

6 Manipulation of Single-Peaked Electorates

Faliszewski et al. [17] completely characterized, *for three-candidate elections*, which scoring protocols have polynomial-time constructive coalition weighted manipulation problems and which have NP-complete constructive coalition weighted manipulation problems. We achieve a far more sweeping dichotomy theorem—our result applies to all scoring protocols, regardless of the number of candidates. In the constructive coalition weighted manipulation problem, the input is the candidate set C, the nonmanipulative voters (each a preference order over C and a weight), the manipulative voters (each just a weight), and a candidate $p \in C$, and the question is whether there is a way of setting the preferences of the manipulative voters such that p is a winner under the given election rule when all the manipulative and nonmanipulative voters vote in a weighted election.

Our extension of this three-candidate, single-peaked electorate result to the case of any scoring protocol over single-peaked electorates is somewhat complicated. Yet, since it is a complete characterization—a dichotomization of the complexities, in fact—it is in some sense simply reflecting the subtlety and complexity of scoring systems.

Theorem 13. *Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ be an m-candidate scoring protocol and consider the constructive coalition weighted manipulation problem for single-peaked electorates.*

- *If $\alpha_2 > \alpha_{\lfloor \frac{m-1}{2} \rfloor + 2}$ and there exist integers $m_1, m_2 > 0$, $i_1, i_2 > 1$ such that $m_1 + m_2 + 1 = m$, $i_1 \leq m_1 + 1$, $i_2 \leq m_2 + 1$, and $(\alpha_1 - \alpha_{i_1})(\alpha_1 - \alpha_{i_2}) > (\alpha_{i_1} - \alpha_{i_1+1})(\alpha_{i_2} - \alpha_{i_2+1})$, then the problem is* NP*-complete.*
- *If $\alpha_2 = \alpha_{\lfloor \frac{m-1}{2} \rfloor + 2}$ and $\alpha_1 > \alpha_2 > \alpha_m$ and ($\alpha_2 > \alpha_{m-1}$ or $\alpha_1 - \alpha_m > 2(\alpha_2 - \alpha_m)$), then the problem is* NP*-complete.*
- *In all other cases, the problem is in* P.

The "P" cases of Theorem 13's dichotomy align with our theme of single-peakedness often foiling combinatorial protections.

7 Related Work and Additional Discussion

The two papers most related to our work are Walsh [26] and Faliszewski et al. [17]. Walsh's paper first raised the issue of the effect of single-peaked electorates on manipulation, and for the particular case he looked at—weighted coalition manipulation under single transferable vote elections—he showed that manipulation remains hard even for single-peaked electorates. Faliszewski et al. showed cases where single-peakedness removes complexity shields against manipulation, and also opened the study of (nonpartition) control. Our paper in contrast with Walsh's stresses cases where single-peakedness removes combinatorial protections. And we go beyond Faliszewski et al. by for the first time studying bribery of single-peaked electorates and partition-control of single-peaked electorates.

Although [26] and [17] are by far the most related work, other work is much worth mentioning. Bartholdi and Trick [2], Doignon and Falmagne [11], and Escoffier et al. [14] provided efficient algorithms for finding single-peaked orderings. And Conitzer [8] studied the effect of single-peaked electorates on preference elicitation. Two of the papers just mentioned [14, 8] raise the issue of nearly single-peaked electorates, and we commend as a particularly important open issue the question of what effect nearly single-peaked electorates have on complexity.

The literature now contains many papers on the complexity (when single-peaked preferences are not assumed) of manipulation and control (as a pointer to those, see [18] and the citations therein), and contains a few papers on the younger topic of the complexity of bribery (e.g., Faliszewski et al. [16] and Faliszewski et al. [19]). Although the nonunique-winner model and the unique-winner model very typically have the same complexity results, Faliszewski et al. [15] (drawing also on Conitzer et al. [9]) show that this is not always the case.

A worry that comes immediate to the minds of social choice theorists can be expressed as follows: Since it is known that, for single-peaked electorates, "median voting" leaves voters with voting sincerely being an optimal strategy, single-peaked elections are not interesting in terms of other election systems, since median voting should be used. A detailed discussion of this worry would itself fill a paper. But we briefly mention why the above objection is not as serious as it might at first seem. First, the nonmanipulability claims regarding single-peaked elections and median voting are about manipulability, and so say nothing at all about, for example, control. Indeed, weakCondorcet in effect is a type of median voting on single-peaked electorates, and our partition of voters algorithm makes it clear that control can be exercised in interesting ways. Second, even if median voting does have nice properties, the simple truth is that in the real world, society—for virtually all elections and electorates—has chosen (perhaps due to transparency, comfort, or tradition) to use voting systems that clash sharply with median voting. The prominence of plurality voting is the most dramatic such case. So since in the real world we do use a rich range of election systems, it does make sense to understand their behavior. Third, one must be very careful with terms such as "strategy-proof." The paper people most commonly mention as showing that median voting is strategy-proof is Barberà [1]. But that paper's results are about "social choice *functions*" (election rules that always have exactly one winner), not—as this paper is—about election rules that select a set of winners. So one cannot simply assume that for our case median voting (say, weakCondorcet elections) never gives an incentive to misrepresent preferences. We should further stress that discussions of strategy-proofness typically assume that manipulators come in with complete preference orders, but in the Bartholdi et al. [3] model (which this paper and most complexity papers use when studying manipulation), the manipulative coalition is a blank slate with its only goal being to make a certain candidate p be a winner.)

8 Conclusions

The theme of this paper is that single-peaked electorates often tame combinatorial explosion. We saw this first for the case of the winner problem. In that case, this taming is good. It shows that for single-peaked electorates, election systems such as Kemeny have efficient winner algorithms, despite their Θ_2^p-hardness in the general case. But then for bribery and control (and in part, manipulation), we saw many cases where NP-hard problems fell to polynomial time for single-peaked electorates, via algorithms that bypassed the general-case combinatorial explosions of covers and partitions. Since those NP-hardness results were protections against attacks on elections, our results should serve as a warning that those protections are at their very core dependent on the extreme flexibility of voter preference collections the general case allows. For single-peaked electorates, those protections vanish.

Acknowledgments This work was supported in part by DFG grants BR-2312/3-2 and BR-2312/6-1, NSF grants CCF-0426761, IIS-0713061, and CCF-0915792, the European Science Foundation's EUROCORES program LogICCC, and Friedrich Wilhelm Bessel Research Awards to Edith Hemaspaandra and Lane A. Hemaspaandra. This work was done in part during visits of Edith Hemaspaandra and Lane A. Hemaspaandra to the Universities of Dusseldorf and Munich. We thank Piotr Faliszewski, Felix Fischer, Paul Harrenstein, Ariel Procaccia, Jörg Rothe, Hans Georg Seedig, Troels Sørensen, and the anonymous COMSOC and AAAI referees for enjoyable discussions and helpful suggestions.

References

[1] S. Barberà. An introduction to strategy-proof social choice functions. *Social Choice and Welfare*, 18(4): 619–653, 2001.

[2] J. Bartholdi, III and M. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165–169, 1986.

[3] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[4] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[5] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[6] F. Brandt, M. Brill, E. Hemaspaandra, and L. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. Technical Report TR-955, Dept. of Comp. Sci., Univ. of Rochester, Rochester, NY, 2010.

[7] J.-A.-N. de Caritat, Marquis de Condorcet. *Essai sur l'Application de L'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix*. 1785. Facsimile reprint of original published in Paris, 1972, by the Imprimerie Royale.

[8] V. Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35:161–191, 2009.

[9] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[10] O. Davis, M. Hinich, and P. Ordeshook. An expository development of a mathematical model of the electoral process. *American Political Science Review*, 54(2):426–448, 1970.

[11] J. Doignon and J. Falmagne. A polynomial time algorithm for unidimensional unfolding representations. *Journal of Algorithms*, 16:218–233, 1994.

[12] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference*, pages 613–622. ACM Press, 2001.

[13] E. Ephrati and J. Rosenschein. A heuristic technique for multiagent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1):13–67, 1997.

[14] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *Proceedings of the 18th European Conference on Artificial Intelligence*, pages 366–370. IOS Press, 2008.

[15] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 983–990. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[16] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[17] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. In *Proc. of the 12th Conf. on Theoretical Aspects of Rationality and Knowledge*, pages 118–127. ACM, 2009.

[18] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, pages 375–406. Springer, 2009.

[19] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[20] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[21] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[22] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[23] D. Pennock, E. Horvitz, and C. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 729–734. AAAI Press, 2000.

[24] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, 2007.

[25] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, 2003.

[26] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 3–8. AAAI Press, 2007.

Felix Brandt and Markus Brill
Institut für Informatik
Technische Universität München
85748 Garching bei München, Germany
Email: `{brandtf,brill}@in.tum.de`

Edith Hemaspaandra
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623, USA
Email: `eh@cs.rit.edu`

Lane A. Hemaspaandra
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA
Email: `lane@cs.rochester.edu`

The Efficiency of Fair Division with Connected Pieces

Yonatan Aumann and Yair Dombb

Abstract

We consider the issue of fair division of goods, using the cake cutting abstraction, and aim to bound the possible degradation in social welfare due to the fairness requirements. Previous work has considered this problem for the setting where the division may allocate each player any number of unconnected pieces. Here, we consider the setting where each player must receive a single connected piece. For this setting, we provide tight bounds on the maximum possible degradation to both utilitarian and egalitarian welfare due to three fairness criteria – proportionality, envy-freeness and equitability.

1 Introduction

Cake Cutting. The problem of fair division of goods is the subject of extensive literature in the social sciences, law, economics, game theory and more. The famous "cake cutting" problem abstracts the fair division problem in the following way. There are n players wishing to divide between themselves a single "cake". The different players may value differently the various sections of the cake, e.g. one player may prefer the marzipan, another the cherries, and a third player may be indifferent between the two. The goal is to obtain a "fair" division of the cake amongst the players. There are several possible definitions to what constitutes a "fair" division, with *proportionality*, *envy-freeness* and *equitability* being the major fairness criteria considered (these notions will be defined in detail later). Many previous works considered the problem of obtaining a fair devision under these (and other) criteria.

Social Welfare. While fairness is clearly a major consideration in the division of goods, another important consideration is the social welfare resulting from the division. Clearly, a division may be envy-free but very inefficient, e.g. in the total welfare it provides to the players. Accordingly, the question arises what, if any, is the tradeoff between these two desiderata? How much social welfare does one have to sacrifice in order to achieve fairness? The answer to this question may, of course, depend on the exact definition of fairness, on the one hand, and the social welfare of interest, on the other.

The first analysis of such questions was provided in [CKKK09], where Caragiannis et al. consider the three leading fairness criteria – proportionality, envy-freeness and equitability – and quantify the possible loss in utilitarian social welfare due to such fairness requirements. Here we continue this line of research, extending the results in two ways. Firstly, the [CKKK09] analysis allows dividing the cake into any number of pieces, possibly even infinite. Thus, each player may get a collection of pieces, rather than a single one. While this may be acceptable in some cases, it may not be so in others, or at least highly undesirable , e.g. in the division of real estate, where players naturally prefer getting a connected plot. Similarly, in the cake scenario itself, allowing unconnected pieces may lead to a situation where, in Stromquist's words [Str80], "a player who hopes only for a modest interval of the cake may be presented instead with a countable union of crumbs". Accordingly, in this work, we focus on divisions in which each player gets a single connected piece of the cake. In addition, we consider both the utilitarian and the egalitarian social welfare functions, whereas Caragiannis et al. considered only utilitarian welfare. For each of these welfare functions, we give tight bounds on the possible loss in welfare due to the three fairness criteria.

1.1 Definitions and Notations

We consider a rectangular cake that can be divided by making parallel cuts. The cake can thus be represented by the interval $[0,1]$, where each cut is some point $p \in [0,1]$. The cake needs to be divided to n players (we use the notation $[n]$ for the set $\{1,\ldots,n\}$), each of which has a valuation function $v_i(\cdot)$ assigning a non-negative value to every possible interval of the cake. As customary, we require that for all i, $v_i(\cdot)$ is a nonatomic measure on $[0,1]$ having $v_i(0,1) = 1$. Every set of valuation functions $\{v_i(\cdot)\}_{i=1}^n$ defines an *instance* of the cake cutting problem.

Since we consider only divisions in which every player gets a single connected interval, a division of the cake to n players can be represented by a vector

$$x = (x_1, \ldots, x_{n-1}, \pi) \in [0,1]^{n-1} \times S_n$$

with $0 \leq x_1 \leq x_2 \leq \cdots \leq x_{n-1} \leq 1$. Here, x_i determines the position of the i-th cut, and π is a permutation that determines which piece is given to which player. For convenience, we denote $x_0 = 0$ and $x_n = 1$, so we can write that player $i \in [n]$ receives the interval $(x_{\pi(i)-1}, x_{\pi(i)})$. We use the notation $u_i(x)$ for the utility that player i gets in the division x, i.e. $u_i(x) = v_i(x_{\pi(i)-1}, x_{\pi(i)})$. We denote by X the set of all possible division vectors, and note that X is a compact set.

Fairness Criteria. We say that a division $x \in X$ is:

- **Proportional** if every player gets at least $\frac{1}{n}$ of the cake (by her own valuation). Formally, x is a proportional division if for all $i \in [n]$, $u_i(x) \geq \frac{1}{n}$.
- **Envy-Free** if no player prefers getting the piece alloted to any of the other players. Formally, x is an envy-free division if for all $i \neq j \in [n]$, $u_i(x) = v_i(x_{\pi(i)-1}, x_{\pi(i)}) \geq v_i(x_{\pi(j)-1}, x_{\pi(j)})$.
- **Equitable** if all the players get the exact same utility in x (by their own valuations). Formally, x is an equitable division if for all $i, j \in [n]$, $u_i(x) = u_j(x)$.

Stromquist [Str80], showed that for every instance of the cake cutting problem there exists an envy-free division with connected pieces. Since one can easily observe that every envy-free division is in particular proportional, this implies that such proportional divisions also always exist. In this paper we show (Theorem 6) that equitable divisions also always exist for connected pieces (for the case where players need not get a single interval, this is well known).

Social Welfare Functions. For a division $x \in X$, we denote by $u(x)$ the utilitarian social welfare of x, i.e.

$$u(x) = \sum_{i \in [n]} u_i(x) \ .$$

Likewise, we denote by $eg(x)$ the egalitarian social welfare of x, which is

$$eg(x) = \min_{i \in [n]} u_i(x) \ .$$

Note that both these social welfare functions are continuous and thus have maxima in X.

The Price of Fairness. As described above, we aim to quantify the degradation in social welfare due to the different fairness requirements. This is captured by the notion of *Price of Fairness*, in its three forms – *Price of Proportionality*, *Price of Envy-freeness* and *Price of Equitability*, defined as follows. The *Price of Proportionality* (resp. Envy-Freeness, Equitability) of a cake-cutting instance I, with respect to some predefined social welfare function, is defined as the ratio between the maximum possible social welfare for the instance, taken over all possible divisions, and the maximum social welfare attainable when divisions must be proportional (resp. envy-free, resp. equitable). When considering divisions with connected pieces, this restriction is applied to both maximizations. For example, if $X_{EF} \subseteq X$ is the set of all (connected) envy-free divisions of an instance, the egalitarian Price of Envy-Freeness for this instance is

$$\frac{\max_{x \in X} eg(x)}{\max_{y \in X_{EF}} eg(y)}\,.$$

In this work we show bounds on the maximum utilitarian and egalitarian Price of Proportionality, Envy-Freeness and Equitability of *any* instance.

1.2 Results

We analyze the utilitarian and egalitarian Price of Proportionality, Envy-Freeness and Equitability for divisions with connected pieces. We provide tight bounds (in some cases, up to an additive constant factor) for all six resulting cases. The results are summarized in Table 1; the last row presents the relevant previous results by Caragainnis et al. in [CKKK09], for comparison. The meaning of the upper bounds is that the respective price of fairness of *any* possible instance is never greater than the bound. The meaning of the lower bound is that there *exists* an instance that exhibits at least this price of fairness (for the respective class).

<table>
<tr><td>Price of:</td><td>Proportionality</td><td>Envy-Freeness</td><td>Equitability</td><td></td></tr>
<tr><td>Utilitarian</td><td colspan="2">UB: $\frac{\sqrt{n}}{2}+1-o(1)$
LB: $\frac{\sqrt{n}}{2}$</td><td>UB: n
LB: $n-1+\frac{1}{n}$</td><td rowspan="2">connected pieces (this work)</td></tr>
<tr><td>Egalitarian (tight)</td><td>1</td><td>$\frac{n}{2}$</td><td>1</td></tr>
<tr><td>Utilitarian</td><td>UB: $2\sqrt{n}-1$
LB: $\frac{\sqrt{n}}{2}$</td><td>UB: $n-\frac{1}{2}$
LB: $\frac{\sqrt{n}}{2}$</td><td>UB: n
LB: $\frac{(n+1)^2}{4n}$</td><td>non-connected pieces [CKKK09]</td></tr>
</table>

Table 1: All results

Utilitarian Welfare. For the utilitarian social welfare, we show an upper bound of $\frac{\sqrt{n}}{2} + 1 - o(1)$ on the price of envy-freeness, for *any* possible instance. This, we believe, is the first non-trivial upper bound on the Price of Envy-Freeness. It seems that such bounds are hard to obtain since on the one hand we need to consider the "best" possible envy-free division, while on the other hand no efficient method for explicitly constructing any envy-free divisions is known. We show that the same upper bound also applies to the Price of Proportionality.

For the Price of Equitability, we show that it is always bounded by n (though simple, this does require a proof since an equitable division need not even give each player $1/n$). We also provide an almost matching lower bound, showing that for any n there exists an instance with utilitarian Price of Equitability arbitrarily close to $n - 1 + \frac{1}{n}$.

Egalitarian Welfare. When considering the egalitarian social welfare, we show that there is no price for either proportionality or equitability. That is, for any instance there exist both proportional and equitable divisions for which the minimum amount any player gets is no less than if there were no fairness requirements. While perhaps not surprising, the proof for the Price of Equitability is somewhat involved, especially since we require that the divisions be with connected pieces. We note that we are not aware of any previous proof that altogether establishes the existence of an equitable division with connected pieces.

For the Price of Envy-Freeness, we show that it is bounded by $n/2$, and provide a matching family of instances that exhibits this price, for any n.

Paper Organization. In Section 2, we present bounds on the Price of Proportionality and the Price of Envy-Freeness. We begin in 2.1 by presenting the upper bound on the utilitarian Price of Envy-Freeness, and complement it by an example already given in Caragiannis et al. [CKKK09], which is tight up to a small additive factor. Both these upper and lower bounds apply also to the utilitarian Price of Proportionality. In 2.2 we show a simple upper bound of $\frac{n}{2}$ for the egalitarian Price of Envy-Freeness, together with a matching (tight) lower bound. We also show that the egalitarian Price of Proportionality is trivially 1. In Section 3 we present bounds on the Price of Equitability. In addition to the (mentioned above) proof that the egalitarian price is 1, we provide a simple upper bound of n on the utilitarian Price of Equitability, together with a lower bound of $n-1+\frac{1}{n}$. In Section 4 we consider the reverse question to that of the Price of Fairness – namely, how much fairness may one have to give up to achieve social optimality. Finally, we conclude this work and present some open questions in Section 5.

1.3 Related Work

The problem of fair division dates back to the ancient times, and takes many forms. The piece of property to be divided may be divisible or indivisible: Divisible goods can be "cut" into pieces of any size without destroying their value (like a cake, a piece of land, or an investment account), while indivisible goods must be given in whole to one person (e.g. a car, a house, or an antique vase). Since such items cannot be divided, the problem is usually to divide a *set* of such goods between a number of players. Fair division may also relate to the allocation of chores (of which every party likes to get as little as possible); this problem is of a somewhat different flavor from goods allocation, and also has the divisible and indivisible variants.

Modern mathematical treatment of fair division started at the 1940s [Ste49], and was initially concerned mainly with finding methods for allocation of divisible goods. Different algorithms – both discrete and continuous ("moving knife algorithms") – were presented (e.g. [Str80, EP84] and [BT95], which also surveys older algorithms), as well as non-constructive existence theorems [DS61, Str80]. In the past fifteen years, several books appeared on the subject [BT96, RW98, Mou04]. Following the evaluation and cut queries model suggested by Robertson and Webb [RW98], much attention was given to the question of lower bounds on the number of steps or cuts required for such divisions in this and other models [MIBK03, EP06, SW03, Str08, Pro09]. In particular, Stromquist [Str08] proves that no finite protocol (even unbounded) can be devised for an envy-free division of a cake among three or more people in which each player receives a connected piece. However, we note that this result applies only to the model presented in that work (which resembles the one suggested by Robertson and Webb), and not for cases where, for example, some mediator has full information of the players' valuation functions and proposes a division based on this information.

Unlike most of the work on cake cutting, the different notions of the price of fairness are not concerned with *procedures* for obtaining divisions, but rather with the *existence* of divisions with different properties (relating to social optimality and fairness). These notions, namely the Price of Proportionality, the Price of Envy-Freeness and the Price of Equitability, were first presented in a recent paper by Caragiannis et al. [CKKK09]. This line of work has some resemblance to the line of work on the Price of Stability [ADK$^+$04], which attracted much attention in the past decade. The work in [CKKK09] analyzes the price of fairness (via the above three measures) with the utilitarian welfare function for divisible and indivisible goods and chores, giving tight bounds (up to a constant multiplicative factor) in most cases. However, unlike in this work, no special attention was given to the case of connected pieces in divisible goods. The results of [CKKK09] for divisible goods are summarized in the last row of Table 1.

2 The Price of Envy-Freeness and Proportionality

2.1 Utilitarian Welfare

Theorem 1. *For every cake-cutting instance with n players, the utilitarian Price of Envy-Freeness with connected pieces is bounded from above by $\frac{\sqrt{n}}{2} + 1 - o(1)$.*

In fact, we prove an even stronger claim: The above bound applies not only to the distance of the "best" envy-free division from utilitarian optimality, but also to the distance from (utilitarian) optimality of *any* envy-free division.

Proof. Let x be an envy-free division of the cake, and $u(x) = \sum_{i\in[n]} u_i(x)$ its utilitarian social welfare. We show that any other division to connected pieces y has $u(y) \leq \left(\frac{\sqrt{n}}{2} + 1 - \frac{n}{4n^2-4n+2\sqrt{n}}\right) \cdot u(x)$. Our proof is based on the following key observation:

> *Assume that for some $i \in [n]$, $u_i(y) \geq \alpha \cdot u_i(x)$. Since i values any other piece in the division x at most as much as her own, it has to be that in y, i gets an interval that intersects pieces that belonged to at least $\lceil\alpha\rceil$ different players (possibly including i herself).*

We will say that in the division y, player i gets the j-th cut of x if in y, i is given a piece starting at a point $p < x_j$ and ending at the point $p' > x_j$. A more formal statement of our observation is therefore that if in y, i gets at most α cuts of x, it holds that $u_i(y) \leq (\alpha + 1) \cdot u_i(x)$. We can thus bound the ratio $\frac{u(y)}{u(x)}$ by the solution to the following optimization problem, which aims to find values $\{u_i(x)\}_{i=1}^n$ and $\{\alpha_i\}_{i=1}^n$ (the number of cuts of x each player gets) that maximize this ratio.

$$\begin{aligned}
&\text{maximize} && \frac{\sum_{i=1}^n (\alpha_i + 1)u_i(x)}{\sum_{i=1}^n u_i(x)} && && (1)\\
&\text{subject to} && \sum_{i=1}^n \alpha_i = n - 1 && && \\
& && u_i(x) \geq \frac{1}{n} && \forall 1 \leq i \leq n && (2)\\
& && (\alpha_i + 1)u_i(x) \leq 1 && \forall 1 \leq i \leq n && (3)\\
& && \alpha_i \in \{0, \ldots, n-1\} && \forall 1 \leq i \leq n &&
\end{aligned}$$

(2) is a necessary condition for the envy-freeness of x that provides a lower bound for the denominator, and (3) is equivalent to $u_i(y) \leq 1$.

We therefore concentrate on bounding the solution to the above optimization problem. To this end, the following observations are useful:

1. For any choice of values $\{u_i(x)\}_{i=1}^n$, the optimal assignment for the α_i variables is greedy, i.e. giving each player i, in non-increasing order of $u_i(x)$ the maximum possible value for α_i that does not violate any of the constraints. (This holds since otherwise there are players i, j with $u_i(x) > u_j(x)$ and $\alpha_j \geq 1$ such that increasing α_i by one at the expense of α_j is feasible and yields an increase of $u_i(x) - u_j(x) > 0$ in the numerator of (1), without affecting the denominator.) We thus can divide the players into two groups: Those with "high" $u_i(x)$ values, who receive strictly positive α_i values, and those with "low" $u_i(x)$ values, for which $\alpha_i = 0$.

2. Since the players with low $u_i(x)$ values add the same amount to both the numerator and the denominator in the objective function, maximum is obtained when these values are minimized; i.e. in the optimal solution $u_i(x) = \frac{1}{n}$ for all these players.

3. The solution to the problem above is clearly bounded from above by the solution to the same problem where the α_i variables need not have integral values. Clearly, in the optimal solution to such a problem, all the players with $\alpha_i > 0$ have $(\alpha_i + 1)u_i(x) = 1$.

We can thus bound the solution to our optimization problem by the solution to the following problem. Let K be a variable that denotes the number of players that will have $\alpha_i > 0$; by observation (3) above, for every such player, $(\alpha_i + 1)u_i(x) = 1$, and thus their total contribution to the numerator is K. We therefore seek a solution for:

$$\text{maximize} \qquad \frac{K + (n-K)\cdot\frac{1}{n}}{\sum_{i=1}^{K} u_i(x) + (n-K)\cdot\frac{1}{n}} \tag{4}$$

$$\text{subject to} \qquad \sum_{i=1}^{K}\left(\frac{1}{u_i(x)} - 1\right) = n - 1 \tag{5}$$

$$K \leq n$$

It can be verified (e.g. using Lagrange multipliers) that for any value of $K \leq n$ this is maximized when $u_i(x) = u_j(x)$ for all $i, j \in [K]$, i.e. when $u_i(x) = \frac{K}{n-K+1}$ for all $i \in [K]$. We thus conclude that the maximum solution to the above problem maximizes the ratio

$$\frac{K + (n-K)\cdot\frac{1}{n}}{K\cdot\frac{K}{n+K-1} + (n-K)\cdot\frac{1}{n}} \ ;$$

by elementary calculus this is maximized at $K = \sqrt{n}$, where the value is

$$\begin{aligned}\frac{(n\sqrt{n} + n - \sqrt{n})(n + \sqrt{n} - 1)}{n^2 + (n-\sqrt{n})(n+\sqrt{n}-1)} &= \frac{(n^2\sqrt{n} - n\sqrt{n} + \frac{1}{2}n) + (2n^2 - 2n + \sqrt{n}) - \frac{1}{2}n}{2n^2 - 2n + \sqrt{n}} \\ &= \frac{\sqrt{n}}{2} + 1 - \frac{n}{4n^2 - 4n + 2\sqrt{n}} = \frac{\sqrt{n}}{2} + 1 - o(1) \ ,\end{aligned}$$

as stated. □

Since every envy-free division is in particular proportional, we immediately get that the bound on the utilitarian Price of Envy-Freeness also applies to the Price of Proportionality:

Corollary 2. *For every cake-cutting instance with n players, the utilitarian Price of Proportionality in connected pieces is bounded from above by $\frac{\sqrt{n}}{2} + 1 - o(1)$.*

We conclude by showing that these bounds are essentially tight (up to a small additive factor). The construction we show is identical to the one in [CKKK09], and we provide it here again for completeness.

Proposition 3. *The utilitarian Price of Proportionality (and thus also the utilitarian Price of Envy-Freeness) in connected pieces is larger than* $\frac{\sqrt{n}}{2}$*.*

Proof. For some integer m, consider $n = m^2$ players with the following valuation functions. For $i = 1, \ldots, \sqrt{n}$, player i assigns a value of 1 to the piece $(\frac{i-1}{\sqrt{n}}, \frac{i}{\sqrt{n}})$ and 0 to the rest of the cake (we call these players the "focused players"). All other players (players $i = (\sqrt{n}+1), \ldots, n$, the "indifferent players") assign a uniform value to the entire cake. In any proportional division, the indifferent players must get a total of at least $\frac{n-\sqrt{n}}{n}$ of the physical cake, and their total utility is less than 1. This leaves the focused players with at most $\frac{1}{\sqrt{n}}$ of the physical cake, and so they obtain (together) a total utility of at most 1; the utilitarian value of a proportional division is therefore less than 2. On the other hand, the division giving each of the focused players the entire interval they desire (and leaving nothing to the indifferent players) has a utilitarian social welfare of $\sqrt{n}$. The Price of Proportionality for this case is therefore larger than $\frac{\sqrt{n}}{2}$, as stated. □

2.2 Egalitarian Welfare

Proposition 4. *For every cake-cutting instance, the egalitarian Price of Proportionality is* 1*.*

Proof. Let x be a proportional division, and y the egalitarian optimal division. By proportionality, every player i has $u_i(x) \geq \frac{1}{n}$, and thus $eg(x) \geq \frac{1}{n}$. Since y is the egalitarian optimal division, we have that for every $i \in [n]$, $u_i(y) \geq eg(y) \geq eg(x) \geq \frac{1}{n}$; this implies that y is proportional as well. □

Theorem 5. *The egalitarian Price of Envy-Freeness for cake-cutting instances with* n *players and connected pieces is* $\frac{n}{2}$*. In particular, this is also an upper bound on the egalitarian Price of Envy-Freeness for* n *players and non-connected pieces.*

Proof. First, note that if the egalitarian optimal division is itself envy-free, the Price of Envy-Freeness is 1, and that every division with egalitarian welfare of $\frac{1}{2}$ is envy-free. We therefore assume that this is not the case, and that in the egalitarian optimal y division some player i has $u_i(y) < \frac{1}{2}$. Let x be some envy-free division, then x is in particular proportional and thus has $u_i(x) \geq \frac{1}{n}$; the upper bound follows.

It remains to show a lower bound for the connected case. Let $\epsilon > 0$ be an arbitrarily small constant, and consider n players with the following valuation functions. For $i = 1, \ldots, (n-1)$, player i assigns a value of $\frac{1}{2} + \epsilon$ to the piece $(i - \epsilon, i + \epsilon)$ (her "favorite piece"), a value of $\frac{1}{2} - \epsilon$ to the piece $(1 - \frac{2i+1}{2n} - \epsilon, 1 - \frac{2i+1}{2n} + \epsilon)$ (her "second-favorite piece"), and value of 0 to the rest of the cake. Finally, player n assigns a uniform value to the entire cake.

In order for player n to get utility of α, this player needs to receive an α fraction of the cake (in physical size). However, every connected piece of physical size at least $\frac{1}{n} + 2\epsilon$ necessarily contains some other player's "favorite piece", and it is immediate that if a single player receives the entire favorite piece of another player, there is envy. Thus, in every envy-free division of the cake, player n gets utility of less than $\frac{1}{n} + 2\epsilon$. However, there exists a division in which every player gets utility of at least $\frac{1}{2} - \epsilon$. Such a division is achieved by giving players $i = 1 \ldots \lfloor \frac{n-1}{2} \rfloor$ their favorite pieces, players $i = (\lfloor \frac{n-1}{2} \rfloor + 1) \ldots (n-1)$ their second-favorite pieces, and player n the interval $(\frac{1}{2} + \epsilon, 1)$ (the remaining parts of the cake can be given to any of the players closest to them). The stated bound follows as ϵ approaches zero. □

3 The Price of Equitability

In order to talk about the Price of Equitability, we first have to make sure that the concept is well-defined. When non-connected pieces are concerned, it is known that every cake cutting instance has an equitable division [DS61]. However, the proof of Dubins and Spanier allows a "piece" of the cake to be any member of the σ-algebra of subsets, which is quite far from our restricted case of pieces that are all single intervals. Another result by Alon [Alo87] establishes the existence of an equitable division giving every player exactly $\frac{1}{n}$ by each measure; however, such a division may require up to $n^2 - 1$ cuts. The question thus arises whether equitable divisions with connected pieces always exist; to the best of our knowledge, this question has not been addressed before, and we answer it here to the affirmative. Furthermore, we show that such a division requires no sacrifice of egalitarian welfare.

Theorem 6. *For every cake-cutting instance there exists an equitable division of the cake with connected pieces. Furthermore, there always exists such a division in which the egalitarian social welfare is as high as possible in any division with connected pieces.*
This holds even for cake cutting instances that do not have $v_i(0,1) = 1$ for all i (i.e. even if some players' valuation of the entire cake is not 1).

Proof. Recall that the egalitarian welfare is a continuous function and X is compact, and thus $eg(\cdot)$ has a maximum in X; we denote $OPT = \max_{x \in X} eg(x)$. We also denote by $Y \subset X$ the set of divisions with egalitarian value OPT, i.e.

$$Y = \left\{ y = (y_1, \ldots, y_{n-1}, \pi) \in X \;\middle|\; eg(y) = OPT \right\} .$$

We note that Y is a compact set; this follows from the fact that it is a closed subset of X (which is compact itself). To show that Y is closed, we show that $\overline{Y} = X \setminus Y$ is open. Let $z \in \overline{Y}$ be some division not in Y; then the division z must have egalitarian value smaller than OPT and in particular there must exist a player i and $\epsilon > 0$ such that $u_i(z) \leq OPT - \epsilon$. Since player i's valuation of the cake is a nonatomic measure, there must exist $\delta_L, \delta_R > 0$ such that extending i's piece to the interval $(z_{\pi(i)-1} - \delta_L, z_{\pi(i)} + \delta_R)$ increases i's utility (compared to the original division z) by less than ϵ. Therefore, in the ball of radius $\delta = \min\{\delta_L, \delta_R\}$ around z (e.g. in L_∞), every division still gives i utility smaller than OPT, and thus this ball does not intersect Y. It thus follows that $\overline{Y}$ is an open set, and so Y is closed and compact.

Recall that our aim is to show that Y contains an equitable division; to that end, we define a function $\Delta : Y \to \mathbb{R}$ by setting

$$\Delta(y) = \max_{i,j \in [n]} \left\{ u_i(y) - u_j(y) \right\} = \max_{i \in [n]} \left\{ u_i(y) - OPT \right\} .$$

We complete the proof by showing that for any ϵ, there exists a devision $y^{(\epsilon)} \in Y$, such that $\Delta(y^{(\epsilon)}) \leq \epsilon$. Since Y is a compact set and $\Delta(\cdot)$ is continuous, the image of Y is also compact. We therefore conclude that there must be some $y^* \in Y$ with $\Delta(y^*) = 0$ (since the image of Y is in particular a closed subset of $\mathbb{R}$ containing a point $p < \epsilon$ for every $\epsilon > 0$); such y^* is clearly equitable.

It remains to prove that for any ϵ, $y^{(\epsilon)}$ exists. We prove this by induction on the number of players n. For $n = 1$ there is only one possible division, which obtains exactly OPT for the single player. Assume for $n - 1$, we prove for n. Let y be any division in Y (assuming w.l.o.g. that y uses the identity permutation). We first construct a division y' such that for $i = 1, \ldots, n-1$, $u_i(y') = OPT$, by sequentially moving the border y'_i (between players i and $i+1$) to the left as far as possible while keeping that $u_i(y') \geq OPT$. This is possible since in y, $u_i(y) \geq OPT$ and the borders only need to move to the left. Consider the resulting

y'. If $u_n(y') \leq OPT + \epsilon$ we are finished; otherwise, let y'' be the division obtained from y' by moving the border y''_{n-1} (between players $n-1$ and n) as far right as necessary so that $u_n(y'') = OPT + \epsilon$. Now, omit the rightmost piece (that of player n), and consider the $(n-1)$-player cake cutting problem, on the remaining cake. (Note that the players' valuation of the entire new cake need not be identical to their valuation of the original cake, and that the new cake has a different set Y' of egalitarian-optimal divisions.)

Now, it cannot be the case that for this new problem the egalitarian maximum is more than OPT, as that would induce an egalitarian maximum greater than OPT for the entire problem. On the other hand, egalitarian value of OPT is clearly attainable, as it is obtained by y'' (reduced to the first $n-1$ players). Hence, OPT is also the egalitarian maximum for the new $(n-1)$-player problem. Thus, by the inductive hypothesis, there exists a division for this problem that obtains egalitarian welfare OPT and such that no player gets more than $OPT+\epsilon$. Combining this solution with the piece $(y''_{n-1}, 1)$ given to player n, we obtain $y^{(\epsilon)} \in Y$, such that no player gets more than $OPT + \epsilon$. □

Theorem 7. *The utilitarian Price of Equitability in connected pieces is upper-bounded by n, and for any n there is an example in which it is arbitrarily close to $n - 1 + \frac{1}{n}$.*

Proof. We begin by showing an upper bound on the utilitarian Price of Equilibility. From Theorem 6 we have that there always exists an equitable egalitarian-optimal division with connected pieces. Since there also always exists a proportional division (whose egalitarian social welfare is at least $\frac{1}{n}$), the egalitarian-optimal division must have an egalitarian social welfare of at least $\frac{1}{n}$ and thus a utilitarian social welfare of at least 1. Clearly, the maximum utilitarian social welfare attainable in any non-equitable division is less than n, and thus the utilitarian Price of Equitability is also less than n.

For the lower bound, fix some small $\epsilon > 0$ and consider n players with the following valuation functions. For $i = 1, \ldots, (n-1)$, player i assigns value of 1 to the interval $(\frac{i}{n} - \epsilon, \frac{i}{n} + \epsilon)$ and 0 to the rest of the cake. Finally, player n assigns uniform value to the entire cake.

Since any connected piece of (physical) size $\frac{1}{n} + 2\epsilon$ necessarily contains the entire desired piece of at least one player $i \in [n-1]$, the utility of player n in any equitable division must be strictly smaller than $\frac{1}{n} + 2\epsilon$; the utilitarian welfare of such a division is therefore smaller than $1 + 2n\epsilon$. Now, consider the following (non-equitable) division: give player 1 the interval $(0, \frac{1}{n} + \epsilon)$, players $i = 2, \ldots, (n-1)$ the interval $(\frac{i-1}{n} + \epsilon, \frac{i}{n} + \epsilon)$, and player n the interval $(\frac{n-1}{n} + \epsilon, 1)$. The utilitarian welfare of this division is $n - 1 + \frac{1}{n} - \epsilon$. By appropriately choosing ϵ, the Price of Equitability can be arbitrarily close to $n - 1 + \frac{1}{n}$. □

4 Trading Fairness for Efficiency

The work on the Price of Fairness is concerned with the trade-off between two goals of cake division: Fairness, and efficiency (in terms of social welfare). However, the results we presented so far, as well as the results in [CKKK09], concentrate on one direction of this trade-off, namely *how much efficiency may have to be sacrificed to achieve fairness.* We now turn to look at the analogue question of *how much fairness may have to be given up to achieve social optimality*; sadly, it seems that at least for the connected-pieces case, the results are somewhat pessimistic, except for equitability and proportionality with the egalitarian welfare.

In order to answer such questions, one first has to quantify unfairness. The following definitions seem natural:

We say that a division x:

- is *α-unproportional* if some player $i \in [n]$ has $u_i(x) \leq \frac{1}{\alpha \cdot n}$.

- *has envy of* α if there exist players $i, j \in [n]$ for which

$$v_i(x_{\pi(j)-1}, x_{\pi(j)}) \geq \alpha \cdot v_i(x_{\pi(i)-1}, x_{\pi(i)}) = \alpha \cdot u_i(x) \ ,$$

  i.e. if some i feels that $j \neq i$ received a piece worth α-times more than the one she got.

- is α*-inequitable* if there are players $i, j \in [n]$ with $u_i(x) \geq \alpha \cdot u_j(x)$.

Using these "unfairness" notions, we can obtain the following simple results:

Proposition 8. *There are cake-cutting instances where an utilitarian-optimal division is necessarily infinitely unfair, by all three measures above.*

Proof. Consider the cake cutting instance from the proof of Proposition 3. In this instance, the unique utilitarian-optimal division gives no cake at all to the "indifferent players"; it follows that this division is infinitely unproportional and inequitable, and has inifinite envy. ☐

We already know (Proposition 4 and Theorem 6) that egalitarian optimality is not in conflict with neither proportionality nor equitability. However, this is not the case for envy:

Proposition 9. *There are cake-cutting instances where an egalitarian-optimal division necessarily has envy arbitrarily close to* $n-1$*, and this is the maximum possible envy for such divisions.*

Proof. Let $\epsilon > 0$ be an arbitrarily small constant, and consider n players with the following valuation functions, which are fairly similar to those in the proof of Theorem 5. For $i = 1, \ldots, (n-1)$, player i assigns a value of $1 - \frac{1}{n} - \epsilon$ to the piece $(i - \frac{\epsilon}{2}, i + \frac{\epsilon}{2})$ (her "favorite piece"), a value of $\frac{1}{n} + \epsilon$ to the piece $(1 - \frac{2i+1}{2n} - \frac{\epsilon}{2}, 1 - \frac{2i+1}{2n} + \frac{\epsilon}{2})$ (her "second-favorite piece"), and value of 0 to the rest of the cake. Finally, player n assigns uniform value to the entire cake.

It is clear that there is no way for the egalitarian value to exceed $\frac{1}{n} + \epsilon$: In order for that to happen, player n must get a connected piece of physical size larger than $\frac{1}{n} + \epsilon$, which must contain the entire favorite piece of some player $i < n$, and so player i can get utility at most $\frac{1}{n} + \epsilon$. However, egalitarian welfare of $\frac{1}{n} + \epsilon$ can be easily achieved, and in such case player n indeed devours the entire favorite piece of some player $i < n$; this player receives a piece worth (in her eyes) only $\frac{1}{n} + \epsilon$ while she values the piece n receives as worth $1 - \frac{1}{n} - \epsilon$. The envy in every egalitarian-optimal division is therefore $\frac{n-1-\epsilon n}{1+\epsilon n}$, which can be arbitrarily close to $n - 1$ with an appropriate choice of ϵ.

Since the egalitarian-optimal division is always proportional, every player must get at least $\frac{1}{n}$ of the cake in it; therefore, in this player's view, another player may get at most $\frac{n-1}{n}$. It thus follows that in every such division the maximum possible envy is $n - 1$. ☐

5 Conclusions and Open Problems

In this work we analyzed the possible degradation in social welfare due to fairness requirements, when requiring that each player obtain a single connected piece. We obtain that the results vary considerably, depending on the fairness criteria used, and the social welfare function in consideration. The bounds range from provably no degradation for proportionality and equitability under the egalitarian welfare, through an $O(\sqrt{n})$ degradation for envy-freeness and proportionality under the utilitarian welfare, to an $O(n)$ degradation for equitability under the utilitarian welfare and for envy-freeness under the egalitarian welfare. We have also seen that if we seek to trade fairness to achieve social optimality, the "exchange rate" may (at the worst case) be infinite for utilitarian welfare (for all three fairness criteria), or linear for egalitarian welfare and envy-freeness.

Many open questions await further research, including:

- *Small number of connected pieces.* One motivation for considering cake cutting with connected pieces is the desire to avoid situations where a player receives "a pile of crumbs" for his fair share of the cake. On the other hand, requiring that each player receives a single connected interval may be too strict a requirement. A natural middle ground is to require that each player receives only a small number of pieces, e.g. a constant number. The question thus arises to bound the degradation to the social welfare under such requirements. In such an analysis it would be interesting to see how the bounds on degradation behave as a function of the number of permissible pieces.

- *The Egalitarian Price of Fairness with non-connected pieces.* [CKKK09] provide bounds on the Price of Fairness using the utilitarian welfare function, for the setting that non-connected pieces are permissible. Bounding the egalitarian Price of Fairness in this setting remains open. A trivial upper bound on the Price of Envy-freeness is $\frac{n}{2}$, and we have examples of instances where this price is strictly larger than 1, but obtaining tight bounds seems to require additional work and techniques.

- *The egalitarian Price of Proportionality and Price of Equitability for indivisible goods.* [CKKK09] provide analysis for the utilitarian Price of Fairness for such goods. A simple example can be constructed to show a tight bound of $\frac{n}{2}$ for the egalitarian Price of Envy-Freeness for this case. It thus remains open to determine the egalitarian Price of Proportionality and Equitability for such goods.

- *The Price of Fairness for connected chores.* As we already mentioned, fair division of chores has a somewhat different flavor from division of goods, and may require somewhat different techniques. One possible motivation for requiring connected division of chores may be, for example, a case in which a group of gardeners need to maintain a large garden, and so would like to give each of them one (connected) area to be responsible for.

Acknowledgement. We thank Ariel Procaccia for providing helpful comments on an earlier draft of this work.

References

[ADK+04] Elliot Anshelevich, Anirban Dasgupta, Jon M. Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS*, pages 295–304, 2004.

[Alo87] Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247 – 253, 1987.

[BT95] Steven J. Brams and Alan D. Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.

[BT96] Steven J. Brams and Alan D. Taylor. *Fair Division: From cake cutting to dispute resolution.* Cambridge University Press, New York, NY, USA, 1996.

[CKKK09] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. The efficiency of fair division. In *WINE*, pages 475–482, 2009.

[DS61] L. E. Dubins and E. H. Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1):1–17, Jan 1961.

[EP84] S. Even and A. Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285 – 296, 1984.

[EP06] Jeff Edmonds and Kirk Pruhs. Cake cutting really is not a piece of cake. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 271–278, New York, NY, USA, 2006. ACM.

[MIBK03] Malik Magdon-Ismail, Costas Busch, and Mukkai S. Krishnamoorthy. Cake-cutting is not a piece of cake. In *STACS*, pages 596–607, 2003.

[Mou04] Hervé J. Moulin. *Fair Division and Collective Welfare*. Number 0262633116 in MIT Press Books. The MIT Press, 2004.

[Pro09] Ariel D. Procaccia. Thou shalt covet thy neighbor's cake. In *IJCAI*, pages 239–244, 2009.

[RW98] Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can.* A K Peters, Ltd., Natick, MA, USA, 1998.

[Ste49] H. Steinhaus. Sur la division pragmatique. *Econometrica*, 17(Supplement: Report of the Washington Meeting):315–319, Jul 1949.

[Str80] Walter Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980.

[Str08] Walter Stromquist. Envy-free cake divisions cannot be found by finite protocols. *Electronic Journal of Combinatorics*, 15(1), Jan 2008.

[SW03] Jiri Sgall and Gerhard J. Woeginger. A lower bound for cake cutting. In *ESA*, pages 459–469, 2003.

Yonatan Aumann
Department of Computer Science
Bar-Ilan University
Ramat Gan, 52900, Israel
Email: `aumann@cs.biu.ac.il`

Yair Dombb
Department of Computer Science
Bar-Ilan University
Ramat Gan, 52900, Israel
Email: `yair.biu@gmail.com`

Truth, Justice, and Cake Cutting[1]

Yiling Chen, John K. Lai, David C. Parkes, and Ariel D. Procaccia

Superman: "I'm here to fight for truth, justice, and the American way."
Lois Lane: "You're gonna wind up fighting every elected official in this country!"

Superman (1978)

Abstract

Cake cutting is a common metaphor for the division of a heterogeneous divisible good. There are numerous papers that study the problem of fairly dividing a cake; a small number of them also take into account self-interested agents and consequent strategic issues, but these papers focus on fairness and consider a strikingly weak notion of truthfulness. In this paper we investigate the problem of cutting a cake in a way that is truthful and fair, where for the first time our notion of dominant strategy truthfulness is the ubiquitous one in social choice and computer science. We design both deterministic and randomized cake cutting algorithms that are truthful and fair under different assumptions with respect to the valuation functions of the agents.

1 Introduction

The need for resource allocation arises in many AI domains, and in particular in multiagent systems. This has led to a wide interest in the field known as *Multiagent Resource Allocation*, and to various applications of resource allocation techniques (see the survey by Chevalyere et al. [7]). Resource allocation problems deal with either *divisible* or *indivisible* resources, where the distinction is based on whether any fraction of a resource can be given to an agent.

Cutting a cake is often used as a metaphor for allocating a divisible good. The difficulty is not cutting the cake into pieces of equal size, but rather that the cake is not uniformly tasty: different agents prefer different parts of the cake, depending, e.g., on whether the toppings are strawberries or cookies. The goal is to divide the cake in a way that is "fair"; the definition of fairness is a nontrivial issue in itself, which we discuss in the sequel. The cake cutting problem dates back to the 1940s, and for over sixty years has attracted the attention of mathematicians, economists, and political scientists. While most of the work in artificial intelligence, and computer science in general, has focused on the allocation of indivisible resources, recent years have seen an increasing interest among computer scientists in the allocation of divisible resources (see, e.g, [9, 10, 15]).

Slightly more formally, the cake is represented by the interval $[0, 1]$. Each of n agents has a valuation function over the cake, which assigns a value to every given piece of cake and is additive. The goal is to find a partition of the cake among the agents (while possibly throwing a piece away) that satisfies one or several fairness criteria. In this paper we consider the two most prominent criteria. A *proportional* allocation is one where the value each agent has for its own piece of cake is at least $1/n$ of the value it assigns to the entire cake. An

[1]A version that is similar to this extended abstract will appear in the proceedings of AAAI'10. The full version of the paper, which includes all omitted proofs and a longer exposition, will shortly be available online. The paper was presented in the Harvard EconCS seminar (February 2010) and in a workshop on prior-free mechanism design in Guanajuato, Mexico (May 2010).

envy-free (EF) allocation is one where the value each agent assigns to its own piece of cake is at least as high as the value it assigns to any other agent's piece of cake. There is a rather large body of literature on fairly cutting a cake according to these two criteria (see, e.g., the books by Robertson and Webb [16] and Brams and Taylor [6]).

So far we have briefly discussed "justice", but have not yet mentioned "truth." Taking the game-theoretic point of view, an agent's valuation function is its private information, which is reported to a cake cutting algorithm. We would like an algorithm to be *truthful*, in the sense that agents are motivated to report their true valuation functions. Like fairness, this idea of truthfulness also lends itself to many interpretations. One variation, referred to as *strategy-proofness* in previous papers by Brams et al. [4, 5], assumes that an agent would report its truthful valuation rather than lie if there *exist* valuations of the other agents such that reporting truthfully yields at least as much value as lying. In the words of Brams et al., "...the players are risk-averse and never strategically announce false measures if it does not guarantee them more-valued pieces. ... Hence, a procedure is strategy-proof if no player has a strategy that dominates his true value function." [5, page 362].

The foregoing notion is strikingly weak compared to the notion of truthfulness that is common in the social choice literature. Indeed, strategy-proofness is usually taken to mean that an agent can *never* benefit by lying, that is, *for all* valuations of the other agents reporting truthfully yields at least as much value as lying. Put another way, truth-telling is a dominant strategy. This notion is worst-case, in the sense that an agent cannot benefit by lying even if it is fully knowledgeable of the valuations of the other agents. It is also the predominant one in the computer science literature, and in particular in the algorithmic mechanism design literature [14]. In order to prevent confusion we will avoid using the term "strategy-proof," and instead refer to the former notion of Brams et al. as "weak truthfulness" and to the latter standard notion as "truthfulness."

To illustrate the difference between the two notions, consider the most basic cake cutting algorithm for the case of two agents, the *Cut and Choose* algorithm.[2] Agent 1 cuts the cake into two pieces that are of equal value according to its valuation; agent 2 then chooses the piece that it prefers, giving the other piece to agent 1. This algorithm is trivially proportional and EF.[3] It is also weakly truthful, as if agent 1 divides the cake into two pieces that are unequal according to its valuation then agent 2 may prefer the piece that is worth more to agent 1. Agent 2 clearly cannot benefit by lying. However, the algorithm is not truthful. Indeed, consider the case where agent 1 would simply like to receive as much cake as possible, whereas the single-minded agent 2 is only interested in the interval $[0, \epsilon]$ where ϵ is small (for example, it may only be interested in the cherry). If agent 1 follows the protocol it would only receive half of the cake. Agent 1 can do better by reporting that it values the intervals $[0, \epsilon]$ and $[\epsilon, 1]$ equally, since then it would end up with almost the entire cake by choosing to cut pieces $[0, \epsilon], [\epsilon, 1]$.

In this paper we consider the design of truthful and fair cake cutting algorithms. To the best of our knowledge we are the first to do so. However, there is a major obstacle that must be circumvented: regardless of strategic issues, and when there are more than four agents, even finding a proportional and EF allocation in a bounded number of steps with a deterministic algorithm is a long-standing open problem! See [15] for an up-to-date discussion.[4] We shall therefore restrict ourselves to specific classes of valuation functions where efficiently finding fair allocations is a non-issue; the richness of our problem stems from our desire to additionally achieve truthfulness.

[2]This algorithm is described here with the agents taking actions; equivalently, the algorithm acts on behalf of agents using the reported valuations.

[3]Proportionality and envy-freeness coincide if there are two agents and the entire cake is allocated.

[4]To be precise, previous algorithmic work assumed that the entire cake has to be allocated, but this does not seem to be a significant restriction in the context of fairness.

Our results. We first consider deterministic algorithms. We restrict ourselves to the case where the agents hold *piecewise uniform* valuation functions, that is, each agent is interested in a collection of subintervals of $[0, 1]$ with the same marginal value for each fractional piece in each subinterval. This is the case when some parts of the cake satisfy a certain property and an agent desires as much of these parts as possible. Our main result is a deterministic algorithm for any number of agents that is truthful, proportional, EF, and polynomial-time. The proof requires many ingredients, including an application of the classic Max-Flow Min-Cut Theorem.

We next consider randomized algorithms. We slightly relax truthfulness by asking that the algorithm be *truthful in expectation*, that is, an agent cannot hope to increase its expected value by lying for any reports of other agents. For general valuations, we present a simple randomized algorithm that is truthful in expectation, and always outputs an allocation that is proportional and EF. We further establish that this algorithm is tractable under the relatively weak assumption that the agents hold *piecewise linear* valuation functions, that is where the marginal value in each subinterval of interest is a linear function.

Related work. We have recently learned of an independent working paper by Mossel and Tamuz that asks similar questions about truthful and fair cake cutting [13], but they focus on existence theorems. In particular, under general assumptions they show that there exists a mechanism that is truthful in expectation and guarantees each agent a value of more than $1/n$ in expectation. The results are then extended to the case of indivisible goods. The technical overlap between the two papers is very small; we refer the reader's attention to this overlap in a footnote in Section 4.

Thomson [17] showed that in general a truthful and Pareto-optimal algorithm must be dictatorial in the slightly different setting of pie-cutting. Note that Pareto-optimality is not a fairness property and neither implies, nor is implied by, envy-freeness or proportionality.

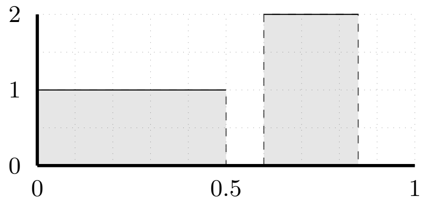
Our deterministic algorithm is related to a method proposed by Bogomolnaia and Moulin [3] in the context of the random assignment problem, and the network flow techniques we employ in our analysis generalize the reinterpretation of this method in terms of network flow due to Katta and Sethuraman [11]. We elaborate in Section 3.

2 Preliminaries

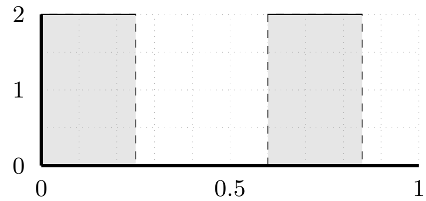
We consider a heterogeneous cake, represented by the interval $[0, 1]$. A *piece of cake* is a *finite* union of subintervals of $[0, 1]$. We sometimes abuse this terminology by treating a piece of cake as the set of the (inclusion-maximal) intervals that it contains. The length of the interval $I = [x, y]$, denoted $\text{len}(I)$, is $y - x$. For a piece of cake X we denote $\text{len}(X) = \sum_{I \in X} \text{len}(I)$.

The set of agents is denoted $N = \{1, \ldots, n\}$. Each agent $i \in N$ holds a private valuation function V_i, which maps given pieces of cake to the value agent i assigns them. Formally, each agent i has a *value density function*, $v_i : [0, 1] \to [0, \infty)$, that is piecewise continuous. The function v_i characterizes how agent i assigns value to different parts of the cake. The value of a piece of cake X to agent i is then defined as $V_i(X) = \int_X v_i(x)dx = \sum_{I \in X} \int_I v_i(x)dx$. We note that the valuation functions are *additive*, i.e. for any two disjoint pieces X and Y, $V_i(X \cup Y) = V_i(X) + V_i(Y)$, and *non-atomic*, that is $V_i([x, x]) = 0$ for every $x \in [0, 1]$. The last property implies that we do not have to worry about the boundaries of intervals, i.e., open and closed intervals are identical for our purposes. We further assume that the valuation functions are *normalized*, i.e. $V_i([0, 1]) = \int_0^1 v_i(x)dx = 1$.

A cake cutting algorithm is a function f from the valuation function of each agent to an allocation $(A_1, \ldots, A_n)$ of the cake such that the pieces are pairwise disjoint. For each $i \in N$ the piece A_i is allocated to agent i, and the rest of the cake, i.e., $[0, 1] \setminus \bigcup_{i \in N} A_i$, is thrown away. Here we are assuming *free disposal*, that is, the algorithm can throw away



(a) Value density function for a piecewise constant valuation that is not piecewise uniform.



(b) Value density function for a piecewise uniform valuation.

Figure 1: An illustration of special value density functions.

resources without incurring a cost.

We say that an allocation $A_1, \ldots, A_n$ is *proportional* if for every $i \in N$, $V_i(A_i) \geq 1/n$, that is, each agent receives at least a $(1/n)$-fraction of the cake according to its own valuation. We say that an allocation is *envy-free (EF)* if for every $i, j \in N$, $V_i(A_i) \geq V_i(A_j)$, i.e., each agent prefers its own piece of cake to the piece of cake allocated to any other agent. A proportional (resp., EF) cake cutting algorithm always returns a proportional (resp., EF) allocation.

Note that when $n = 2$ proportionality implies envy-freeness. Indeed, $V_i(A_i)+V_i(A_{3-i}) \leq 1$, and hence if $V_i(A_i) \geq 1/2$ then $V_i(A_{3-i}) \leq 1/2$. Under the free disposal assumption the converse is not true. For example, an allocation that throws away the entire cake is EF but not proportional. In general, when $n > 2$ proportionality neither implies nor is implied by envy-freeness.[5]

A cake cutting algorithm f is *truthful* if when an agent lies it is allocated a piece of cake that is worth, according to its real valuation, no more than the piece of cake it was allocated when reporting truthfully. Formally, denote $A_i = f_i(V_1, \ldots, V_n)$, and let $\mathcal{V}$ be a class of valuation functions. The algorithm f is truthful if for every agent i, every collection of valuations functions $V_1, \ldots, V_n \in \mathcal{V}$, and every $V_i' \in \mathcal{V}$, it holds that $V_i(f_i(V_1, \ldots, V_n)) \geq V_i(f_i(V_1, \ldots, V_{i-1}, V_i', V_{i+1}, \ldots, V_n))$.

3 Deterministic Algorithms and Piecewise Uniform Valuations

As noted in the introduction, in general there are no known bounded deterministic proportional and EF cake cutting algorithms for more than four agents, even if one is not concerned about strategic issues. Therefore, in this section we restrict ourselves to a specific class of valuation functions.

We say that a valuation function V_i is *piecewise constant* if and only if its corresponding value density function v_i is piecewise constant, that is $[0, 1]$ can be partitioned into a finite number of intervals such that v_i is constant on each interval (see Figure 1(a)). We say that V_i is *piecewise uniform* if moreover v_i is either some constant $c \in \mathbb{R}_+$ (the same one across intervals) or zero. See Figure 1(b) for an illustration.

Piecewise uniform valuation functions imply that agent $i \in N$ is uniformly interested in a finite union of intervals, which we call its *reference piece of cake* and denote by U_i. For example, in Figure 1(b), $U_i = [0, 0.25] \cup [0.6, 0.85]$. Given a piece of cake X, it holds that $V_i(X) = \text{len}(X \cap U_i)/\text{len}(U_i)$. From the computational perspective, the size of the input to the cake cutting algorithm is the number of bits that define the boundaries of the intervals in the agents' reference pieces of cake.

[5]If free disposal is not assumed, that is, the entire cake is allocated, then envy-freeness implies proportionality for any n.

In the rest of this section we assume that the valuation functions are piecewise uniform. We believe that piecewise uniform valuations are very natural. An agent would have such a valuation function if it is simply interested in pieces of the good that have a certain property, e.g., a child only likes portions of the cake that have chocolate toppings, and wants as much cake with chocolate toppings as possible. We consider more general valuations in the next section on randomized algorithms.

3.1 A deterministic algorithm

Before introducing our algorithm we present some required notation. Let $S \subseteq N$ be a subset of agents and let X be a piece of cake. Let $D(S, X)$ denote the portions of X that are valued by at least one agent in S. Formally, $D(S, X) = \left(\bigcup_{i \in S} U_i\right) \cap X$, and is itself a union of intervals.

Let $\text{avg}(S, X) = \text{len}(D(S, X))/|S|$ denote the average length of intervals in X desired by at least one agent in S. We say that an allocation is *exact* with respect to S and X if it allocates to each agent in S a piece of cake of length $\text{avg}(S, X)$ comprised *only* of desired intervals. Clearly this requires allocating all of $D(S, X)$ since the total length of allocated intervals is $\text{avg}(S, X) \cdot |S| = \text{len}(D(S, X))$. Suppose $S = \{1, 2\}$ and $X = [0, 1]$: if $U_1 = U_2 = [0, 0.2]$ then agents 1 and 2 receiving $[0, 0.1]$ and $[0.1, 0.2]$ respectively is an exact allocation; but if $U_1 = [0, 0.2], U_2 = [0.3, 0.7]$ then there is no exact allocation.

The deterministic algorithm for n agents with piecewise uniform valuations is a recursive algorithm that finds a subset of agents with a certain property, makes the allocation decision for that subset, and then makes a recursive call on the remaining agents and the remaining intervals. Specifically, for a given set of agents $S \subseteq N$ and a remaining piece of cake to be allocated X, we find the subset $S' \subseteq S$ of agents with the smallest $\text{avg}(S', X)$. We then give an exact allocation of $D(S', X)$ to S'. We recurse on $S \setminus S'$ and the intervals not desired by any agent in S', i.e. $X \setminus D(S', X)$. The pseudocode of the algorithm is given as Algorithm 1.

Algorithm 1 $(V_1, \ldots, V_n)$

---

1. SubRoutine($\{1, \ldots, n\}, [0, 1], (V_1, \ldots, V_n)$)

SubRoutine(S, X, $V_1, \ldots, V_n$):

1. If $S = \emptyset$, return.
2. Let $S_{\min} \in \underset{S' \subseteq S}{\text{argmin}}\ \text{avg}(S', X)$ (breaking ties arbitrarily).
3. Let $E_1, \ldots, E_n$ be an exact allocation with respect to $S_{\min}, X$ (breaking ties arbitrarily). For each $i \in S_{\min}$, set $A_i = E_i$.
4. Subroutine($S \setminus S_{\min}, X \setminus D(S_{\min}, X), (V_1, \ldots, V_n)$).

In particular, Steps 2 and 3 of SubRoutine imply that if $S = \{i\}$ then $A_i = D(S, X)$. For example, suppose $X = [0, 1]$, $U_1 = [0, 0.1]$, $U_2 = [0, 0.39]$, and $U_3 = [0, 0.6]$. In this case, the subset with the smallest average is $\{1\}$, so agent 1 receives all of $[0, 0.1]$ and we recurse on $\{2, 3\}, [0.1, 1]$. In the recursive call, set $\{2\}$ has average 0.39 - 0.1 = 0.29, set $\{3\}$ has average 0.6 - 0.1 = 0.5, and set $\{2, 3\}$ has average $(0.6 - 0.1)/2 = 0.25$. As a result, the entire set $\{2, 3\}$ is chosen as the set with smallest average, and an exact allocation of $[0.1, 1.0]$ is given to agents 2 and 3. One possible allocation is to give agent 2 $[0.1, 0.35]$ and agent 3 $[0.35, 0.6]$. Note that, if agent 1 uniformly values $[0, 0.2]$ instead, the first call would choose $\{1, 2\}$ as the subset with the smallest average, equally allocating $[0, 0.39]$ between agents 1 and 2 and giving the rest, $[0.39, 0.6]$, to agent 3.

An analysis of the two agent algorithm. To gain intuition, consider the case of two

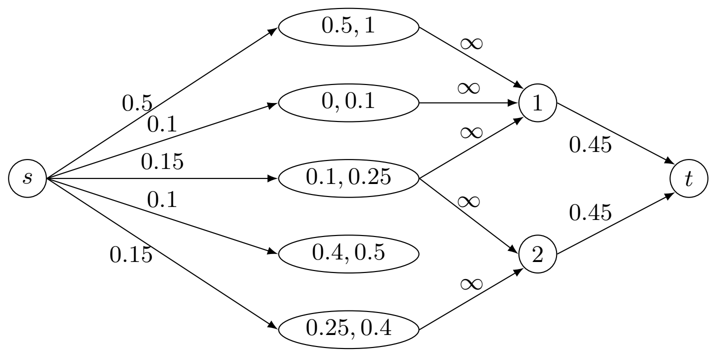


Figure 2: The flow network induced by the example.

agents; designing truthful, proportional and EF algorithms even for this case is nontrivial. Assume that $\text{len}(U_1) \leq \text{len}(U_2)$ for ease of presentation. If in addition, $\text{len}(U_1) > \text{len}(U_1 \cup U_2)/2$ then set $\{1, 2\}$ has the smallest average and we divide $U_1 \cup U_2$ exactly, with each agent getting all of $U_i \setminus U_{3-i}$ and sharing $U_1 \cap U_2$ in a way that $\text{len}(A_1) = \text{len}(A_2)$. Otherwise, agent 1 gets all of U_1 and agent 2 gets $U_2 \setminus U_1$. The algorithm tries to give both agents the same length, with each agent always getting at least half of its desired intervals, leading to proportionality and EF because of piecewise uniform valuations. For sufficient overlap in desired intervals, each receives exactly half of $U_1 \cup U_2$. For totally disjoint reference pieces, each receives just its reference piece. We defer a discussion of truthfulness to the general algorithm; the crux here is to note that each agent i receives all of $U_i \setminus U_{3-i}$, and the algorithm precludes overclaims through providing a nonincreasing share of $U_i \cap U_{3-i}$ as $\text{len}(U_i)$ increases.

Exact Allocations and Maximum Flows. Before turning to properties of truthfulness and fairness, we point out that so far it is unclear whether Algorithm 1 is well-defined. In particular, the algorithm requires an exact allocation E with respect to the subset $S_{\min}$ and X, but it remains to show that such an allocation exists, and to provide a way to compute it. To this end we exploit a close relationship between exact allocations and maximum flows in networks.

For a given set of agents $S \subseteq N$ and a piece of cake to be allocated X, define a graph $G(S, X)$ as follows. We keep track of a set of marks, which will be used to generate nodes in $G(S, X)$. First mark the left and right boundaries of all intervals that are contained in X. For each agent $i \in N$ and subinterval in U_i, mark the left and right boundaries of subintervals that are contained in $U_i \cap X$. When we have finished this process, each pair of consecutive markings will form an interval such that each agent will either uniformly value the entire interval or value none of the interval. In $G(S, X)$, create a node for each interval I formed by consecutive markings, and add a node for each agent $i \in N$, a source node s, and a sink node t. For each interval I, add a directed edge from source s to I with capacity equal to the length of the interval. Each agent node is connected to t by an edge with capacity $\text{avg}(S, X)$. For each interval-agent pair (I, i), add a directed edge with infinite capacity from node I to the agent i if agent i desires interval I.

For example, suppose $U_1 = [0, 0.25] \cup [0.5, 1]$ and $U_2 = [0.1, 0.4]$. If $X = [0, 1]$ then the interval markings will be $\{0, 0.1, 0.25, 0.4, 0.5, 1\}$. Agent 1 values $[0, 0.1]$, both agents value $[0.1, 0.25]$, agent 2 values $[0.25, 0.4]$, neither agent values $[0.4, 0.5]$ and agent 1 values $[0.5, 1]$. It holds that $\text{len}(D(\{1, 2\}, [0, 1])) = 0.9$. Average values are 0.75, 0.3 and 0.45 for sets $\{1\}, \{2\}$ and $\{1, 2\}$ respectively. See Figure 2 for an illustration of the induced flow network.

Lemma 1. *Let $S \subseteq N$, and let X be a piece of cake. There is a flow of size len$(D(S, X))$ in $G(S, X)$ if and only if for all $S' \subseteq S$, avg$(S', X) \geq$ avg(S, X).*

Below we only prove the "if" direction, which is the one we need, using an application of the classic Max-Flow Min-Cut Theorem (see, e.g., [8]).

Proof of "if". Assume that for all $S' \subseteq S$, $\text{avg}(S', X) \geq \text{avg}(S, X)$. By the Max-Flow Min-Cut Theorem, the minimum capacity removed from a graph in order to disconnect the source and sink is equal to the size of the maximum flow. The only edges with finite capacity in $G(S, X)$ are the ones that connect agent nodes to the sink, and the ones that connect the source to the interval nodes.

Construct a candidate minimum cut by disconnecting some set of agent nodes $T \subseteq S$ from the sink at cost $|T| \cdot \text{avg}(S, X)$ and then disconnecting all the (s, I) connections to interval nodes I desired by an agent $i \in S \setminus T$. This means that the total additional capacity we need to remove is $\text{len}(D(S \setminus T, X))$, the total length of intervals desired by at least one agent in $S \setminus T$. By assumption, this is at least $|S \setminus T| \cdot \text{avg}(S, X)$. As a result, this cut has capacity of at least $|T| \cdot \text{avg}(S, X) + |S \setminus T| \cdot \text{avg}(S, X) = |S| \cdot \text{avg}(S, X) = \text{len}(D(S, X))$. $\square$

The following lemma establishes that this flow of size $\text{len}(D(S, X))$ in $G(S, X)$ is, in particular, characterizing an exact allocation. We omit the proof, which follows from the construction of the network.

Lemma 2. *Let $S \subseteq N$, and let X be a piece of cake. There exists an exact allocation with respect to S, X if and only if there exists a maximum flow of size len$(D(S, X))$ in $G(S, X)$.*

By combining Lemma 1 and Lemma 2 we see that the algorithm is indeed well-defined: if S has the smallest average then there exists an exact allocation with respect to S, X.[6] Moreover, we obtain a tractable algorithm for computing an exact allocation, by computing the maximum flow and deriving an exact allocation. A maximum flow can be computed in time that is polynomial in the number of nodes, that is, polynomial in our input size (see, e.g., [8]). We remark without proof that it is also possible to implement Step 2 of SubRoutine in polynomial time, using similar (but slightly more involved) network flow arguments. Therefore, Algorithm 1 can be implemented in polynomial time.

Truthfulness and fairness. Our main tool in proving that Algorithm 1 is truthful, proportional and EF is the following lemma (we omit its proof).

Lemma 3. *Let $S_1, \ldots, S_m$ be the ordered sequence of agent sets with the smallest average as chosen by Algorithm 1 and $X_1, \ldots, X_m$ be the ordered sequence of pieces to be allocated in calls to* SubRoutine. *That is, $X_1 = [0, 1], X_2 = X_1 \setminus D(S_1, X_1), \ldots, X_m = X_{m-1} \setminus D(S_{m-1}, X_{m-1})$. Then for all $i > j$, avg$(S_i, X_i) \geq$ avg(S_j, X_j), and agents that are members of later sets receive weakly more in desired lengths.*

Envy-freeness now follows immediately from the lemma. Indeed, consider an agent $i \in N$. By "chosen" we mean that the agent was part of the subset with smallest average. The agent does not envy agents chosen in the same call to SubRoutine since all agents receive the same length in desired intervals and their valuations are piecewise uniform. By Lemma 3, the agent does not envy agents chosen in earlier calls because the amount agents receive weakly increases with each call. The agent does not envy agents chosen in later calls because all intervals desired by the agent are removed from consideration when the agent receives its allocation.

[6] Note that the network in Figure 2 does not satisfy the average minimality requirement and does not provide a corresponding exact allocation.

We provide a sketch of truthfulness, which follows by showing that an agent $i \in N$ has no incentive to change the choice of $S_{\min}$ and cannot profitably manipulate the exact allocation for a given $S_{\min}$.

1. Manipulations that change $S_{\min}$. Consider two subcases.

   (a) When i reports truthfully, $S_{\min} = S', i \notin S'$. An agent cannot affect $\text{avg}(T, X)$ if $i \notin T$, so the agent cannot cause some other $S'', i \notin S''$ to be chosen. The agent can cause $S'', i \in S''$, to be chosen, but then $\text{avg}(S'', X) \leq \text{avg}(S', X)$ and it follows from Lemma 3 that the agent does not gain.

   (b) When i reports truthfully, $S_{\min} = S', i \in S'$. Assume without loss of generality that $|S| \geq 2$. In this case, all agents in S', including i, receive exactly $\text{avg}(S', X) = k$ in intervals. Agent i can cause selection of some S'' by misstating its valuation. If $i \in S''$, then $\text{avg}(S'', X) \geq k$ for this to be profitable. If $i \notin S''$, then S'' was not chosen when i reports truthfully, so $\text{avg}(S'', X) \geq k$. In either case, agents $j \in S' \setminus \{i\}$ previously received k, but now receive at least k by observing that $\text{avg}(S'', X) \geq k$ and applying Lemma 3. Agent i receives at most $\text{len}(D(S', X))$ minus the intervals received by agents $j \in S' \setminus \{i\}$.[7] These agents receive weakly more if i manipulates, and thus, manipulations are not profitable.

2. Manipulations that change the exact allocation for a given $S_{\min}, i \in S_{\min}$. By definition each agent in $S_{\min}$ receives exactly $\text{avg}(S_{\min}, X)$ in desired intervals. If agent i decreases this value, it receives strictly less. If agent i increases this value by lying, then other agents receive more of the actual $D(S_{\min}, X)$, leaving less for agent i.

We omit the proof of proportionality, but it follows after establishing that no desired pieces are thrown away. Overall, we have the following theorem.

Theorem 4. *Assume that the agents have piecewise uniform valuation functions. Then Algorithm 1 is truthful, proportional, EF, and polynomial-time.*

Relation to work on the random assignment problem. Consider a setting where indivisible items must be assigned to agents. In the *random assignment problem* items can be assigned to agents randomly, i.e., a random assignment is a probability distribution over deterministic assignments. A random assignment that gives an item to an agent with probability p can be interpreted as assigning a p-fraction of the item to the agent. Crucially, in the papers discussed below the assumption is that each agent is only interested in receiving one item.

Bogomolnaia and Moulin [3] consider the random assignment problem when the agents have dichotomous preferences over the items, in the sense that for each agent the set of items can be partitioned into acceptable and unacceptable items (where all acceptable items have value 1 and unacceptable items have value 0). They provide a random assignment method called the *egalitarian assignment solution* and show that it is truthful, EF, and satisfies other highly desirable properties.

Interestingly, the cake cutting problem under piecewise uniform valuation functions is similar to a random assignment problem, as one can mark the beginning and end of each agent's desired intervals and treat the subintervals between consecutive marks as items. However, there are two fundamental differences between our setting and [3]. First, in our setting agents are interested in receiving as much of their desired "items" as possible (rather than just one item). Second, in our setting dichotomous preferences would mean that agents

[7] Lemma 3 also applies to agent i, but since it lies, it may receive intervals that are not desired and outside of $D(S', X)$.

value all desired subintervals equally, which is clearly not the case since these subintervals have different lengths.[8] Nevertheless, it turns out that the egalitarian assignment solution is very similar to the special case of Algorithm 1 under this strong assumption. Katta and Sethuraman [11] observe that the egalitarian assignment solution can be computed in polynomial time using network flow techniques, so our arguments above are an independent generalization of this observation. Interestingly, it is noted in [11] that the egalitarian assignment solution is identical to another independent algorithm for finding a lexicographically optimal flow in a network due to Megiddo [12].

In earlier work Bogomolnaia and Moulin [2] study random assignments under *strict* ordinal preferences, and propose a solution that satisfies a weaker notion of truthfulness (which does not imply truthfulness in our setting) as well as envy-freeness and other properties. In terms of the agents' preferences this setting is incomparable to ours since agents may be indifferent between subintervals. However, in our setting agents cannot hold arbitrary ordinal preference profiles over subintervals between consecutive marks, since if two agents desire two subintervals, both agents would value the longer subinterval more than the shorter.

The results of [2] were extended by Katta and Sethuraman [11] to the case where agents are allowed to be indifferent between items. While the assumptions of [11] regarding preferences are weaker than ours, they establish that in this more general setting even Bogomolnaia and Moulin's weaker notion of truthfulness is in fact incompatible with envy-freeness and an additional efficiency requirement; the algorithm that they propose satisfies the last two properties and hence is not (even weakly) truthful.

4 Randomized Algorithms and Piecewise Linear Valuations

In the previous section we saw that designing deterministic truthful and fair algorithms is not an easy task, even if the valuation functions of the agents are rather restricted. In this section we shall demonstrate that by allowing randomness we can obtain significantly more general results.

A *randomized cake cutting algorithm* outputs a random allocation given the reported valuation functions of the agents. There are very few previous papers regarding randomized algorithms for cake cutting. A rare example is the paper by Edmonds and Pruhs [9], where they give a randomized algorithm that achieves approximate proportionality with high probability. We are looking for a more stringent notion of fairness. We say that a randomized algorithm is *universally proportional* (resp., *universally EF*) if it always returns an allocation that is proportional (resp., EF).

One could also ask for *universal truthfulness*, that is, require than an agent may never benefit from lying, regardless of the randomness of the algorithm. A universally truthful algorithm is simply a probability distribution over deterministic truthful algorithms. However, asking for both universal fairness and universal truthfulness would not allow us to enjoy the additional flexibility that randomization provides. Therefore, we slightly relax our truthfulness requirement. Informally, we say that a randomized algorithm is *truthful in expectation* if, for all possible valuation functions of the other agents, the expected value an agent receives for its allocation cannot increase by lying, where the expectation is taken over the randomness of the algorithm.

We remark that while truthfulness in expectation seems natural, fairness (i.e., proportionality and envy-freeness) is something that we would like to hold *ex-post*; fairness is a

[8] In general no discretization of the cake would necessarily yield subintervals of equal length that correspond to dichotomous preferences. If we assume that desired intervals have rational endpoints then such a discretization can be found, but the number of subintervals would be exponentially large, leading to computational intractability.

property of the specific allocation that is being made, and continues to be relevant after the algorithm has terminated. Interestingly enough, if we were to turn this around, then achieving universal truthfulness and envy-freeness/proportionality in expectation is trivial: simply allocate the entire cake to a uniformly random agent!

4.1 A randomized algorithm

In order to design a randomized algorithm that is truthful in expectation, universally proportional, and universally EF, we consider a very special type of allocation. In the following we will not require the free disposal assumption, that is, we will consider partitions $X_1, \ldots, X_n$ of the cake such that $\bigcup_i X_i = [0, 1]$. We say that a partition $X_1, \ldots, X_n$ is *perfect* if for all $i, j \in N$, $v_i(X_j) = 1/n$. Consider the following randomized algorithm.

Algorithm 2 $(V_1, \ldots, V_n)$

1. Find a perfect partition $X_1, \ldots, X_n$.
2. Draw a random permutation π over N.
3. For each $i \in N$, set $A_i = X_{\pi(i)}$.

Lemma 5. *Algorithm 2 is truthful in expectation, universally proportional, and universally EF.*[9]

Proof. The fact that the algorithm is universally proportional and universally EF follows from the definition of perfect partitions: every agent has value $1/n$ for every piece!

We turn to truthfulness in expectation. The value an agent $i \in N$ obtains by reporting truthfully is exactly $1/n$. If agent i lies then the algorithm may choose a different partition $X'_1, \ldots, X'_n$. However, for any partition $X'_1, \ldots, X'_n$ the expected value of agent i when given a random piece is

$$\sum_{j \in N} \frac{1}{n} \cdot V_i(X'_j) = \frac{1}{n} \left(\sum_{j \in N} V_i(X'_j) \right) = \frac{1}{n},$$

where the second equality follows from the fact that the valuation functions are additive. □

Finding perfect partitions. Lemma 5 holds much promise, in that it is valid for all valuation functions. But there still remains the obstacle of actually finding a perfect partition given the valuation functions of the agents. Does such a partition exist, and can it be computed? More than two decades ago, Noga Alon [1] proved that if the valuation functions of the agents are defined by the integral of a continuous probability measure then there *exists* a perfect partition; this is a generalization of his famous theorem on necklace splitting. Unfortunately, Alon's elegant proof is nonconstructive (which is unusual for a proof in combinatorics), and to this day there is no known constructive method under general assumptions on the valuation functions. This is not surprising since a perfect partition induces an EF allocation, and finding an EF allocation in a bounded number of steps for more than four agents is an open problem.

To obtain a computational method, we consider valuation functions that are *piecewise linear*. A valuation function V_i is considered piecewise linear if and only if its corresponding value density function v_i is piecewise linear on $[0, 1]$. Piecewise linear valuation functions are significantly more general than the class of piecewise constant valuation functions. A piecewise linear valuation function can be concisely represented by the intervals on which v_i is linear, and for each interval the two parameters of the linear function. The following

[9]Mossel and Tamuz [13] make the same observation.

lemma provides us with a tractable method of finding a perfect partition when the agents have piecewise linear valuation functions.

Lemma 6. *Assume that the agents have piecewise linear valuation functions. Consider the following procedure. We make a mark at 0 and 1, and for each agent $i \in N$ make a mark at the left and right boundaries of each interval where v_i is linear. Next, we divide each interval I_j between two consecutive marks into $2n$ consecutive and connected subintervals $I_j^1, \ldots, I_j^{2n}$ of equal length. For each such I_j and every $i \in N$ add the subintervals I_j^i and I_j^{2n-i+1} to X_i. Then the overall partition is perfect.*

The lemma's proof is omitted. By combining Lemma 6 with Lemma 5 we obtain the following result.

Theorem 7. *Assume that the agents have piecewise linear valuation functions. Then there exists a randomized algorithm that is truthful in expectation, universally proportional, universally EF, and polynomial-time.*

5 Discussion

We have made progress on truthful and fair algorithms for cake cutting. In unpublished work, we show the nonexistence of simpler methods that make only contiguous allocations (and look closer to generalizations of the classic cut-and-choose algorithm) even for two agents both of whom are uniformly interested in a single (but different) subinterval. In future work we would like to generalize the deterministic algorithm to piecewise constant valuations and drop the free-disposal assumption. For practical settings, allowing more expressiveness (e.g., piecewise linear but a requirement that intervals are above some threshold length) seems important.

Acknowledgments

We thank Noga Alon for his wise advice regarding the tractable implementation of Step 2 of SubRoutine. We also thank Anna Bogomolnaia and Hervé Moulin for pointing our attention to the connection between our problem and the random assignment problem.

References

[1] N. Alon. Splitting necklaces. *Advances in Mathematics*, 63:241–253, 1987.

[2] A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100:295–328, 2001.

[3] A. Bogomolnaia and H. Moulin. Random matching under dichotomous preferences. *Econometrica*, 72:257–279, 2004.

[4] S. J. Brams, M. A. Jones, and C. Klamler. Better ways to cut a cake. *Notices of the AMS*, 53(11):1314–1321, 2006.

[5] S. J. Brams, M. A. Jones, and C. Klamler. Proportional pie-cutting. *Int. Journal of Game Theory*, 36(3–4):353–367, 2008.

[6] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press, 1996.

[7] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, 2nd edition, 2001.

[9] J. Edmonds and K. Pruhs. Balanced allocations of cake. In *Proc. of 47th FOCS*, pages 623–634, 2006.

[10] J. Edmonds and K. Pruhs. Cake cutting really is not a piece of cake. In *Proc. of 17th SODA*, pages 271–278, 2006.

[11] A. Katta and J. Sethuraman. A solution to the random assignment problem on the full preference domain. *Journal of Economic Theory*, 131:231–250, 2006.

[12] N. Megiddo. A good algorithm for lexicographically optimal flows in multi-terminal networks. *Bulletin of the American Mathematical Society*, 83:407–409, 1979.

[13] E. Mossel and O. Tamuz. Truthful fair division. Manuscript, 2010.

[14] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1–2):166–196, 2001.

[15] A. D. Procaccia. Thou shalt covet thy neighbor's cake. In *Proc. of 21st IJCAI*, pages 239–244, 2009.

[16] J. M. Robertson and W. A. Webb. *Cake Cutting Algorithms: Be Fair If You Can.* A. K. Peters, 1998.

[17] W. Thomson. Children crying at birthday parties. Why? *Journal of Economic Theory*, 31:501–521, 2007.

Yiling Chen
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA
Email: `yiling@eecs.harvard.edu`

John K. Lai
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA
Email: `jklai@post.harvard.edu`

David C. Parkes
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA
Email: `parkes@eecs.harvard.edu`

Ariel D. Procaccia
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA
Email: `arielpro@seas.harvard.edu`

Online Cake Cutting

Toby Walsh

Abstract

We propose an online form of the cake cutting problem. This models situations where players arrive and depart during the process of dividing a resource. We show that well known fair division procedures like cut-and-choose and the Dubins-Spanier moving knife procedure can be adapted to apply to such online problems. We propose some desirable properties that online cake cutting procedures might possess like online forms of proportionality and envy-freeness, and identify which properties are in fact possessed by the different online cake procedures.

1 Introduction

Congratulations. Today is your birthday so you take a cake into the office to share with your colleagues. At tea time, people slowly start to arrive. However, as some people have to leave early, you cannot wait for everyone to arrive before you start sharing the cake. How do you proceed fairly?

This is an example of what we call an *online* cake cutting problem. Most previous studies of cake cutting procedures have assumed that all the players are available at the time of the division. Here, players arrive and depart (either with their cake or perhaps after they have eaten their cake) as the cake is being divided. Such online problems occur in the real world as in our birthday example, but also on the internet where agents are often connecting asynchronously.

Online cake cutting poses some new challenges. On the one hand, the online aspect of such problems makes fair division more difficult than in the offline case. How can we ensure that a player does not envy another player when we may have to distribute cake to the second player before the first player is present (and we can hope to determine information about their valuation function)? On the other hand, the online aspect of such problems may make fair division easier than in the offline case. If players don't see cake that has already been distributed before they arrive, perhaps they do not envy it?

2 Online cake cutting

As is common in the literature [2], we will often assume that each player is risk averse so they maximize the minimum value of the cake that they will receive, regardless of what the other players do. A risk averse player will not choose a strategy that could yield more value if it also entails the possibility of getting less value. We will also usually assume that each player is ignorant of the value functions of the other players. We disuss relaxing these assumptions in the conclusions.

We formulate cake cutting as dividing the unit interval between the different players, where each player has a (typically additive and continuous) valuation function on the intervals that they are allocated. We do not suppose that players assigns the same value to the whole cake. Although we can normalize the valuation functions, we shall see that is not necessary as all the cake cutting procedures are *scale invariant*. Depending on the application, we may demand that players receive a continuous slice of cake or some union of slices.

In an online cake cutting problem, the players arrive in some given order. Players are allocated their cake and then depart. The order in which players depart can be fixed or can be change according to how they value the cake. For example, the player present who most values a cut slice of cake might be the next to depart. Alternatively, the player to depart might always be the player who arrived the longest time ago. We will assume that at least one player departs before the last player arrives (otherwise we can formulate this as an offline cake cutting procedure). To prevent trivial allocations, we also assume that at least one player receives some cake. However, we do not assume that all players receive cake or that all the cake is allocated. Formally an online cake cutting problem is defined by a procedure which given the valuation functions of the players who are present in the room and the number of players who will take part in total either allocates some cake to one of the present players (who then departs) or indicates that we wait until the next player arrives. This can model both a fixed arrival and departure order, as well as one in which the order depends on the valuation functions.

An important dimension of online cake cutting is what is known and by whom about the total number of players. For example, the total number of players might be known by all players. On the other hand, the players might only know a bound on the total number of players (e.g. you've invited 20 work colleagues to share your birthday cake but not all of them might turn up). However, there are several other possibilities (e.g. certain players might have complete certainty about n whilst others have complete uncertainty). In addition, an interesting generalization is when cake is being allocated before the total number of players is fixed.

3 Desirable properties

What properties do we want from an online cake cutting procedure? The literature on cake cutting studies various notions of fainerness like proportionality and envy freeness, as well as various forms of strategy proofness. The generalization of cake cutting to an online setting gives rise to some natural extensions of these notions.

3.1 Proportionality

A cake cutting procedure is *proportional* iff each of the n players assigns at least $\frac{1}{n}$ of the total value to their piece(s). Unfortunately, as we shall show, online cake cutting procedures cannot always be proportional. Suppose you only like icing. The problem is that you may not be able to prevent all the cake that is iced being distributed before you enter the room. We therefore consider weaker forms of proporitionality that are achievable. One more limited form of proportionality is that any player receives a fair proportion of the cake that remains when they arrive. A cake cutting procedure is *forward proportional* iff each player assigns at least $\frac{r}{n-k}$ of the total value of the cake to their pieces where r is the fraction of the total value assigned by the player to the (remaining) cake when they arrive and k is the number of players who have already left at this point.

3.2 Envy freeness

A stronger notion of fairness is envy freeness. A cake cutting procedure is *envy free* iff no player values another player's pieces more than their own. Note that envy freeness implies proportionality but not vice versa. With online cake cutting, envy freeness is also impossible to achieve in general. We therefore consider weaker forms of envy freeness that are achievable. A cake cutting procedure is *forward envy free* iff no player values the pieces of cake allocated to other players after their arrival more than their own. Players can, however, value the cake allocated to players who have already departed more than our own cake. This models situations where, for instance, we do not envy cake we don't see being allocated, or players eat their cake before departing and we do not envy cake that has already been eaten. Note that forward envy freeness implies forward proportionality but not vice versa. Similarly, envy freeness implies forward envy freeness but not vice versa. An even weaker form of envy freeness is when a player does not envy cake that is allocated to other players whilst they are in the room. A cake cutting procedure is *immediately envy free* iff no player values the pieces of cake allocated to another player after their arrival and before their departure more than their own. Note that forward envy freeness implies immediate envy freeness but not vice versa.

3.3 Equitability

Another fairness property is equitability. A cake cutting procedure is *equitable* iff all players assign the same value to the pieces of cake to which they are allocated (and so no player envies anothers valuation). For 3 or more player, equitability and envy freeness can be incompatible [2]. Equitability is a difficult property to achieve, even more so in our online setting. Unlike proportionality or envy freeness, there seems little merit in considering weaker forms of equitability. Either all players assign the same value to their allocated cake or they do not. There is no advantage to ignoring the value of the cake allocated to players who have already departed.

3.4 Efficiency

Another important notion is efficiency. Efficiency is also called Pareto optimality. A cake cutting procedure is *Pareto optimal* iff there is no other allocation to the one returned that is more valuable for one player and at least as valuable for the others. Note that Pareto optimality does not in itself ensure fairness since allocating all the cake to one player is Pareto optimal. A cake cutting procedure is *weakly Pareto optimal* iff there is no other allocation to the one returned that is more valuable for all players. A cake cutting procedure that is Pareto optimal is weakly Pareto optimal but not vice versa.

3.5 Strategy proofness

Another consideration is whether players have an incentive to act truthfully. A cake cutting procedure is *weakly truthful* iff a player will do at least as well by telling the truth whatever valuations are held by the other players [1]. A stronger notion (often called strategy proofness in social choice) is that players must not be able to profit even when they know how others value the cake. As in [3], we say that a cake cutting procedure is *truthful* iff there are no valuations where a player will do better by lying.

4 Other properties

We consider some other properties of (online) cake cutting procedures.

4.1 Surjectivity

This property has been studied in social choice but appears to have received less attention in fair division. It indicates whether the cake can be divided in every possible way. A cake cutting procedure is *surjective* iff there are valuation functions for the players such that every possible partition of the cake into n pieces is possible. Note that this definition only considers allocations where each player receives one continuous slice of cake. However, the definition of surjectivity could be easily extended to allocations where players can receive multiple slices. Our definition of surjectivity also ignores which player receives a particular slice. If an online cake cutting procedure is surjective, then there is an arrival ordering of the players and valuation functions such that any given player can receive a particular slice.

4.2 Scale invariance

Players may have different scales for their valuation functions. Scale invariance indicates that this is unimportant. A cake cutting procedure is *scale invariant* iff the allocation of cake is unchanged when a player's valuation is uniformly multiplied by a constant factor. It turns out that scale invariance is not difficult to achieve. Indeed, all the online cake cutting procedures we shall consider here are scale invariant.

4.3 Sequentiality

In some situations we may want cake to be cut from one end. This may be the case, for instance, when the cake represents time on a shared device. An onlike cake cutting procedure is *sequential* iff the slice given to any player is to the left of any slice given to a player who is later to depart.

4.4 Order monotonicity

A player's allocation of cake typically depends on their arrival order. We say that a cake cutting procedure is *order monotonic* iff a player's valuation of their cake does not decrease when they are moved earlier in the arrival ordering (and all other players have the same arrival ordering). Note that as the moved player can receive cake of greater value, players who depart after them may now receive cake with less value. A positive interpretation of order monotonicity is that players are encouraged to participate as early as possible. On the other hand, players who have to arrive late may receive less value.

5 General results

The fact that some players may depart before others arrive place some fundamental limitations on the fairness of online cake cutting procedures.

Theorem 1. *No online cake cutting procedure is proportional, envy free or equitable.*

Proof: Suppose the procedure is proportional. Then every player is allocated some cake. As the cake cutting procedure is online, at least one player departs before the final player arrives. Since the valuation function of the final player to arrive is not known when the first player departs, the cake allocated to the first player to depart cannot depend on the valuation function of the final player to arrive. Similarly, the valuation function of the final player to arrive cannot change who is the first player to depart. Consider the situation in which the final player to arrive has a valuation function that only values the cake allocated to the first player to depart. Whatever cake is allocated to the final player to arrive will be of no value to them. Hence the cake cutting procedure cannot be proportional.

Suppose the procedure is envy free. We consider the case where all players have valuation functions that assign some value to every slice. Every player is allocated some cake otherwise they will envy the players who are allocated cake (and by assumption a cake cutting procedure must allocate cake to at least one player). As before, the cake allocated to the first player to depart cannot depend on the valuation function of the final player to arrive. We now modify the valuation function of the last player to arrive so that the value of the cake remaining when the first player departs is $\frac{1}{n^2}$ of the value it was before. Even if we allocate all the remaining cake to the last player to arrive, the value of this cake cannot now equal the value they assign to the cake allocated to the first player to depart. Hence the last player to arrive will envy the first player to depart. By a similar argument, the procedure cannot be equitable. ♡

Online cake cutting procedures can, however, possess many of the other properties.

Theorem 2. *Online cake cutting procedures can be forward proportional, forward envy free, weakly Pareto optimal, truthful, scale invariant, sequential and order monotonic.*

Proof: Consider the online cake cutting procedure which allocates all the cake to the first player to arrive. ♡

Unfortunately, allocating all the cake to one player is not very fair to the other players. We therefore consider some specific online cake cutting procedures which divide the cake more equitably. It remains an important open problem to identify natural axioms that these procedures satisfy which are not satisfied by the trivial allocation of all cake to one player.

6 Online Cut-and-Choose

The cut-and-choose procedure for two players dates back to antiquity. It appears nearly three thousand years ago in Hesiod's poem *Theogeny* where Prometheus divides a cow and Zeus selects the part he prefers. Cut-and-choose is also enshrined in the United Nation's 1982 Convention of the Law of the Sea where it is put forward to divide the seabed for mining. In cut-and-choose, one player cuts the cake and the other takes the "half" that they most prefer. Cut-and-choose is proportional, envy free, Pareto optimal, weakly truthful, and surjective. However, it is not equitable, nor it is truthful.

We can use cut-and-choose as the basis of an online cake cutting procedure. The first player to arrive cuts the cake and waits for the next player to arrive. Either the next player to arrive chooses this piece and departs, or the next player to arrive declines this piece and the waiting player takes this piece and departs. If more players are to arrive, the remaining player cuts the cake and we repeat the process. Otherwise, the remaining player is the last player to be allocated cake and departs with whatever is left. We assume that all players know how many players will arrive.

Running Example:. *Suppose there are three players, the first player values only* $[\frac{1}{2}, 1]$*, the second player values only* $[\frac{1}{3}, 1]$*, and the third player values only* $[0, \frac{3}{4}]$*. We suppose that they uniformly value slices within these intervals. If we operate the online version of cut-and-choose, the first player will arrive and cut off the slice* $[0, \frac{2}{3}]$ *as they assign this slice* $\frac{1}{3}$ *the total value of the cake. The second player then arrives. As they assign this slice with* $\frac{1}{2}$ *the total value of the cake and they are only expecting* $\frac{1}{3}$ *of the total, the second player is happy to take this slice and depart. The first player then cuts off the slice* $[\frac{2}{3}, \frac{5}{6}]$ *as they assign this* $\frac{1}{3}$ *of the total value of the cake (and* $\frac{1}{2}$ *of the value remaining after the second player departed with their slice). The third player then arrives. As they assign the slice* $[\frac{2}{3}, \frac{5}{6}]$ *with all of the total value of the remaining cake and they are only expecting* $\frac{1}{2}$ *of whatever remains, the third player is happy to take this slice and depart. The first player now takes what remains, the slice* $[\frac{5}{6}, 1]$*. It can be claimed that everyone is happy as the first player received a "fair" proportion of the cake, whilst the other two players received slices that were of even greater proportional value to them.*

This online version of the cut-and-choose procedure has many (but not all) of the desirable properties described earlier.

Theorem 3. *The online cut-and-choose procedure is forward proportional, immediately envy free, weakly truthful, surjective, scale invariant and sequential. However, it is not proportional, (forward) envy free, equitable, (weakly) Pareto optimal, truthful or order monotonic.*

Proof: Consider the player cutting the cake. As they are risk averse, and as there is a chance that they will have to take the slice of cake that they cut, they will cut a slice that is at least $\frac{1}{k}$ of the total remaining value where k is the number of players still to be allocated cake. Similarly they will not cut a slice that is more than $\frac{1}{k}$ of the total remaining value for fear that the next player to arrive will take it, leaving behind cake that if it is divided proportionally gives them a slice of small value. Hence, the procedure is forward proportional and weakly truthful. It is also immediately envy free since each slice that the cutting player sees being allocated has the same value. To demonstrate surjectivity, consider the partition that allocates the ith player with the slice $[a_i, a_{i+1}]$ where $a_1 = 0$ and $a_{n+1} = 1$. We construct a valuation for the ith player ($i < n-1$) that assigns a value 0 to $[0, a_i]$, a value 1 to $[a_i, a_{i+1}]$, a value 0 to $[a_{i+1}, a_{i+2}]$, a value $n-i$ to $[a_{i+2}, 1]$. For the $n-1$th player, we construct a valuation function that assigns a value 0 to $[0, a_{n-1}]$, and values of 1 to both $[a_{n-1}, a_n]$ and $[a_n, 1]$. Finally, we construct a valuation function for the nth player that assigns a value 0 to $[0, a_n]$, and a value of 1 to $[a_n, 1]$. With these valuation functions, the ith player gets the slice $[a_i, a_{i+1}]$. Finally, it is easy to see that the procedure is scale invariant and sequential.

To show that this procedure is not proportional, (forward) envy free, equitable, (weakly) Pareto optimal truthful or order monotonic consider 4 players and a cake in which the first player places a value of 3 units on $[0, \frac{1}{4}]$, 1 unit on $[\frac{1}{4}, \frac{3}{4}]$ and 8 units on $[\frac{3}{4}, 1]$, the second player places a value of 0 units on $[0, \frac{1}{4}]$, 4 units on $[\frac{1}{4}, \frac{1}{2}]$, 8 units on $[\frac{1}{2}, \frac{5}{8}]$, and 0 units on $[\frac{5}{8}, 1]$, the third player places a value of 6 units on $[0, \frac{1}{4}]$ 0 units on $[\frac{1}{4}, \frac{1}{2}]$, 1 unit on $[\frac{1}{2}, \frac{5}{8}]$, 2 units on $[\frac{5}{8}, \frac{3}{4}]$, and 3 units on $[\frac{3}{4}, 1]$, and the fourth player places a value of 0 units on $[0, \frac{1}{4}]$, 9 units on $[\frac{1}{4}, \frac{1}{2}]$, 1 unit on $[\frac{1}{2}, \frac{3}{4}]$, and 2 units on $[\frac{3}{4}, 1]$.

If we apply the online cut-and-choose procedure, the first player will cut off and keep the slice $[0, \frac{1}{4}]$, the second player will cut off and keep $[\frac{1}{4}, \frac{1}{2}]$, The third player will now cut the cake into two pieces: $[\frac{1}{2}, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$. The fourth player will take the slice $[\frac{3}{4}, 1]$, leaving the third player with the slice $[\frac{1}{2}, \frac{3}{4}]$

The procedure is not proportional as the fourth player only receives $\frac{1}{6}$ of the total value of the cake, not (forward) envy free as the first player envies the fourth player, and not equitable as players receive cake of different value. The procedure is not (weakly) Pareto optimal as allocating the first player with $[\frac{3}{4}, 1]$, the second player with $[\frac{1}{2}, \frac{3}{4}]$, the third player with $[0, \frac{1}{4}]$, and the fourth player with $[\frac{1}{4}, \frac{1}{2}]$ gives all players a slice of greater value.

The proceure is not truthful as the second player can get a larger and more valuable slice by misrepresenting their preferences and cutting the cake into the slice $[\frac{1}{4}, \frac{5}{8}]$. Finally, the procedure is not order monotonic as the value of the cake allocated to the fourth player decreases from 2 units to $\frac{3}{2}$ units when they arrive before the third player. ♡

7 Online moving knife

Another class of procedure for cutting cakes uses one or more moving knives. For example, in the Dubins-Spanier procedure for n players [6], a knife is moved across the cake from left to right. When a player shouts "stop", the cake is cut and this player takes the piece to the left of the knife. The procedure then continues with the remaining $n-1$ players until just one player is left (who takes whatever remains). This procedure is proportional but is not envy-free. However, only the first $n-2$ players to be allocated slices of cake can be envious.

We can use the Dubins-Spanier procedure as the basis of an online moving knife procedure. The first k players ($k \geq 2$) to arrive perform one round of a moving knife procedure to select a slice of the cake. Whoever chooses this slice, departs. At this point, if all players have arrived, we continue the moving knife procedure with $k-1$ players. Alternatively the next player arrives and we start again a moving knife procedure with k players. As before, we assume that all players know how many players will arrive.

Running Example:. *Consider again the example in which there are three players, the first player values only $[\frac{1}{2}, 1]$, the second player values only $[\frac{1}{3}, 1]$, and the third player values only $[0, \frac{3}{4}]$. If we operate the online version of the moving knife procedure, the first two players will arrive and perform one round of the moving knife procedure. The second player will be the first to call "cut" and will depart with the slice $[0, \frac{5}{9}]$ (as this has $\frac{1}{3}$ of the total value of the cake for them). The third player will now arrive and perform a round of the moving knife procedure with the first player using the remaining cake, $[\frac{5}{9}, 1]$. The third player will be the first to call "cut" and will depart with the slice $[\frac{5}{9}, \frac{47}{72}]$ (as this has $\frac{1}{2}$ the total value of the remaining value for them). The first player will then depart with what remains, the slice $[\frac{47}{72}]$. It can be claimed that everyone is happy as the second and third players received a "fair"' proportion of the cake that was left when they first arrived, whilst the first player received an even greater proportional value.*

This online version of the moving knife procedure has the same desirable properties as the online version of the cut-and-choose procedure.

Theorem 4. *The online moving knife procedure is forward proportional, immediately envy free, weakly truthful, surjective, scale invariant and sequential. However, it is not proportional, (forward) envy free, equitable, (weakly) Pareto optimal, truthful or order monotonic.*

Proof: Suppose j players ($j > 1$) have still to be allocated cake. Consider any player who has arrived. They will call "cut" as soon as the knife reaches $\frac{1}{j}$ of the value of the cake left for fear that they will will receive cake of less value at a later stage. Hence, the procedure is weakly truthful and forward proportional. The procedure is also immediately envy free as they will assign less value to any slice that is allocated after their arrival and before their departure. To demonstrate surjectivity, consider the partition that allocates the ith player with the slice $[a_i, a_{i+1}]$ where $a_1 = 0$ and $a_{n+1} = 1$. We construct a valuation for the ith player ($i < n$) that assigns a value 0 to $[0, a_i]$, a value 1 to $[a_i, a_{i+1}]$, a value $n-i$ to $[a_{i+1}, 1]$. Finally, we construct a valuation function for the nth player that assigns a value 0 to $[0, a_n]$, and a value of 1 to $[a_n, 1]$. With these valuation functions, the ith player gets the slice $[a_i, a_{i+1}]$. Finally, it is easy to see that the procedure is scale invariant and sequential.

To show that this procedure is not proportional, (forward) envy free, equitable, (weakly) Pareto optimal truthful consider again the example with 4 players used in the last proof. We suppose that $k = 2$ (i.e. at any one time, two players are watching the knife). The first player calls "cut" and departs with the slice $[0, \frac{1}{4}]$. The second player calls "cut" and departs with the slice $[\frac{1}{4}, \frac{1}{2}]$. Finally, the third player calls "cut" and departs with the slice $[\frac{1}{2}, \frac{2}{4}]$, leaving the fourth player with the slice $[\frac{3}{4}, 1]$.

The procedure is not proportional as the fourth player only receives $\frac{1}{6}$ of the total value of the cake, not (forward) envy free as the first player envies the fourth player, and not equitable as players receive cake of different value. The procedure is not (weakly) Pareto optimal as allocating the first player with $[\frac{3}{4}, 1]$, the second player with $[\frac{1}{2}, \frac{3}{4}]$, the third player with $[0, \frac{1}{4}]$ and the fourth player with $[\frac{1}{4}, \frac{1}{2}]$ gives all players a slice of greater value.

The proceure is not truthful as the second player can get a larger and more valuable slice by misrepresenting their preferences and not calling "cut" until the knife is about to reach $\frac{5}{8}$th of the way along the cake.

Finally, to show that the procedure is not order monotonic consider 3 players and a cake in which the first player places a value of 2 units on each of $[0, \frac{1}{3}]$, $[\frac{1}{3}, \frac{2}{3}]$, and $[\frac{2}{3}, 1]$, the second player places a value of 0 units on $[0, \frac{1}{3}]$, 3 units on each of $[\frac{1}{3}, \frac{2}{3}]$ and $[\frac{2}{3}, 1]$, and the third player places a value of 2 units on $[0, \frac{1}{6}]$, 0 units on each of $[\frac{1}{6}, \frac{1}{3}]$ and $[\frac{1}{3}, \frac{2}{3}]$, and 4 units on $[\frac{2}{3}, 1]$. As before, we suppose that $k = 2$ (i.e. at any one time, two players are watching the knife). The first player calls "cut" and departs with the slice $[0, \frac{1}{3}]$. The second player calls "cut" and departs with the slice $[\frac{1}{3}, \frac{2}{3}]$, leaving the third player with the slice $[\frac{2}{3}, 1]$. On the other hand, if the third player arrives ahead of the second player then the value of the cake allocated to them drops from 4 units to 2 units. Hence the procedure is not order monotonic. ♡

8 Online Mark-and-Choose

A possible drawback of both of the online cake cutting procedures proposed so far is that the first player to arrive can be the last player to depart. What if we want a procedure in which players can depart soon after they arrive? The next procedure has such a property. Players will depart as soon as the next player arrives (except for the last player to arrive who takes whatever cake remains). However, the new procedure is no longer sequential. It may not allocated cake from one end. In addition, the new procedure does not necessarily allocate continuous slices of cake.

In the online mark-and-choose procedure, the first player to arrive marks the cake into n pieces. The second player to arrive selects one piece to give to the first player who then departs. The second player then marks the remaining cake into $n - 1$ pieces and waits for the third player to arrive. The procedure repeats in this way until the last player arrives. The last player to arrive selects which of the two halves marked by the penultimate player should be allocated to the penultimate player. The last player then takes whatever remains.

Running Example:. *Consider again the example in which there are three players, the first player values only $[\frac{1}{2}, 1]$, the second player values only $[\frac{1}{3}, 1]$, and the third player values only $[0, \frac{3}{4}]$. If we operate the online version of the mark-and-choose procedure, the first player will arrive and mark the cake into 3 equally valued pieces: $[0, \frac{2}{3}]$, $[\frac{2}{3}, \frac{5}{6}]$, and $[\frac{5}{6}, 1]$, The second player then arrives and selects the least valuable piece for the first player to take. In fact, both $[\frac{2}{3}, \frac{5}{6}]$ and $[\frac{5}{6}, 1]$ are each worth $\frac{1}{4}$ of the total value of the cake to the second player. They will therefore choose between them arbitrarily. Suppose the second player decides to give the slice $[\frac{2}{3}, \frac{5}{6}]$ to the first player. Note that the first player assigns this slice with $\frac{1}{3}$ of the total value of the cake. This leaves behind two sections of cake: $[0, \frac{2}{3}]$ and $[\frac{5}{6}, 1]$. The second player then marks what remains into two equally valuable pieces: the first is the interval $[0, \frac{7}{12}]$ and the second contains the two intervals $[\frac{7}{12}, \frac{2}{3}]$ and $[\frac{5}{6}, 1]$. The third player then arrives and selects the least valuable piece for the second player to take. The first piece is worth $\frac{7}{12}$ of the total value of the cake to the third player. As this is over half the total value, the other piece must be worth less. In fact, the second piece is worth $\frac{1}{4}$ of the total value. The third player therefore gives the second piece to the second player. This leaves the third player with the remaining slice $[0, \frac{7}{12}]$. It can again be claimed that everyone is happy as the first players received a "fair"' proportion of the cake that was left when they arrived, whilst both the second and third player received an even greater proportional value.*

This procedure again has the same desirable properties as the online version of the cut-and-choose and moving knife procedures.

Theorem 5. *The online mark-and-choose procedure is forward proportional, immediately envy free, weakly truthful, surjective, and scale invariant. However, it is not proportional, (forward) envy free, equitable, (weakly) Pareto optimal, truthful, order monotonic or sequential.*

Proof: Any player marking the cake will divide it into slices of equal value (for fear that they will be allocated one of the less valuable slices). Similarly, a player selecting a slice for another player will select the slice of least value to them (to maximize the value that they will receive next). Hence, the procedure is weakly truthful and forward proportional. The procedure is also immediately envy free as they will assign less value to the slice that they select for the departing player than the value of the slices that they mark. To demonstrate surjectivity, consider the partition that allocates the ith player with the slice $[a_i, a_{i+1}]$ where $a_1 = 0$ and $a_{n+1} = 1$. We construct a valuation for the ith player ($i < n$) that assigns a value 0 to $[0, a_i]$, a value 1 to $[a_i, a_{i+1}]$, a value $n - i$ to $[a_{i+1}, 1]$. Finally, we construct a valuation function for the nth player that assigns a value 0 to $[0, a_n]$, and a value of 1 to $[a_n, 1]$. With these valuation functions, the ith player gets the slice $[a_i, a_{i+1}]$. Finally, it is easy to see that the procedure is scale invariant.

To show that this procedure is not proportional, (forward) envy free, equitable, (weakly) Pareto optimal or truthful consider again the example with 4 players used in the last two proofs. The first player marks and is assigned the slice $[0, \frac{1}{4}]$ by the second player. The second player then marks and is assigned the slice $[\frac{1}{4}, \frac{1}{2}]$. The third player then marks and is assigned the slice $[\frac{1}{2}, \frac{3}{4}]$, leaving the fourth player with the slice $[\frac{3}{4}, 1]$.

The procedure is again not proportional as the fourth player only receives $\frac{1}{6}$ of the total value of the cake, not (forward) envy free as the first player envies the fourth player, and not equitable as players receive cake of different value. The procedure is not (weakly) Pareto optimal as allocating the first player with $[\frac{3}{4}, 1]$, the second player with $[\frac{1}{2}, \frac{3}{4}]$, the third player with $[0, \frac{1}{4}]$, and the fourth player with $[\frac{1}{4}, \frac{1}{2}]$ gives all players a slice of greater value.

The proceure is not truthful as the second player can get a larger and more valuable slice by misrepresenting their preferences and marking the cake into the slices $[\frac{1}{4}, \frac{5}{8}]$, $[\frac{5}{8}, \frac{3}{4}]$, and $[\frac{3}{4}, 1]$. In this situation, the third player will allocate the second player with the slice $[\frac{1}{4}, \frac{5}{8}]$ which is of greater value to the second player. It is also easy to see that the procedure is not sequential.

Finally, to show that the procedure is not order monotonic consider 3 players and a cake in which the first player places a value of 4 units on each of $[0, \frac{1}{3}]$, $[\frac{1}{3}, \frac{2}{3}]$ and $[\frac{2}{3}, 1]$, the second player places a value of 0 units on $[0, \frac{1}{3}]$, 6 units on $[\frac{1}{3}, \frac{2}{3}]$, and 3 units on each of $[\frac{2}{3}, \frac{5}{6}]$, and $[\frac{5}{6}, 1]$, and the third player places a value of 2 unit on $[0, \frac{1}{6}]$, 0 units on each of $[\frac{1}{6}, \frac{1}{3}]$ and $[\frac{1}{3}, \frac{2}{3}]$, and 5 units on each of $[\frac{2}{3}, \frac{5}{6}]$ and $[\frac{5}{6}, 1]$. The first player marks and is allocated the slice $[0, \frac{1}{3}]$. The second player marks and is allocated the slice $[\frac{1}{3}, \frac{2}{3}]$, leaving the third player with the slice $[\frac{2}{3}, 1]$. On the other hand, suppose the third player arrives ahead of the second player. In this case, the third player marks the cake into two slice, $[\frac{1}{3}, \frac{5}{6}]$ and $[\frac{5}{6}, 1]$. The second player allocates the third player the slice $[\frac{5}{6}, 1]$. Hence, the value of cake allocated to the third player drops from 10 units to 5 units when they go second in the arrival order. Hence the procedure is not order monotonic. ♡

9 Bounded number of players

One variation of online cake cutting is when the number of players is not known but all players have the (same) upper bound, n_{max} on the number of persons to be allocated cake. We consider three cases: players know their arrival position and when the last player arrives; players do not know their arrival position but do know when the last player arrives; players do not know when the last player arrives.

9.1 Known arrival order and last player

In this case, each player knows how many players have arrived before them, and players know when no more players are to arrive. In this case, we can still operate the online cut-and-choose procedure. Given the risk averse nature of the players, each player will cut off a slice of cake of value $\frac{1}{n_{max}-k}$ of the total where k is the number of players who have already been allocated cake.

9.2 Unknown arrival order but known last player

In this case, players do not know how many players have arrived before them, but do know when no more players are to arrive. We can again operate the online cut-and-choose procedure. The first player will cut off a slice of cake of value $\frac{1}{n_{max}-k}$ of the total where k is the number of players already allocated cake (e.g. in the first round, the first player cuts off a slice of value $\frac{1}{n_{max}}$ of the total, if this is accepted by the second player, they then cut off a slice of value $\frac{1}{n_{max}-1}$ of the total, and so on).

We can suppose that the second player to arrive will look at the cake and deduce they are the second player to arrive (since they will assign the total value of the cake to the two pieces). If they are not the last player to arrive, they will accept the offered slice if it is greater than or equal to $\frac{1}{n_{max}}$ of the total. If they are the last player to arrive, they will accept the offered slice if it is greater than or equal to $\frac{1}{2}$ of the total. Otherwise, if there are no more players are to arrive, they will take whatever cakes remain. If there are more players to arrive, they will cut off a new slice of value $\frac{1}{n_{max}-j}$ of the total where j is the number of players already allocated cake (e.g. the second player first cuts off a slice of value $\frac{1}{n_{max}-1}$ of the total, if this is accepted by the next player to arrive, the second player then cuts off a slice of value $\frac{1}{n_{max}-2}$ of the total, and so on).

We can suppose that the third (or any later) player to arrive can only deduce that they are not the first or second player to arrive. If they are not the last player to arrive, they will accept the offered slice if it is greater than or equal to $\frac{1}{n_{max}-1}$ of the total. If they are the last player to arrive, they will accept the offered slice if it is greater than or equal to $\frac{1}{2}$ of the total. Otherwise, if there are no more players are to arrive, they will take whatever cakes remain. If there are more players to arrive, they will cut off a new slice of value $\frac{1}{n_{max}-j}$ of the total where j is the number of players already allocated cake (e.g. they first cut off a slice of value $\frac{1}{n_{max}-2}$ of the total, if this is accepted by the next player to arrive, they then cut off a slice of value $\frac{1}{n_{max}-3}$ of the total, and so on).

9.3 Unknown last player

In the third case, players do know when no more players are to arrive. We now have a potential deadlock problem in operating the online cut-and-choose procedure. We need some mechanism to ensure that the last player to arrive is allocated cake. One option is to introduce a clock. If a player waits longer than a certain time, then they can take whatever cake remains. With this modification, we can again operate the online cut-and-choose procedure.

9.4 Moving knife procedures

We can also use the online moving knife procedure when there is only a bound on the number of players to be allocated cake. The results are very similar to the online cut-and-choose procedure, and depend on whether players know when the last player arrives and on whether players know how many players have been allocated cake before them.

10 Related work

There is an extensive literature on fair division and cake cutting procedures. See, for instance, [2] for an introduction. There has, however, been considerably less work on fair division problems similar to those considered here.

Thomson considers a generalization of fair division problems where the number of players may increase [7]. He explores from an axiomatic perspective whether it is possible to have a procedure in which players' allocations are monotonic (i.e. their values do not increase as the number of players increase) combined with other common properties like weak Pareto optimality.

Cloutier *et al.* consider a different generalization of the cake cutting problem in which the number of players is fixed but there are multiple cakes [5]. This can model situations where, for example, players wish to choose shifts across multiple days. Note that this problem can be reduced to multiple single cake cutting problems unless the players' valuations across cakes are linked (e.g. you prefer the same shift each day compared to different shifts).

A number of authors have studied distributed mechanisms for fair division (see, for example, [4]). In such mechanisms, players typically agree locally on deals to exchange some of the goods in their possession. The usual goal is to identify conditions under which the system converges to a fair or envy free allocation.

11 Conclusions

We have proposed an online form of the cake cutting problem. This permits us to explore the concept of fair division when players arrive and depart during the process of dividing a resource. It can be used to model situations, such as on the internet, when we need to divide resources asynchronously. There are many possible future directions for this work. One extension would be to indivisible goods. Another extension would be to undesirable goods (like chores) where we want as little of them as possible. In addition, it would be interesting to consider variants of the online cake cutting problem where players have information about the valuation functions of the other players.

Acknowledgements

NICTA is funded by the Department of Broadband, Communications and the Digital Economy, and the Australian Research Council.

References

[1] S.J. Brams, M.A. Jones, and C. Klamler. Better ways to cut a cake. *Notices of the AMS*, 53(11):1314–1321, 2006.

[2] S.J. Brams and A.D. Taylor. *Fair Division: From cake-cutting to dispute resolution.* Cambridge University Press, Cambridge, 1996.

[3] Y. Chen, J.K. Lai, D.C. Parkes, and A.D. Procaccia. Truth, justice, and cake cutting. In *Proceedings of the 24th National Conference on AI.* Association for Advancement of Artificial Intelligence, 2010.

[4] Y. Chevaleyre, U. Endriss, and N. Maudet. Distributed fair allocation of indivisible goods. Working paper, ILLC, University of Amsterdam, 2009.

[5] J. Cloutier, K.L. Nyman, and F.E. Su. Two-player envy-free multi-cake division. *Mathematical Social Sciences*, 59(1):26 – 37, 2010.

[6] L.E. Dubins and E.H. Spaier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(5):1–17, 1961.

[7] W. Thomson. The fair division of a fixed supply among a growing population. *Mathematics of Operations Research*, 8(3):319–326, 1983.

Toby Walsh
NICTA and UNSW
Sydney, Australia
Email: `toby.walsh@nicta.com.au`

Three Hierarchies of Simple Games Parameterized by "Resource" Parameters

Tatiana Gvozdeva, Lane A. Hemaspaandra, and Arkadii Slinko

Abstract

This paper contributes to the program of numerical characterization and classification of simple games outlined in the classic monograph of von Neumann and Morgenstern (1944). We suggest three possible ways to classify simple games beyond the classes of weighted and roughly weighted games. To this end we introduce three hierarchies of games and prove some relations between their classes. We prove that our hierarchies are true (i.e., infinite) hierarchies. In particular, they are strict in the sense that more of the key "resource" (which may, for example, be the size or structure of the "tie-breaking" region where the weights of the different coalitions are considered so close that we are allowed to specify either winningness or nonwinningness of the coalition), yields the flexibility to capture strictly more games.

1 Introduction

A simple game is a mathematical object that is used in economics and political science to describe the distribution of power among coalitions of players [10, 11]. Recently simple games have been studied as access structures of secret sharing schemes [2]. They have also appeared in a variety of mathematical and computer science contexts under various names, e.g., monotone boolean [5] or switching functions and threshold functions [6]. Simple games are closely related to hypergraphs, coherent structures, Sperner systems, clutters, and abstract simplicial complexes. The term "simple" was introduced by von Neumann and Morgenstern (1944) because in this type of games players strive not for monetary rewards but for power, and each coalition is either all-powerful or completely ineffectual. However these games are far from being simple.

An important class of simple games—well studied in economics—is the weighted majority games [10, 11]. In such a game every player is assigned a real number, his weight. The winning coalitions are the sets of players whose weights total at least q, a certain threshold. However, it is well known that not every simple game has a representation as a weighted majority game [10]. The first step in attempting to characterize nonweighted games was the introduction of the class of roughly weighted games [9]. Formally, a simple game G on the player set $P = [n] = \{1, 2, \ldots, n\}$ is *roughly weighted* if there exist nonnegative real numbers $w_1, \ldots, w_n$ and a real number q, called the *quota*, not all equal to zero, such that for $X \in 2^P$ the condition $\sum_{i \in X} w_i > q$ implies X is winning, and $\sum_{i \in X} w_i < q$ implies X is losing. This concept realizes a very common idea in social choice that sometimes a rule needs an additional "tie-breaking" procedure that helps to decide the outcome if the result falls on a certain "threshold." Taylor and Zwicker [9] demonstrated the usefulness of this concept. Rough weightedness was studied by Gvozdeva and Slinko [4], where it was characterized in terms of trading transforms, similar to the characterization of weightedness by Elgot [3] and Taylor and Zwicker [8].

It might seem that nonweighted games and even games without rough weights are weird. However, an important observation of von Neumann and Morgenstern [10, Section 53.2.6] states that they "correspond to a different organizational principle that deserves closer study." In some of these games, as they noted, all the minimal winning coalitions are minorities and at the same time "no player has any advantage over any other" (e.g., the

Fano game introduced later). This is an attractive feature for secret sharing as in the case of large number of users it is advantageous to keep minimal authorized coalitions relatively small. This is may be why weighted threshold secret sharing schemes were largely ignored and were characterized only recently [1].

The parameter of the first hierarchy reflects the balance of power between small and large coalitions; the larger this parameter the more powerful some of the small coalitions are. Gvozdeva and Slinko [4] proved that for a game G that is not roughly weighted there exists a certificate of nonweightedness (see the definition in Section 2) of the form

$$\mathcal{T} = (X_1, \ldots, X_j, P; Y_1, \ldots, Y_j, \emptyset), \tag{1}$$

where $X_1, \ldots, X_j$ are winning coalitions of G, P is the grand coalition, and $Y_1, \ldots, Y_j$ are losing coalitions. However, sometimes it is possible to have more than one grand coalition in the certificate. This may occur when coalitions $X_1, \ldots, X_j$ are small but nonetheless winning.

A certificate of nonweightedness of the form

$$\mathcal{T} = (X_1, \ldots, X_j, P^\ell; Y_1, \ldots, Y_j, \emptyset^\ell) \tag{2}$$

will be called ℓ-potent of length $j + \ell$. Each game that possesses such a certificate will be said to belong to the class of games $\mathcal{A}_q$, where $q = \ell/(j + \ell)$. The parameter q can take values in the open interval $(0, \frac{1}{2})$. We will show that $\mathcal{A}_p \supseteq \mathcal{A}_q$ for any p and q such that $0 < p \leq q < \frac{1}{2}$ and that the inclusion $\mathcal{A}_p \supseteq \mathcal{A}_q$ is strict as soon as $p < q$.

Another hierarchy emerges when we allow several thresholds instead of just one in the case of roughly weighted games. We say that a simple game G belongs to the class $\mathcal{B}_k, k \in \{1, 2, 3, \ldots\}$, if there are k thresholds $0 < q_1 \leq q_2 \leq \cdots \leq q_k$ and any coalition with total weight of players smaller than q_1 is losing, any coalition with total weight greater than q_k is winning . We also impose an additional condition that, if a coalition X has total weight $w(X)$ which satisfies $q_1 \leq w(X) \leq q_k$, then $w(X) = q_i$ for some i. All games of the class $\mathcal{B}_1$ are roughly weighted. In fact, as we'll prove in Section 4 almost all roughly weighted games to this class: $\mathcal{B}_1$ is exactly the class of roughly weighted games with nonzero quota. We will show that the Fano game [4] belongs to $\mathcal{B}_2$ but does not belong to $\mathcal{B}_1$. We prove that $\mathcal{B}$-hierarchy is strict, that is,

$$\mathcal{B}_1 \subsetneq \mathcal{B}_2 \subsetneq \cdots \subsetneq \mathcal{B}_\ell \subsetneq \cdots,$$

with the union of these classes being the class of all simple games.

Yet another way to capture more games is by making the threshold "thicker." We here will not use a point but rather an interval $[a, b]$ for the threshold, $a \leq b$. That is, all coalitions with total weight less than a will be losing and all coalitions whose total weight is greater than b winning. This time—in contrast with the k limit of $\mathcal{B}_k$—we do not care how many different values weights of coalitions falling in $[a, b]$ may take on. A good example of this situation would be a faculty vote, where if neither side controls a 2/3 majority (calculated in faculty members or their grant dollars), then the Dean would decide the outcome as he wished. We can keep weights normalized so that the lower end of the interval is fixed at 1. Then the right end of the interval α becomes a "resource" parameter. Formally, a simple game G belongs to class $\mathcal{C}_\alpha$ if all coalitions in G with total weight less than 1 are losing and every coalition whose total weight is greater than α is winning. We show that the class of all simple games is split into a hierarchy of classes of games $\{\mathcal{C}_\alpha\}_{\alpha \in [1,\infty)}$ defined by this parameter. We show that as α increases we get strictly greater descriptive power, i.e., strictly more games can be described, that is, if $\alpha < \beta$, then $\mathcal{C}_\alpha \subsetneq \mathcal{C}_\beta$. In this sense the hierarchy is strict. This strict hierarchy result, and our strict hierarchy results for hierarchies $\mathcal{A}$ and $\mathcal{B}$, have very much the general flavor of hierarchy results found in computer science:

more resources yield more power (whether computational power to accept languages as in a deterministic or nondeterministic time hierarchy theorem, or as is the case here, description flexibility to capture more games).

The strictness of the latter hierarchy was achieved because we allowed games with arbitrary (but finite) numbers of players. The situation will be different if we keep the number of players n fixed. Then there is an interval $[1, s(n)]$ such that all games with n players belong to $\mathcal{C}_{s(n)}$ and $s(n)$ is minimal with this property. There will be also finitely many numbers $q \in [1, s(n)]$ such that the interval $[1, q]$ represents more n-player games than any interval $[1, q']$ with $q' < q$. We call the set of such numbers the nth *spectrum* and denote it $\mathrm{Spec}(n)$. We also call a game with n players *critical* if it belongs to $\mathcal{C}_\alpha$ with $\alpha \in \mathrm{Spec}(n)$ but does not belong to any $\mathcal{C}_\beta$ with $\beta < \alpha$. We calculate the spectrum for $n < 7$ and also produce a set of critical games, one for each element of the spectrum. We also try to give a reasonably tight upper bound for $s(n)$.

All three of our hierarchies provide measures of how close a given game is to being a simple weighted voting game. That is, they each quantify the nearness to being a simple weighted voting game (e.g., hierarchies $\mathcal{B}$ and $\mathcal{C}$ quantify based on the extent and structure of a "flexible tie-breaking" region). And the main theme and contribution of this paper is that we prove for each of the three hierarchies that allowing more quantitative distance from simple weighted voting games yields *strictly more* games, i.e., the hierarchies are proper hierarchies.

2 Preliminaries

Definition 1. *A simple game is a pair $G = (P, W)$, where W is a subset of the power set 2^P satisfying the monotonicity condition:*

if $X \in W$ and $X \subsetneq Y \subseteq P$, then $Y \in W$,

and $W \notin \{\emptyset, 2^P\}$ (nontriviality assumption).

Elements of the set W are called *winning coalitions*. We also define the set $L = 2^P \setminus W$ and call elements of this set *losing coalitions*. A winning coalition is said to be *minimal* if every its proper subset is a losing coalition. Due to monotonicity, every simple game is fully determined by the set of its minimal winning coalitions. A player which does not belong to any minimal winning coalitions is called *dummy*.

For $X \subseteq P$ we will denote its complement $P - X$ as X^c.

Definition 2. *A simple game is called proper if $X \in W$ implies that $X^c \in L$ and is called strong if $X \in L$ implies that $X^c \in W$. A simple game that is proper and strong is called a constant-sum game.*

The following definition is given as it has appeared in [4].

Definition 3. *A simple game $G = (P, W)$ is called roughly weighted if there exist non-negative real numbers $w_1, \ldots, w_n$ and a nonnegative real number q, not all equal to zero, such that for $X \in 2^P$ the condition $\sum_{i \in X} w_i < q$ implies $X \in L$ and $\sum_{i \in X} w_i > q$ implies $X \in W$. We say that $[q; w_1, \ldots, w_n]$ is a rough voting representation for G; the number q is called the* quota.

Example 1 (The Fano game)**.** *This important example first appeared in [10, Section 53.2.6]. Let $P = [7]$ be identified with the set of seven points of the projective plane of order two, called the Fano plane. Let us take the seven lines of this projective plane as minimal winning coalitions:*

$$\{1, 2, 3\},\ \{3, 4, 5\},\ \{1, 5, 6\},\ \{1, 4, 7\},\ \{2, 5, 7\},\ \{3, 6, 7\},\ \{2, 4, 6\}. \tag{3}$$

We will denote them by $X_1, \ldots, X_7$, respectively. This, as is easy to check, defines a constant-sum game the Fano. As we will see slightly later, it has no rough voting representation. As we can see from the list of minimal winning coalitions they are all minorities, yet symmetry makes all players equal in this example.

We remind the reader that a sequence of coalitions

$$\mathcal{T} = (X_1, \ldots, X_j; Y_1, \ldots, Y_j) \tag{4}$$

is a trading transform [9] if the coalitions $X_1, \ldots, X_j$ can be converted into the coalitions $Y_1, \ldots, Y_j$ by rearranging players. This can also be expressed as

$$|\{i : a \in X_i\}| = |\{i : a \in Y_i\}| \qquad \text{for all } a \in P.$$

We say that the length of $\mathcal{T}$ is j.

Definition 4. *A trading transform $(X_1, \ldots, X_j; Y_1, \ldots, Y_j)$ with all coalitions $X_1, \ldots, X_j$ winning and all coalitions $Y_1, \ldots, Y_j$ losing is called a certificate of nonweightedness. This certificate is said to be potent if the grand coalition P is among $X_1, \ldots, X_j$ and the empty coalition is among $Y_1, \ldots, Y_j$.*

Elgot proved (using a different terminology) that the existence of a certificate of nonweightedness implies that the game is not weighted and that every nonweighted game has one. Taylor and Zwicker [9] showed that for a nonweighted game with n player this certificate can be found of length at most 2^{2^n}; Gvozdeva and Slinko [4] lowered this bound to $(n+1)2^{\frac{1}{2}n \log_2 n}$.

Theorem 1 (Criterion of rough weightedness [4])**.** *A simple game G with n players is roughly weighted iff for no positive integer $j \leq (n+1)2^{\frac{1}{2}n \log_2 n}$ does there exist a potent certificate of nonweightedness of length j.*

In Example 1 the following eight winning coalitions $X_1, \ldots, X_7, P$ of the Fano game can be transformed into the following eight losing coalitions: $X_1^c, \ldots, X_7^c, \emptyset$. So the sequence

$$(X_1, \ldots, X_7, P; X_1^c, \ldots, X_7^c, \emptyset) \tag{5}$$

is a potent certificate of nonweightedness for this game. So the game is not roughly weighted, thanks to Theorem 1.

Theorem 2 ([4])**.** *The following games are roughly weighted:*

- *every game with 4 or less players,*
- *every strong or proper game with 5 or less players, and*
- *every constant sum game with 6 or less players.*

Definition 5 ([9], p. 6)**.** *We say that a player p in a game is a* dictator *if p belongs to every winning coalition and to no losing coalition. If all coalitions containing player p are winning, this player is called a* passer. *A player p is called a* vetoer *if p is contained in the intersection of all winning coalitions.*

Proposition 1 ([4])**.** *Suppose G is a simple game with n players. Then G is roughly weighted if any one of the following three conditions holds:*

(a) G has a passer.

(b) G has a vetoer.

(c) G has a losing coalition that consists of $n-1$ players.

Due to Proposition 1(a) there is one trivial way to make any game roughly weighted. This can be done by adding an additional player and making her a passer. Then we can introduce rough weights by assigning weight 1 to the passer and weight 0 to any other player and setting the quota equal to 0. Note, that if the original game is not roughly weighted, then such rough representation is unique. In our view, adding a passer trivializes the game but does not make it closer to a weighted majority game; this is why in definitions of our hierarchies $\mathcal{B}$ and $\mathcal{C}$ we disallow thresholds to be equal 0.

As in [4] we would like to represent trading transforms algebraically. Let $T = \{-1, 0, 1\}$ and let $T^n = T \times T \times \cdots T$ (n times). With any pair (X, Y) of subsets $X, Y \in [n]$ we define

$$\mathbf{v}_{X,Y} = \chi(X) - \chi(Y) \in T^n,$$

where $\chi(X)$ and $\chi(Y)$ are the characteristic vectors of subsets X and Y, respectively.

Let now $G = (P, W)$ be a simple game. We will associate an algebraic object with G. For any pair (X, Y), where X is winning and Y is losing, we put the pair in correspondence with the vector $\mathbf{v}_{X,Y}$. The set of all such vectors we will denote $I(G)$ and will call the ideal of the game. Saying that $(X_1, \ldots, X_j; Y_1, \ldots, Y_j)$ is a certificate of nonweightedness is equivalent to saying that the following vector sum of the ideal is $\mathbf{0}$: $\mathbf{v}_{X_1,Y_1} + \mathbf{v}_{X_2,Y_2} + \cdots + \mathbf{v}_{X_j,Y_j} = \mathbf{0}$. An ℓ-potent certificate $(X_1, \ldots, X_j, P^\ell; Y_1, \ldots, Y_j, \emptyset^\ell)$ will be represented as

$$\mathbf{v}_{X_1,Y_1} + \mathbf{v}_{X_2,Y_2} + \cdots + \mathbf{v}_{X_j,Y_j} + \ell \cdot \mathbf{1} = \mathbf{0},$$

where $\mathbf{1}$ is a vector whose entries are each 1.

3 The $\mathcal{A}$-Hierarchy

This hierarchy of classes $\mathcal{A}_\alpha$ tries to capture the richness of the class of games that do not have rough weights, and does so by introducing a parameter $\alpha \in (0, \frac{1}{2})$. As we already discussed, the larger this parameter the more power is given to some relatively small coalitions. Our method of classification is based on the existence of potent certificates of nonweightedness for such games [4].

Definition 6. *Let q be a rational number. A game G belongs to the class $\mathcal{A}_q$ of $\mathcal{A}$-hierarchy if G possesses a potent certificate of nonweightedness*

$$(X_1, \ldots, X_m; Y_1, \ldots, Y_m), \tag{6}$$

with ℓ grand coalitions among $X_1, \ldots, X_m$ and ℓ empty coalitions among $Y_1, \ldots, Y_m$, such that $q = \ell/m$. If α is irrational,we set $\mathcal{A}_\alpha = \bigcap_{q<\alpha} \mathcal{A}_q$.

It is easy to see that, if $q \geq \frac{1}{2}$, then $\mathcal{A}_q$ is empty. Indeed, suppose $q \geq \frac{1}{2}$ and $\mathcal{A}_q$ is not empty. Then there is a game G with a certificate of nonweightedness

$$\mathcal{T} = (X_1, \ldots, X_k, P^m; Y_1, \ldots, Y_k, \emptyset^m) \tag{7}$$

with $m \geq k$. This is not possible since m copies of P contain more elements than are contained in the sets $Y_1, \ldots, Y_k$ taken together and so (7) is not a trading transform. So our hierarchy consists of a family of classes $\{\mathcal{A}_\alpha\}_{\alpha \in (0, \frac{1}{2})}$. We would like to show that this hierarchy is strict, that is, a smaller parameter captures more games.

Proposition 2. *If* $0 < \alpha \leq \beta < \frac{1}{2}$*, then* $\mathcal{A}_\alpha \supseteq \mathcal{A}_\beta$*.*

Proof. It is sufficient to prove this statement when α and β are rational. Suppose that we have a game G in $\mathcal{A}_\beta$ that possesses a certificate of length n_1 with k_1 grand coalitions and $\beta = k_1/n_1$. Let $\alpha = k_2/n_2$. We can then represent these numbers as $\beta = m_1/n$ and $\alpha = m_2/n$, where $n = \operatorname{lcm}(n_1, n_2)$. Since $\alpha \leq \beta$, we have $m_2 \leq m_1$. Since $n = n_1 h$ and $m_1 = k_1 h$ for some integer h, we can now combine h certificates for G to obtain one with length n and m_1 grand coalitions. We reclassify the $m_1 - m_2$ grand coalitions into ordinary winning coalitions, and we will get a certificate for G of length n with m_2 grand coalitions. So $G \in \mathcal{A}_\alpha$. $\square$

We say that a game G is *critical* for $\mathcal{A}_\alpha$ if it belongs to $\mathcal{A}_\alpha$ but does not belong to any $\mathcal{A}_\beta$ with $\beta > \alpha$.

Theorem 3. *If* $0 < \alpha < \beta < \frac{1}{2}$*, then* $\mathcal{A}_\alpha \supsetneq \mathcal{A}_\beta$*.*

Proof. First, we will construct a two-parameter family of simple games. For any integers $a \geq 2$ and $b \geq 2$ let $G = ([a^2 + a + b + 1], W)$ be a simple game for which a coalition X is winning, exactly if $|X| > a^2 + 1$ or X contains a subset whose characteristic vector is a cyclic permutation of $(\underbrace{1, \ldots, 1}_{a+1}, \underbrace{0, \ldots, 0}_{a^2+b})$.

Let $X_1, \ldots, X_{a^2+a+b+1}$ be winning coalitions, whose characteristic vectors are cyclic permutations of $(\underbrace{1, \ldots, 1}_{a+1}, \underbrace{0, \ldots, 0}_{a^2+b})$. Also let $Y_1, \ldots, Y_{a^2+a+b+1}$ be losing coalitions, whose characteristic vectors are cyclic permutations of

$$(\underbrace{1, \ldots, 1}_{a}, 0, \underbrace{1, \ldots, 1}_{a}, 0, \underbrace{1, \ldots, 1}_{a}, 0, \ldots, \underbrace{1, \ldots, 1}_{a}, 0, 0, 1, \underbrace{0, \ldots, 0}_{b-1}),$$

where there are a groups of symbols $\underbrace{1, \ldots, 1}_{a}, 0$.

This game possesses the following potent certificate of nonweightedness

$$\mathcal{T} = (X_1, \ldots, X_{a^2+a+b+1}, P^{a^2-a}; Y_1, \ldots, Y_{a^2+a+b+1}, \emptyset^{a^2-a}). \tag{8}$$

So $G \in \mathcal{A}_{\frac{a^2-a}{2a^2+b+1}}$. Let us prove that G is critical for this class, that is, it does not belong to any $\mathcal{A}_{q'}$ for $q' > q$. Note that the vectors $\mathbf{v}_i = \mathbf{v}_{X_i,Y_i}$ belong to the ideal of this game. Note also that the sum of all coefficients of $\mathbf{v}_i$ is $\mathbf{v}_i \cdot \mathbf{1} = a - a^2$ and that for any other vector $\mathbf{v} \in I(G)$ from the ideal of this game we have $\mathbf{v} \cdot \mathbf{1} \geq a - a^2$.

Suppose G also has a potent certificate of nonweightedness

$$(A_1, \ldots, A_s, P^t; B_1, \ldots, B_s, \emptyset^t). \tag{9}$$

with $q' = \frac{t}{t+s} > \frac{a^2-a}{2a^2+b+1} = q$. The latter is equivalent to $\frac{a^2+a+b+1}{a^2-a} > \frac{s}{t}$. Let $\mathbf{u}_i = \mathbf{v}_{A_i,B_i} \in I(G)$, then (9) can be written as

$$\mathbf{u}_1 + \mathbf{u}_2 + \cdots + \mathbf{u}_s + t \cdot \mathbf{1} = \mathbf{0}.$$

As $\mathbf{u}_i \cdot \mathbf{1} \geq a - a^2$, taking the dot product of both sides with $\mathbf{1}$ we get $t(a^2+a+b+1) \leq s(a^2-a)$, which is equivalent to $\frac{a^2+a+b+1}{a^2-a} \leq \frac{s}{t}$, so we have reached a contradiction.

We will now show that any rational number between 0 and $\frac{1}{2}$ is representable as $\frac{a^2-a}{2a^2+1+b}$ for some positive integers $a \geq 2$ and $b \geq 2$. Let $\frac{p}{q} \in (0, \frac{1}{2})$. Then $q - 2p > 0$ and it is possible to choose a positive integer k such that $k^2 p(q - 2p) - kq - 3 > 0$. Take $a = kp$ and $b = k^2 p(q - 2p) - kq - 1$. Substituting these values we get $\frac{a^2-a}{2a^2+1+b} = \frac{p}{q}$. $\square$

4 $\mathcal{B}$-Hierarchy

The $\mathcal{B}$-hierarchy generalizes the idea behind rough weightedness to allow more "points of flexibility."

Definition 7. *A simple game $G = (P, W)$ belongs to $\mathcal{B}_k$ if there exist real numbers $0 < q_1 \leq q_2 \leq \cdots \leq q_k$, called thresholds, and a weight function $w: P \to \mathbb{R}^{\geq 0}$ such that*

(a) if $\sum_{i \in X} w(i) > q_k$, then X is winning,

(b) if $\sum_{i \in X} w(i) < q_1$, then X is losing,

(c) if $q_1 \leq \sum_{i \in X} w(i) \leq q_k$, then $w(X) = \sum_{i \in X} w(i) \in \{q_1, \ldots, q_k\}$.

Games from $\mathcal{B}_k$ will be sometimes called k-rough.

The condition $0 < q_1$ in Definition 7 is essential. If we allow the first threshold q_1 be zero, then every simple game can be represented as a 2-rough game. To do this we assign weight 1 to the first player and 0 to everyone else. It is also worthwhile to note that adding a passer does not change the class of the game, that is, a game G belongs to $\mathcal{B}_k$ iff the game G' obtained from G by adding a passer belongs to $\mathcal{B}_k$. This is because a passer can be assigned a very large weight. Thus $\mathcal{B}_1$ consists of the roughly weighted simple games with nonzero quota.

Example 2. *We know that the Fano game is not roughly weighted. Let us assign weight 1 to every player of this game and select two thresholds $q_1 = 3$ and $q_2 = 4$. Then each coalition whose weight falls below the first threshold is in L, and each coalition whose total weight exceeds the second threshold is in W. If a coalition has total weight of three or four, i.e., its weight is equal to one of the thresholds, it can be either winning or losing. Thus the Fano is a 2-rough game.*

Theorem 4. *For every natural number $k \in \mathbb{N}^+$, there exists a game in $\mathcal{B}_{k+1} \setminus \mathcal{B}_k$.*

Proof. We will construct a simple game that is a $(k+1)$-rough but not k-rough. Let $G_{k+1,n} = ([n], W)$ be a simple game with $n = 2k+4$ players. We have $k+2$ types of players with the ith type consisting of two elements $2i-1$ and $2i$. The set of minimal winning coalitions of this game is $W^m = \{\{2i-1, 2i\} \mid i = 1, 2, \ldots, k+2\}$.

If we assign weight 1 to every player, then $G_{k+1,n}$ is $(k+1)$-rough game with thresholds $q_1 = 2$, $q_2 = 3$, $\ldots$, $q_{k+1} = k+2$. Let us assume that this game is j-rough for some $j < k+1$, and let w be the new weight function and let $r_1, \ldots, r_j$ be the new thresholds. By $\max\{a, b\}$ let us denote the element of the set $\{a, b\}$, that has the bigger weight (relative to w). We know that $w(\max\{2i-1, 2i\}) \geq r_1/2 > 0$ for each type i. Consider losing coalition $\{\max\{1,2\}, \max\{3,4\}\}$ with one player from the first type and one from the second type. It has weight

$$w(\{\max\{1,2\}, \max\{3,4\}\}) = w(\max\{1,2\}) + w(\max\{3,4\}) \geq \frac{r_1}{2} + \frac{r_1}{2} = r_1.$$

Assume the worst-case scenario, i.e., that $w(\{\max\{1,2\}, \max\{3,4\}\})$ is equal to r_1. Let us then create a new losing coalition $\{\max\{1,2\}, \max\{3,4\}, \max\{5,6\}\}$ by adding a new player from the third type. It is easy to see that

$$r_1 = w(\{\max\{1,2\}, \max\{3,4\}\}) < w(\{\max\{1,2\}, \max\{3,4\}, \max\{5,6\}\}).$$

So the weight of the new coalition is at least r_2. Assume the worst-case scenario again, and make the weight of $\{\max\{1,2\}, \max\{3,4\}, \max\{5,6\}\}$ be equal to r_2. Proceed by adding a player from the next type to the losing coalition in this manner. At the jth step we will have

$$\begin{aligned} r_j =& w(\{\max\{1,2\},\dots,\max\{2j+1,2j+2\}\}) < \\ & w(\{\max\{1,2\},\dots,\max\{2j+1,2j+2\},\max\{2j+3,2j+4\}\}). \end{aligned}$$

The coalition that was constructed last is losing since it does not contain two players from the same type. So it cannot have weight greater then r_j, which it does. This is a contradiction. Thus $G_{k+1,n}$ is not j-rough for any $j < k+1$. $\square$

In all examples above the number of thresholds of a simple game is equal to the cardinality of the largest losing coalition minus the cardinality of the smallest minimal winning coalition plus one. This is not always the case.

Example 3. *Let $G = ([7], W)$ be a simple game with minimal winning coalitions $\{1,2\},\{6,7\},\{3,4,5\}$ and all coalitions of four players except $\{2,3,4,6\}$. This game is not roughly weighted, because we have the following potent certificate of nonweightedness*

$$\begin{aligned} \mathcal{T} =& \{\{1,2\}^7, \{3,4,5\}^9, P; \{2,3,5\}^3, \{2,3,4\}^3, \\ & \{2,3,6\},\{2,3,7\},\{1,3,4\},\{1,3,5\},\{1,4,5\}^6,\emptyset\}. \end{aligned}$$

Let us assign weight 0 to the third player and $\frac{1}{2}$ to everyone else. Then the following hold:

- $w(\{1,2\}) = w(\{6,7\}) = w(\{3,4,5\}) = 1$ *and* $w(\{2,3,4,6\}) = \frac{3}{2}$.
- *If X is winning coalition with four or more players, then $w(X) \geq \frac{3}{2}$.*
- *If X is losing coalition with three players, then $w(X) \in \{1, \frac{3}{2}\}$.*
- *If X is losing coalition with fewer than three players, then $w(X) \leq 1$.*

Thus G is a 2-rough game with thresholds 1 and $\frac{3}{2}$. Note that the third player has weight zero but he is not a dummy.

5 $\mathcal{C}$-hierarchy

Let us consider another extension of the idea of rough weightedness. This time we will use a threshold interval instead of a single threshold or (as in $\mathcal{B}$-hierarchy) a collection of threshold points. It is convenient to "normalize" the weights so that the left end of our threshold interval is 1. We do not lose any generality by doing this.

Definition 8. *We say that a simple game $G = (P, W)$ is in the class $\mathcal{C}_\alpha$, $\alpha \in \mathbb{R}^{\geq 1}$, if there exists a weight function $w\colon P \to \mathbb{R}^{\geq 0}$ such that for $X \in 2^P$ the condition $w(X) > \alpha$ implies that X is winning, and $w(X) < 1$ implies X is losing. Games from $\mathcal{C}_\alpha$ will be sometimes called rough$_\alpha$.*

The roughly weighted games with nonzero quota form the class $\mathcal{C}_1$. From Example 2 we can conclude that the Fano game is in $\mathcal{C}_{4/3}$ (by giving each player weight 1/3). We also note that adding or deleting a passer does not change the class of the game.

Definition 9. *We say that a game G is* critical *for $\mathcal{C}_\alpha$ if it belongs to $\mathcal{C}_\alpha$ but does not belong to any $\mathcal{C}_\beta$ with $\beta < \alpha$.*

It is clear that if $\alpha \leq \beta$, then $\mathcal{C}_\alpha \subseteq \mathcal{C}_\beta$. However, we can show more.

Proposition 3. *Let c and d be natural numbers with $1 < d < c$. Then there is a simple game G that is rough$_{c/d}$, but that for each $\alpha < c/d$ is not rough$_\alpha$.*

Proof. Define a game $G = (P, W)$, where $P = [cd]$. Similarly to the proof of Theorem 4 we have c types of players with d players in each type and the different types do not intersect. Winning coalitions are sets with more than $c+1$ players and also sets having at least d players from the same type. By i_j we will denote the ith player of jth type.

If we assign weight $1/d$ to each player, then the lightest winning coalition (d players from the same type) has weight 1 and the heaviest losing coalition has weight c/d. Thus G belongs to $\mathcal{C}_{c/d}$.

Let us show that G is not rough$_\alpha$ for any $\alpha < c/d$. Suppose G is rough$_\alpha$ relative to a weight function w. Let $\max\{1_j, \ldots, d_j\}$ be the element of the set $\{1_j, \ldots, d_j\}$ that has the biggest weight relative to w.

For any type j we know that $w(\max\{1_j, 2_j, \ldots, d_j\}) \geq \frac{1}{d}$. The coalition

$$Y = \{\max\{1_1, \ldots, d_1\}, \ldots, \max\{1_c, \ldots, d_c\}\}$$

is losing by definition. Moreover, it has weight $w(Y) \geq c/d$. So c/d is the smallest number that can be taken as α so that G is rough$_\alpha$. □

Theorem 5. *For each $1 \leq \alpha < \beta$, it holds that $\mathcal{C}_\alpha \subsetneq \mathcal{C}_\beta$.*

Proof. We know that $\mathcal{C}_\alpha \subseteq \mathcal{C}_\beta$. If β is a rational number, then by Proposition 3 there exists a game G that is rough$_\beta$ but is not rough$_\alpha$. If β is an irrational number, then choose a rational number r, such that $\alpha < r < \beta$. By Proposition 3 there exists a game G that is rough$_r$ but is not rough$_\alpha$. So $\mathcal{C}_\alpha \subsetneq \mathcal{C}_r$. All that remains to notice is that $\mathcal{C}_r \subseteq \mathcal{C}_\beta$. □

Theorem 6. *Let G be a simple game that is not roughly weighted and is critical for $\mathcal{C}_a$. Suppose G also belongs to $\mathcal{A}_q$ for some $0 < q < \frac{1}{2}$. Then*

$$a \geq \frac{1-q}{1-2q}.$$

Proof. Obviously we can assume that q is rational. Since G is in $\mathcal{A}_q$, it possesses a certificate of nonweightedness $\mathcal{T}$ of the kind

$$\mathcal{T} = (X_1, \ldots, X_t, P^s; Y_1, \ldots, Y_t, \emptyset^s).$$

Suppose we have a weight function $w\colon P \to \mathbb{R}^{\geq 0}$ instantiating $G \in \mathcal{C}_\alpha$. Then since $w(X_i) \geq 1$ and $w(P) \geq a$, we have

$$w(X_1) + \cdots + w(X_t) + sw(P) \geq t + sa. \tag{10}$$

On the other hand, $w(Y_i) \leq a$ and

$$w(Y_1) + \cdots + w(Y_t) \leq ta. \tag{11}$$

From these two inequalities we get $t + sa \leq ta$ or $a \geq \frac{t}{t-s}$. Since $q = \frac{s}{t+s}$ we obtain $a \geq \frac{1-q}{1-2q}$, which proves the theorem. □

6 Degrees of Roughness of Games with Small Number of Players

First, we will derive bounds on the largest number $s(n)$ of the spectre $\mathrm{Spec}(n)$.

Theorem 7. $\frac{1}{2}\left\lfloor\frac{n}{2}\right\rfloor \le s(n) \le \frac{n-2}{2}$.

Proof. Let G be a game with n players. Without loss of generality we can assume that G doesn't contain passers. Moreover the maximal value of $s(n)$ is achieved on games that are not roughly weighted. By Proposition 1 the biggest losing coalition contains at most $n-2$ players and the smallest winning coalition has at least two players. If we assign weight $\frac{1}{2}$ to every player, then G is in $\mathcal{C}_{(n-2)/2}$.

We can use a game similar to the one from Theorem 4 to prove the lower bound. Suppose our game has n players. If n is odd, then one player will be a dummy. The remaining $2\left\lfloor\frac{n}{2}\right\rfloor$ players will be divided into $\left\lfloor\frac{n}{2}\right\rfloor$ pairs: $\{1,2\}$, $\{2,3\}$, $\ldots$, $\{m-1,m\}$, where $m = \left\lfloor\frac{n}{2}\right\rfloor$. These pairs are declared minimal winning coalitions. Given any weight function w we have $w(\max\{2i-1,2i\}) \ge \frac{1}{2}$ for each i. Then

$$w(\{\max\{1,2\},\ldots,\max\{m-1,m\}) \ge \frac{m}{2},$$

while this coalition is losing. So $s(n) \ge m/2$ which proves the lower bound. □

Now let us calculate the spectra for $n \le 6$. By Theorem 2 all games with four players are roughly weighted. Since we may assume that the game does not have passers we may assume that the quota is nonzero. Hence we have $\mathrm{Spec}(4) = \{1\}$.

Let $G = ([n], W)$ be a simple game. The problem of finding the smallest α such that $G \in \mathcal{C}_\alpha$ is a linear programming question. Indeed, let $W^{\min}$ and $L^{\max}$ be the set of minimal winning coalitions and the set of maximal losing coalitions, respectively. We need to find the minimum α such that the following system of linear inequalities is consistent:

$$\begin{cases} w(X) \ge 1 & \text{for } X \in W^{\min}, \\ w(Y) \le \alpha & \text{for } Y \in L^{\max}. \end{cases}$$

This is equivalent to the following optimization problem:

Minimize: w_{n+1}.
Subject to: $\sum_{i\in X} w_i \ge 1$ and $\sum_{i\in Y} w_i - w_{n+1} \le 0;\ X \in W^{min}, Y \in L^{max}$.

Theorem 8. *Spec*$(5) = \left\{1, \frac{6}{5}, \frac{7}{6}, \frac{8}{7}, \frac{9}{8}\right\}$.

Proof. Let G be a critical game with five players. If G has a passer, then as was noted, the passer can be deleted without changing the class of G, hence $G \in \mathcal{C}_1$. If G has no passers and does not belong to $\mathcal{C}_1$, then it is not roughly weighted. By Theorem 2 each game that is not roughly weighted is not strong (recall Definition 2) and is not proper. Thus we have a winning coalition X such that X^c is also winning and a losing coalition Y such that Y^c is also losing.

By Proposition 1 we may assume that the cardinalities of both X and Y are 2. Without loss of generality we assume that $X = \{1,2\}$ and $X^c = \{3,4,5\}$. Note that Y cannot be contained in X^c as otherwise Y^c contains X and is not losing. So without loss of generality we assume that $Y = \{1,5\}, Y^c = \{2,3,4\}$.

We have two levels of as yet unclassified coalitions, which can be set either losing or winning:

$$\begin{aligned} &\text{level 1} : \{1,3,4\},\{1,3,5\},\{1,4,5\},\{2,3,5\},\{2,4,5\}, \\ &\text{level 2} : \{1,3\},\{1,4\},\{2,5\},\{3,5\},\{4,5\}. \end{aligned}$$

We wrote Maple code using the "LPSolve" command. First we choose losing coalitions on level 1 and delete all subsets of them from level 2. We add every unclassified coalition

α	Minimal winning coalitions and maximal losing coalitions	Weight representation
$\frac{9}{8}$	$W^{min} = \{\{1,2\},\{1,3,5\},\{1,4,5\},\{3,4,5\}\}$, $L^{max} = \{\{1,5\},\{1,3,4\},\{2,3,4\},\{2,3,5\},\{2,4,5\}\}$	$w_1 = \frac{5}{8}, w_2 = \frac{3}{8}, w_5 = \frac{4}{8}$, $w_3 = w_4 = \frac{2}{8}$
$\frac{8}{7}$	$W^{min} = \{\{1,2\},\{2,5\},\{1,3,4\},\{3,4,5\}\}$, $L^{max} = \{\{1,3,5\},\{1,4,5\},\{2,3,4\}\}$	$w_1 = w_5 = \frac{3}{7}, w_2 = \frac{4}{7}$, $w_3 = w_4 = \frac{2}{7}$
$\frac{7}{6}$	$W^{min} = \{\{1,2\},\{1,4,5\},\{3,4,5\}\}$, $L^{max} = \{\{1,3,4\},\{1,3,5\},\{2,3,4\},\{2,3,5\},\{2,4,5\}\}$	$w_1 = w_2 = \frac{3}{6}$, $w_3 = w_4 = w_5 = \frac{2}{6}$
$\frac{6}{5}$	$W^{min} = \{\{1,2\},\{1,3\},\{1,4\},\{2,5\},\{3,5\},\{4,5\}\}$, $L^{max} = \{\{1,5\},\{2,3,4\}\}$	$w_1 = w_5 = \frac{3}{5}$, $w_2 = w_3 = w_4 = \frac{2}{5}$

Table 1: Examples of critical simple games for every number of 5th spectrum

from level 1 to winning coalitions. After that we choose losing coalitions on level 2. We run through all possible combinations of losing coalitions on both levels and solve the respective linear programming problems.

The results of these calculations are displayed in Table 1. □

Theorem 9. *The 6th spectrum Spec*(6) *is the set*

$$\left\{1, \frac{3}{2}, \frac{4}{3}, \frac{5}{4}, \frac{6}{5}, \frac{7}{6}, \frac{8}{7}, \frac{9}{7}, \frac{9}{8}, \frac{10}{9}, \frac{11}{9}, \frac{11}{10}, \frac{12}{11}, \frac{13}{10}, \frac{13}{11}, \frac{13}{12}, \frac{14}{11}, \frac{14}{13}, \frac{15}{13}, \frac{15}{14}, \frac{16}{13}, \frac{16}{15}, \frac{17}{13}, \frac{17}{14}, \frac{17}{15}, \frac{17}{16}, \frac{18}{17}\right\}.$$

Proof. Is omitted due to lack of space. The code and the list of critical games are available from the authors. □

7 Conclusion and Further Research

Economics has studied extensively weighted majority games. This class was previously extended to the class of roughly weighted games [9, 4]. However, many games are not even roughly weighted and some of these games are important both for theory and applications. In this paper we introduce three hierarchies, each of which partitions the class of games without rough weights according to some parameter that can be viewed as capturing some resource - either a measure of our flexibility on the size and structure of the tie-breaking region or allowing certain types of certificates of nonweightedness. It is important to look for further connections between the classes of the three hierarchies, and we commend that direction to the interested reader.

In this paper we studied only the $\mathcal{C}$-spectrum here. Some interesting questons about this spectrum still remain, especially the bounds for $s(n)$ are of considerable interest. It is interesting to study both the $\mathcal{A}$-spectrum and $\mathcal{B}$-spectrum as well.

Acknowledgments

During a talk on roughly weighted games at the March 2010 Dagstuhl Seminar on Computational Foundations of Social Choice, Bill Zwicker asked whether those games were merely the start of some hierarchy. That question motivated this paper, and we are deeply grateful to Bill both for that question and for encouragement and helpful conversations. We are also very grateful to Edith Elkind for helpful conversations.

Hemaspaandra was partly supported by NSF grants CCF-0426761 and CCF-0915792 and a Friedrich Wilhelm Bessel Research Award from the Alexander von Humboldt Foundation. Gvozdeva and Slinko were supported by the University of Auckland Faculty of Science Research Development Fund 3624495/9844.

References

[1] A. Beimel, T. Tassa, and E. Weinreb. Characterizing Ideal Weighted Threshold Secret Sharing. In: *Theory of Cryptography.* Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Volume 3378/2005: 600–619, 2005

[2] G.R. Blakley. Safeguarding cryptographic keys. *Proceedings of the National Computer Conference* 48: 313–317, 1979.

[3] C.C. Elgot. Truth Functions Realizable by Single Threshold Organs, *Proc. of the Second Annual Symposium on Switching Circuit Theory and Logical Design*, 225–245, 1961.

[4] T. Gvozdeva and A. Slinko. Weighted and Roughly Weighted Simple Games, Mathematical Social Sciences. to appear.

[5] A.D. Korshunov. Monotone Boolean functions. *Russian Math. Surveys* 58(5): 929–1001, 2003.

[6] S. Muroga. *Threshold logic and Its Applications.* Wiley Interscience, NY, 1971.

[7] M. Richardson. On finite projective games. *Proceedings of the American Mathematical Society* 7: 458–465, 1956.

[8] A.D. Taylor and W.S. Zwicker. A characterization of weighted voting. *Proceedings of the American Mathematical Society* 115: 1089–1094, 1992.

[9] A.D. Taylor and W.S. Zwicker. *Simple games.* Princeton University Press. Princeton. NJ, 1999.

[10] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior. Third edition.* Princeton University Press. Princeton, NJ, 1953.

[11] L.S. Shapley. Simple Games: An Outline of the Descriptive Theory. *Behavioral Science* 7: 59–66, 1962.

Tatyana Gvozdeva
Department of Mathematics
Private Bag 92019, Auckland Mail Centre
University of Auckland
Auckland 1142, NEW ZEALAND
Email: `t.gvozdeva@math.auckland.ac.nz`

Lane A. Hemaspaandra
Department of Computer Science
University of Rochester
Rochester, NY 14627-0226, USA
URL: `www.cs.rochester.edu/u/lane/`

Arkadii Slinko
Department of Mathematics
Private Bag 92019, Auckland Mail Centre
University of Auckland
Auckland 1142, NEW ZEALAND
Email: `slinko@math.auckland.ac.nz`

Optimal Partitions in Additively Separable Hedonic Games[1]

Haris Aziz, Felix Brandt, and Hans Georg Seedig

Abstract

We conduct a computational analysis of partitions in additively separable hedonic games that satisfy standard criteria of fairness and optimality. We show that computing a partition with maximum egalitarian or utilitarian social welfare is NP-hard in the strong sense whereas a Pareto optimal partition can be computed in polynomial time when preferences are strict. Perhaps surprisingly, *checking* whether a given partition is Pareto optimal is coNP-complete in the strong sense, even when preferences are symmetric and strict. We also show that checking whether there exists a partition which is both Pareto optimal and envy-free is Σ_2^p-complete. Furthermore, checking whether there exists a partition which is both envy-free and Nash stable is NP-complete when preferences are symmetric.

1 Introduction

Ever since the publication of von Neumann and Morgenstern's *Theory of Games and Economic Behavior* in 1944, coalitions have played a central role within game theory. The crucial questions in coalitional game theory are which coalitions can be expected to form and how the members of coalitions should divide the proceeds of their cooperation. Traditionally the focus has been on the latter issue, which led to the formulation and analysis of concepts such as the core, the Shapley value, or the bargaining set. Which coalitions are likely to form is commonly assumed to be settled exogenously, either by explicitly specifying the coalition structure, a partition of the players in disjoint coalitions, or, implicitly, by assuming that larger coalitions can invariably guarantee better outcomes to its members than smaller ones and that, as a consequence, the grand coalition of all players will eventually form.

The two questions, however, are clearly interdependent: the individual players' payoffs depend on the coalitions that form just as much as the formation of coalitions depends on how the payoffs are distributed.

Coalition formation games, as introduced by Drèze and Greenberg (1980), provide a simple but versatile formal model that allows one to focus on coalition formation as such. In many situations it is natural to assume that a player's appreciation of a coalition structure only depends on the coalition he is a member of and not on how the remaining players are grouped. Initiated by Banerjee et al. (2001) and Bogomolnaia and Jackson (2002), much of the work on coalition formation now concentrates on these so-called *hedonic games*.

The main focus in hedonic games has been on notions of *stability* for coalition structures such as Nash stability, individual stability, contractual individual stability, or core stability and characterizing conditions under which they are guaranteed to be non-empty (see, e.g., Bogomolnaia and Jackson, 2002; Burani and Zwicker, 2003). The most prominent examples of hedonic games are two-sided matching games in which only coalitions of size two are admissible (Roth and Sotomayor, 1990).

[1]A preliminary version of this work was invited for presentation in the session 'Cooperative Games and Combinatorial Optimization' at the 24th European Conference on Operational Research (EURO 2010) in Lisbon. This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR-2312/6-1 (within the European Science Foundation's EUROCORES program LogICCC) and BR 2312/7-1.

General coalition formation games have also received attention from the artificial intelligence community, where the focus has generally been on computing partitions that give rise to the greatest social welfare (see, e.g., Sandholm et al., 1999). The computational complexity of hedonic games has been investigated with a focus on the complexity of computing stable partitions for different models of hedonic games (Ballester, 2004; Dimitrov et al., 2006; Cechlárová, 2008). We refer to Hajduková (2006) for a critical overview.

Among hedonic games, *additively separable hedonic games (ASHGs)* are a particularly natural and succinct representation in which each player has a value for any other player and the value of a coalition to a particular player is computed by simply adding his values of the players in his coalition.

Additive separability satisfies a number of desirable axiomatic properties (Barberà et al., 2004). ASHGs are the non-transferable utility generalization of *graph games* studied by Deng and Papadimitriou (1994). Sung and Dimitrov (2010) showed that for ASHGs, checking whether a core stable, strict-core stable, Nash stable, or individually stable partition exists is NP-hard. Dimitrov et al. (2006) obtained positive algorithmic results for subclasses of additively separable hedonic games in which each player divides other players into friends and enemies. Branzei and Larson (2009) examined the tradeoff between stability and social welfare in ASHGs.

Contribution In this paper, we analyze concepts from fair division in the context of coalition formation games. We present the first systematic examination of the complexity of computing and verifying optimal partitions of hedonic games, specifically ASHGs. We examine various standard criteria from the social sciences: *Pareto optimality*, *utilitarian social welfare*, *egalitarian social welfare* and *envy-freeness* (see, e.g., Moulin, 1988).

In Section 3, we show that computing a partition with maximum egalitarian social welfare is NP-hard. Similarly, computing a partition with maximum utilitarian social welfare is NP-hard in the strong sense even if preferences are symmetric and strict.

In Section 4, the complexity of Pareto optimality is studied. We prove that checking whether a given partition is Pareto optimal is coNP-complete in the strong sense even for strict and symmetric preferences. By contrast, we present a polynomial-time algorithm for computing a Pareto optimal partition when preferences are strict. Thus, we identify a natural problem in coalitional game theory where verifying a possible solution is presumably harder than actually finding one.[2] Our computational hardness results imply computational hardness of equivalent problems for *hedonic coalition nets* (Elkind and Wooldridge, 2009).

In Section 5, we consider complexity questions regarding envy-free partitions. We show that checking whether there exists a partition which is both Pareto optimal and envy-free is Σ_2^p-complete. We present an example which exemplifies the tradeoff between satisfying stability (such as Nash stability) and envy-freeness and use the example to prove that checking whether there exists a partition which is both envy-free and Nash stable is NP-complete even when preferences are symmetric.

Our computational hardness results imply computational hardness of equivalent problems for *hedonic coalition nets* (Elkind and Wooldridge, 2009).

2 Preliminaries

In this section, we provide the terminology and notation required for our results.

[2]This is also the case for an unrelated problem in social choice theory (Hudry, 2004).

2.1 Hedonic games

A *hedonic coalition formation game* is a pair $(N, \mathcal{P})$ where N is a set of players and $\mathcal{P}$ is a *preference profile* which specifies for each player $i \in N$ the preference relation $\succsim_i$, a reflexive, complete and transitive binary relation on set $\mathcal{N}_i = \{S \subseteq N \mid i \in S\}$.

The statement $S \succ_i T$ means that i strictly prefers S over T. Also $S \sim_i T$ means that i is indifferent between coalitions S and T. A *partition* π is a partition of players N into disjoint coalitions. By $\pi(i)$, we denote the coalition in π which includes player i.

A game $(N, \mathcal{P})$ is *separable* if for any player $i \in N$ and any coalition $S \in \mathcal{N}_i$ and for any player j not in S we have the following: $S \cup \{j\} \succ_i S$ if and only if $\{i, j\} \succ_i \{i\}$; $S \cup \{j\} \prec_i S$ if and only if $\{i, j\} \prec_i \{i\}$; and $S \cup \{j\} \sim_i S$ if and only if $\{i, j\} \sim_i \{i\}$.

We consider utility-based models rather than purely ordinal models. In *additively separable preferences*, a player i gets value $v_i(j)$ for player j being in the same coalition as i and if i is in coalition $S \in \mathcal{N}_i$, then i gets utility $\sum_{j \in S \setminus \{i\}} v_i(j)$.

A game $(N, \mathcal{P})$ is *additively separable* if for each player $i \in N$, there is a utility function $v_i : N \to \mathbb{R}$ such that $v_i(i) = 0$ and for coalitions $S, T \in \mathcal{N}_i$, $S \succsim_i T$ if and only if $\sum_{j \in S} v_i(j) \geq \sum_{j \in T} v_i(j)$.

A preference profile is *symmetric* if $v_i(j) = v_j(i)$ for any two players $i, j \in N$ and is *strict* if $v_i(j) \neq 0$ for all $i, j \in N$ such that $i \neq j$. We consider ASHGs (additively separable hedonic games) in this paper. Unless mentioned otherwise, all our results are for ASHGs. For any player i, let $F(i) = \{j \mid v_i(j) > 0\}$ be the set of players which i strictly likes. Similarly, let $E(i) = \{j \mid v_i(j) < 0\}$ be the set of players which i strictly dislikes.

2.2 Fair and optimal partitions

In this section, we formulate concepts from the social sciences especially the economics and the fair division literature for the context of hedonic games. For a utility-based hedonic game $(N, \mathcal{P})$ and partition π, we will denote the utility of player $i \in N$ by $u_\pi(i)$. The different notions of fair or optimal partitions are defined as follows.[3]

1. The *utilitarian social welfare* of a partition is defined as the sum of individual utilities of the players: $u_{ut}(\pi) = \sum_{i \in N} u_\pi(i)$. A *maximum utilitarian partition* maximizes the utilitarian social welfare.

2. The *elitist social welfare* is given by the utility of the player that is best off: $u_{el}(\pi) = \max\{u_\pi(i) \mid i \in N\}$. A *maximum elitist partition* maximizes the utilitarian social welfare.

3. The *egalitarian social welfare* is given by the utility of the agent that is worst off: $u_{eg}(\pi) = \min\{u_\pi(i) \mid i \in N\}$. A *maximum egalitarian partition* maximizes the egalitarian social welfare.

4. An *ordered utility vector* associated with partition π is given by $(u_\pi(p(1)), \ldots, u_\pi(p(n)))$ where p is a permutation of players such that $u_\pi(p(i)) \leq u_\pi(p(j))$ where $p(i) \leq p(j)$. Then a partition π with the *maximum leximin social welfare* is one which has lexicographically the greatest ordered utility vector. We refer to π as a *maximum leximin partition.*

5. A partition π of N is *Pareto optimal* if there exists no partition π' of N which *Pareto dominates* π, that is for all $i \in N$, $\pi'(i) \succsim_i \pi(i)$ and there exists at least one player $j \in N$ such that $j \in N$, $\pi'(j) \succ_j \pi(j)$.

[3]All welfare notions considered in this paper (utilitarian, elitist, egalitarian, and leximin) are based on the interpersonal comparison of utilities. Whether this assumption can reasonably be made is debatable.

6. *Envy-freeness* is a notion of fairness. In an *envy-free* partition, no player has an incentive to replace another player.

For the sake of brevity, we will consider all the notions described above as optimality criteria although envy-freeness is more concerned with fairness. We consider the following computational problems with respect to the optimality criteria defined above.

OPTIMALITY: Given $(N, \mathcal{P})$ and a partition π of N, is π optimal?
EXISTENCE: Does an optimal partition for a given $(N, \mathcal{P})$ exist?
SEARCH: If an optimal partition for a given $(N, \mathcal{P})$ exists, find one.

EXISTENCE is trivially true for all criteria of optimality concepts. By the definitions, it follows that there exist partitions which satisfy maximum utilitarian social welfare, elitist social welfare, egalitarian social welfare and leximin ordering respectively. The partition consisting of the grand coalition and the partition of singletons satisfy envy-freeness. During our computational analysis, we will assume familiarity of the reader with basic computational complexity classes. We recall that a problem is said to be NP-hard in the strong sense if it remains so even when its numerical parameters are bounded by a polynomial in the length of the input.

3 Complexity of maximizing social welfare

In this section, we examine the complexity of maximizing social welfare in ASHGs. Our first result is the following one.

Theorem 1. *Computing a maximum utilitarian partition is NP-hard in the strong sense even with symmetric and strict preferences.*

Proof. We prove Theorem 1 by a reduction from the MAXCUT problem. Before defining the MAXCUT problem, recall that a *cut* is a partition of the vertices of a graph into two disjoint subsets. The *cut-set* of the cut is the set of edges whose end points are in different subsets of the partition. In a weighted graph, the *weight of the cut* is the sum of the weights of the edges in the cut-set. Then, MAXCUT is the following problem:

MAXCUT
INSTANCE: An undirected weighted graph $G = (V, E)$ with a weight function $w : E \to \mathbb{R}^+$ and an integer k.
QUESTION: Does there exist a cut of weight at least k in G?

We present a polynomial-time reduction from MAXCUT to UTILSEARCH, the problem of computing a maximum utilitarian partition. Consider an instance I of MAXCUT with a connected undirected graph $G = (V, E)$ and positive weights $w(i, j)$ for each edge (i, j). Let $W = \sum_{(i,j)\in E} w(i, j)$. We show that if there is there a polynomial-time algorithm for computing a maximum utilitarian social welfare partition, then we have a polynomial-time algorithm for MAXCUT.

Consider the following method which in polynomial time reduces I to an instance I' of UTILSEARCH. I' consists of $|V| + 2$ players $N = \{m_1, \ldots, m_{|V|}, s_1, s_2\}$. For any two players m_i and m_j, $v_{m_i}(m_j) = v_{m_j}(m_i) = -w(i, j)$. For any player m_i and player s_j, $v_{m_i}(s_j) = v_{s_j}(m_i) = W$. Also $v_{s_1}(s_2) = v_{s_2}(s_1) = -W(|V| + 1)$.

We first prove that partition π^* with maximum utilitarian social welfare u^* consists of exactly two coalitions with s_1 and s_2 in different coalitions. We do so by proving two claims. The first claim is that every player m_i is either in a coalition with s_1 or s_2. Assume this is not true and there exists a partition π such that $u_{ut}(\pi) = u^*$ and m_i is not in the same coalition with s_1 or s_2. Then, if m_i joins $\pi(s_1)$, $u_{ut}(\pi)$ increases at least by $2W$ and it decreases by at most $2\sum_{j\in N} w(i,j) < 2W$. Therefore, $u_{ut}(\pi)$ increases which is a contradiction. The second claim is that s_1 and s_2 are in different coalitions in π^*. Assume this is not true and there exists a partition π with utilitarian social welfare u^* such that s_1 and s_2 are together in a coalition. Then the welfare of π can be increased by at least $2(|V|+1)(W) - 2|V|W = 2W$ if s_2 breaks up and forms a singleton coalition. This is a contradiction.

We are now ready to present the reduction. Assume there exists a polynomial-time algorithm which computes a feasible maximum utilitarian social welfare partition π. From the two claims above, we can assume that partition π has two coalitions with s_1 and s_2 in different coalitions. Then, $u_{ut}(\pi) = 2(X + \sum_{m_i \notin \pi(m_j)} -v_{m_i}(m_j))$ where $X = -W + (|V|+1)W \geq 2W$ if $|V| \geq 2$. We also know that $\sum_{m_i \notin \pi(m_j)} -v_{m_i}(m_j)) < W$. We can obtain a cut (A,B) from π where $A = \{i \mid m_i \in \pi(s_1)\}$ and $B = \{i \mid m_i \in \pi(s_2)\}$. Let the weight of the cut (A,B) be c. We know that $c \leq c^*$ where c^* is the weight of the maxcut for instance I. It is now shown that (A,B) is a maxcut if and only if $u_{ut}(\pi) = u^*$. Assume $u_{ut}(\pi) = u^*$ but (A,B) is a not a maxcut. In that case there exists a maxcut (C,D) such that $\sum_{i\in C, j\in D} w(i,j) > \sum_{i\in A, j\in B} w(i,j)$. Therefore, there exists a partition $\pi' = \{\{s_1 \cup \{m_i \mid i \in A\}\}, \{s_2 \cup \{m_i \mid i \in B\}\}\}$ where $u_{ut}(\pi') = 2(X + \sum_{m_i \notin \pi'(m_j)} -v_{m_i}(m_j)) > 2(X + \sum_{m_i \notin \pi(m_j)} -v_{m_i}(m_j))$. This is a contradiction as $u_{ut}(\pi) = u^*$.

Now assume that (A,B) is a maxcut but $u_{ut}(\pi) < u^*$. Then there exists another partition π^* such that $u_{ut}(\pi') = 2(X + \sum_{m_i \notin \pi'(m_j)} -v_{m_i}(m_j)) = u^*$. Therefore, the graph cut corresponding to π^* has a bigger maxcut value than (A,B) which is a contradiction. □

Computing a maximum elitist partition is much easier.

Proposition 1. *There exists a polynomial-time algorithm to compute a maximum elitist partition.*

Proof. Recall that for any player i, $F(i) = \{j \mid v_i(j) > 0\}$. Let $f(i) = \sum_{j\in F(i)} v_i(j)$. Both $F(i)$ and $f(i)$ can be computed in linear time. Let $k \in N$ be the player such that $f(k) \geq f(i)$ for all $i \in N$. Then $\pi = \{\{\{k\} \cup F(k)\}, N \setminus \{\{k\} \cup F(k)\}\}$ is a partition which maximizes the elitist social welfare. □

As a corollary, we can verify whether a partition π has maximum elitist social welfare by computing a partition π^* with maximum elitist social welfare and comparing $u_{el}(\pi)$ with $u_{el}(\pi^*)$. Just like maximizing the utilitarian social welfare, maximizing the egalitarian social welfare is hard:

Theorem 2. *Computing a maximum egalitarian partition is NP-hard in the strong sense.*

Proof. We provide a polynomial-time reduction from the following NP-hard problem (Woeginger, 1997):

MaxMinMachineCompletionTime
INSTANCE: A set of m identical machines $M = \{M_1, \ldots, M_m\}$, a set of n independent jobs $J = \{J_1, \ldots, J_n\}$ where job J_i has processing time p_i.
OUTPUT: Allot jobs to the machines such that the minimum processing time (without machine idle times) of all machines is maximized.

Let I be an instance of MAXMINMACHINECOMPLETIONTIME and let $P = \sum_{i=1}^{n} p_i$. From I we construct an instance I' of EGALSEARCH. The ASHG for instance I' consists of $N = \{i \mid M_i \in M\} \cup \{s_i \mid J_i \in J\}$ and the preferences of the players are as follows: for all $i = 1, \ldots m$ and all $j = 1, \ldots, n$ let $v_i(s_j) = p_j$ and $v_{s_j}(i) = P$. Also, for $1 \leq i, i' \leq m, i \neq i'$ let $v_i(i') = -(P+1)$ and for $1 \leq j, j' \leq n, j \neq j'$ let $v_{s_j}(v_{s_{j'}}) = 0$. Each player i corresponds to machine M_i and each player s_j corresponds to job J_j.

Let π be the partition which maximizes $u_{eg}(\pi)$. We show that players $1, \ldots, m$ are in separate coalitions and each player s_j is in $\pi(i)$ for some $1 \leq i \leq m$. We do so by proving two claims. The first claim is that for $i, j \in \{1, \ldots m\}$ such that $i \neq j$, we have that $i \notin \pi(j)$. Assume there exist exactly two players i and j for which this is not the case. Then we know that $u_\pi(i) = -(P+1) + \sum_{s_j \in \pi(i)} p_j$. Since $\sum_{s_j \in \pi(i)} p_j \leq P$, we know that $u_\pi(i) = u_\pi(j) < 0$, $u_\pi(a) \geq 0$ for all $a \in N \setminus \{i, j\}$ and thus $u_{eg}(\pi) < 0$. However, if i deviates and forms a a singleton coalition in new partition π', then $u_{\pi'}(i) = 0$ and $u_{\pi'}(j) \geq 0$ and the utility of other players has not decreased. Therefore, $u_{eg}(\pi') \geq 0$ which is a contradiction.

The second claim is that each player s_j is in a coalition with a player i. Assume this was not the case so that there exists at least one such player s_j. Since we already know that all is are in separate coalitions, then $u_\pi(a) > 0$ for all $a \in N \setminus \{s_j\}$ and $u_{eg}(\pi) = u_\pi(s_j) = 0$. Then s_j can deviate and join $\pi(i)$ for any $1 \leq i \leq m$ to form a new partition π'. By that, the utility of no player decreases and $u_{\pi'}(s_j) > 0$. If this is done for all such s_j, we have $u_{eg}(\pi') > 0$ for the new partition π' which is a contradiction.

A job allocation $\mathsf{Alloc}(\pi)$ corresponds to a partition π where s_j is in $\pi(i)$ if job J_j is assigned to M_i for all j and i. Note that the utility $u_\pi(i) = \sum_{s_j \in \pi(i)} v_i(s_j) = \sum_{s_j \in \pi(i)} p_j$ of a player corresponds to the total completion time of all jobs assigned to M_i according to $\mathsf{Alloc}(\pi)$. Let π^* be a maximum egalitarian partition. Assume that there is another partition π' and $\mathsf{Alloc}(\pi')$ induces a strictly greater minimum completion time. We know that $u_{\pi^*}(s_j) = u_{\pi''}(s_j) = P$ for all $1 \leq j \leq n$ and $u_{\pi^*}(i) \leq P$ for all $1 \leq i \leq m$. But then from the assumption we have $u_{eg}(\pi') > u_{eg}(\pi^*)$ which is a contradiction. □

Since a maximum leximin partition is also a maximum egalitarian partition, we have the corollary that computing a partition with maximum leximin social welfare is NP-hard.

4 Complexity of Pareto optimality

We now consider the complexity of computing a Pareto optimal partition. The complexity of Pareto optimality has already been considered in several settings such as house allocation (Abraham et al., 2005). Bouveret and Lang (2008) examined the complexity of Pareto optimal allocations in resource allocation problems. We show that checking whether a partition is Pareto optimal is hard even under severely restricted settings.

Theorem 3. *The problem of checking whether a partition is Pareto optimal is coNP-complete in the strong sense, even if preferences are symmetric and strict.*

Proof. The reduction is from E3C (EXACT-3-COVER) to deciding whether a given partition is Pareto dominated by another partition or not. We recall the E3C problem.

E3C (EXACT-3-COVER):
INSTANCE: A pair (R, S), where $R = \{1, \ldots, r\}$ is a set and S is a collection of subsets of R such that $|R| = 3m$ for some positive integer m and $|s| = 3$ for each $s \in S$.
QUESTION: Is there a sub-collection $S' \subseteq S$ which is a partition of R?

It is known that E3C remains NP-complete even if each $r \in R$ occurs in at most three members of S. Let (R, S) be an instance of E3C where R is a set and S is a collection of

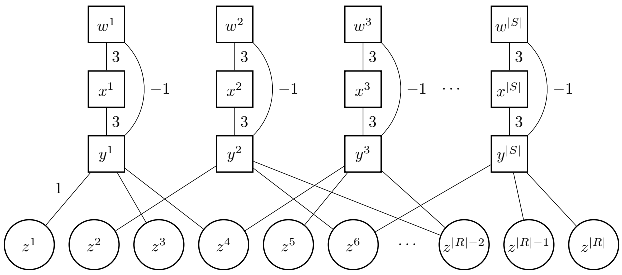


Figure 1: A graph representation of an ASHG derived from an instance of E3C. The (symmetric) utilities are given as edge weights. Some edges and labels are omitted: All edges between any y^s and z^r have weight 1 if $r \in s$. All $z^{r'}, z^{r''}$ with $r' \neq r''$ are connected with weight $\frac{1}{|R|-1}$. All other edges missing in the complete undirected graph have weight -4.

subsets of R such that $|R| = 3m$ for some positive integer m and $|s| = 3$ for each $s \in S$. (R, S) can be reduced to an instance $((N, \mathcal{P}), \pi)$, where $(N, \mathcal{P})$ is an ASHG defined in the following way. Let $N = \{w^s, x^s, y^s \mid s \in S\} \cup \{z^r \mid r \in R\}$. The players preferences are symmetric and strict and are defined as follows:

- $v_{w^s}(x^s) = v_{x^s}(y^s) = 3$ for all $s \in S$
- $v_{y^s}(w^s) = v_{y^s}(w^{s'}) = -1$ for all $s, s' \in S$
- $v_{y^s}(z^r) = 1$ if $r \in s$ and $v_{y^s}(z^r) = -1$ if $r \notin s$ and
- $v_{z^r}(z^{r'}) = 1/(|R| - 1)$ for any $r, r' \in R$
- $v_a(b) = -4$ for any $a, b \in N$ and $a \neq b$ for which $v_a(b)$ is not already defined,

The partition π in the instance $((N, \mathcal{P}), \pi)$ is $\{\{x^s, y^s\}, \{w^s\} \mid s \in S\}\} \cup \{\{z^r \mid r \in R\}\}$. We see that the utilities of the players are as follows: $u_\pi(w^s) = 0$ for all $s \in S$; $u_\pi(x^s) = u_\pi(y^s) = 3$ for all $s \in S$; and $u_\pi(z^r) = 1$ for all $r \in R$.

Assume that there exists $S' \subseteq S$ such that S' is a partition of R. Then we prove that π is not Pareto optimal and there exists another partition π' of N which Pareto dominates π. We form another partition

$$\pi' = \{\{x^s, w^s\} \mid s \in S'\} \cup \{\{y^s, z_i, z_j, z_k\} \mid s \in S' \wedge i, j, k \in s\} \cup \{\{x^s, y^s\}, \{w^s\} \mid s \in (S \backslash S')\}\}.$$

In that case, $u_{\pi'}(w^s) = 3$ for all $s \in S'$; $u_{\pi'}(w^s) = 0$ for all $s \in S \backslash S'$; $u_\pi(x^s) = u_\pi(y^s) = 3$ for all $s \in S$; and $u_\pi(z^r) = 1 + 2/(|R| - 1)$ for all $r \in R$. Whereas the utilities of no player in π' decreases, the utility of some players in π' is more than in π. Since π' Pareto dominates π, π is not Pareto optimal.

We now show that if there exists no $S' \subseteq S$ such that S' is a partition of R, then π is Pareto optimal. We note that -4 is a sufficiently large negative valuation to ensure that if $v_a(b) = v_b(a) = -4$, then $a, b \in N$ cannot be in the same coalition in a Pareto optimal partition. For the sake of contradiction, assume that π is not Pareto optimal and there exists a partition π' which Pareto dominates π. We will see that if there exists a player $i \in N$ such that $u_{\pi'} > u_\pi$, then there exists at least one $j \in N$ such that $u_{\pi'} < u_\pi$. The only players whose utility can increase are $\{x^s \mid s \in S\}$, $\{w^s \mid s \in S\}$ or $\{z^r \mid r \in R\}$. We consider these player classes separately. If the utility of player x^s increases, it can only increase from 3 to 6 so that x^s is in the same coalition as y^s and w^s. However, this means that y^s gets a decreased utility. The utility of y^s can increase or stay the same only if it forms a coalition with some z^rs. However in that case, to satisfy all z^rs, there needs to exist an $S' \subseteq S$ such that S' is a partition of R.

Assume the utility of a player w^s for $s \in S$ increases. This is only possible if w^s is in the same coalition as x^s. Clearly, the coalition formed is $\{w^s, x^s\}$ because coalition $\{w^s, x^s, y^s\}$ brings a utility of 2 to y^s. In that case y^s needs to form a coalition $\{y^s, z_i, z_j, z_k\}$ where $s = \{i, j, k\}$. If y^s forms a coalition $\{y^s, z_i, z_j, z_k\}$, then all players $y^{s'}$ for $s' \in (S \setminus \{s\})$ need to form coalitions of the form $\{y^{s'}, z_{i'}, z_{j'}, z_{k'}\}$ such that $s' = \{i', j', k'\}$. Otherwise, their utility of 3 decreases. This is only possible if there exists a set $S' \subseteq S$ of R such that S' is a partition of R.

Assume that there exists a partition π' that Pareto dominates π and utility of a player $u_{\pi'}(z^r) > u_{\pi}(z^r)$ for some $r \in R$. This is only possible if each z^r forms the coalition of the form $\{z^r, z^{r'}, z^{r''}, y^s\}$ where $s = \{r, r', r''\}$. This can only happen if there exists a set $S' \subseteq S$ of R such that S' is a partition of R. □

The fact that checking whether a partition is Pareto optimal is coNP-complete has no obvious implications on the complexity of *computing* a Pareto optimal partition. In fact we present a polynomial-time algorithm to compute a partition which is Pareto optimal for strict preferences.

Theorem 4. *For strict preferences, a Pareto optimal partition can be computed in polynomial time.*

Proof. We first describe the algorithm. Set RemainingPlayers to N and set i to 1. Take any player $l_i \in$ RemainingPlayers and form a coalition S_i in which players $j \in$ RemainingPlayers such that $v_{l_i}(j) > 0$ are added. Player l_i will be called the *leader* of coalition S_i. Remove S_i from RemainingPlayers. Increment i by 1 and repeat until RemainingPlayers $= \emptyset$. Return $\{S_1, \ldots, S_m\}$.

We now prove the correctness of the algorithm via induction on the number of coalitions formed. The induction hypothesis is: *Consider the kth first formed coalitions $S_1, \ldots, S_k$. Assume, there exists a partition $\pi' \neq \pi$, such that π' Pareto dominates π. Then $S_1, \ldots, S_k \in \pi'$.* Less formally and in other words, the hypothesis can be stated as follows: *Assume that the first k coalitions $S_1, \ldots, S_k$ have formed. Then neither of the following can happen:*

1. *Some players from $S_1, \ldots, S_k$ move out of their respective coalitions and cause a Pareto improvement.*
2. *Some players from $N \setminus \bigcup_{i \in \{1,\ldots,k\}} S_i$ move to players in coalitions $S_1, \ldots, S_k$ and cause a Pareto improvement.*

Base case: Consider the coalition S_1. Then l_1, the leader of S_1 has no incentive to leave. If he leaves with a subset of players in S_1, he can only become less happy. Other players from S_1 cannot leave S_1 because their leaving makes at least one player less happy. The only possibility left is if S_1 joins $B \subseteq (N \setminus S_1)$ to cause a Pareto improvement. We know that this is not possible as player l_1 would be worse off. Similarly, no player j can move from $N \setminus S_1$ and cause a Pareto improvement because l_1 becomes worse off.
Induction step: Assume that the hypothesis is true. Then we prove that the same holds for the formed coalitions $S = S_1, \ldots, S_k, S_{k+1}$. By the hypothesis, we know that player cannot leave coalitions $S_1, \ldots, S_k$ and cause a Pareto improvement and since preferences are strict, no player can move from $N \setminus \bigcup_{i \in \{1,\ldots,k\}} S_i$ move to coalitions in $S_1, \ldots, S_k$ and cause a Pareto improvement as at least one player in S_{k+1} dislike him.

Now consider S_{k+1}. The leader of S_{k+1} is l_{k+1}. We first show that l_{k+1} cannot cause a Pareto improvement by moving to a coalition outside of S_{k+1}. This is clear because l_{k+1} can only lose utility when he leaves coalition S_{k+1} with a subset of or all of the players.

Similarly, other players in S_{k+1} cannot move out of S_{k+1} without decreasing the payoff of some player in S_{k+1}. Similarly, since the preferences are strict, no player can move from $N \setminus \bigcup_{i\in\{1,\dots,k+1\}} S_i$ and cause a Pareto improvement. $\square$

A standard criticism of Pareto optimality is that it can lead to inherently unfair allocations. To address this criticism, the algorithm can be modified to obtain less lopsided partitions. Whenever an arbitrary player is selected to become the 'leader' among the remaining players, choose a player that does not get extremely high elitist social welfare among the remaining players. Nevertheless, even this modified algorithm may output an partition that fails to be individually rational.[4]

Another natural algorithmic question is to check whether it is possible for all players to attain their maximum possible utility at the same time. We observe that this problem can be solved in polynomial time for any separable game. We will omit the details of the algorithm but the general idea behind the algorithm is to build up coalitions and ensure that a player i and $F(i)$, all the player i likes are in the same coalition. While ensuring this, if there is a player j and a player $j' \in E(j)$ (disliked by j), then return 'no.'

5 Complexity of envy-freeness

Envy-freeness is a well-sought criterion in resource allocation, especially *cake cutting*. Lipton et al. (2004) proposed envy-minimization in different ways and examined the complexity of minimizing envy in resource allocation settings. Bogomolnaia and Jackson (2002) mentioned envy-freeness in hedonic games but focused on stability in hedonic games. We already know that envy-freeness can be easily achieved by the partition of singletons.[5] Therefore, in conjunction with envy-freeness, we seek to satisfy other properties such as stability or Pareto optimality. A partition is *Nash stable* is there is no incentive for a player to be deviate to another (possibly empty) coalition. For symmetric ASHGs, it is known that Nash stable partitions always exists and they correspond to partitions for which the utilitarian social welfare is a local optimum (see, e.g., Bogomolnaia and Jackson, 2002). We now show that for symmetric ASHGs, there may not exist any partition which is both envy-free and Nash stable.

Example 1. Consider an ASHG $(N, \mathcal{P})$ where $N = \{1, 2, 3\}$ and $\mathcal{P}$ is defined as follows: $v_1(2) = v_2(1) = 3$, $v_1(3) = v_3(1) = 3$ and $v_2(3) = v_3(2) = -4$. Then there exists no partition which is both envy-free and Nash stable.

We use the game in Example 1 as a gadget to prove the following.[6]

Theorem 5. *For symmetric preferences, checking whether there exists a partition which is both envy-free and Nash stable is NP-complete in the strong sense.*

Proof. The problem is clearly in NP since envy-freeness and Nash stability can be verified in polynomial time. We reduce the problem from E3C. Let (R, S) be an instance of E3C where R is a set and S is a collection of subsets of R such that $|R| = 3m$ for some positive integer m and $|s| = 3$ for each $s \in S$. (R, S) can be reduced to an instance $(N, \mathcal{P})$ where $(N, \mathcal{P})$ is an ASHG defined in the following way. Let $N = \{y^s \mid s \in S\} \cup \{z_1^r, z_2^r, z_3^r \mid r \in R\}$. We set all preferences as symmetric. The players preferences are as follows:

[4] It can be shown that, for general preferences, computing a partition that is Pareto optimal and individually rational at the same time is weakly NP-hard.

[5] The partition of singletons also satisfies individual rationality.

[6] Example 1 and the proof of Theorem 5 also apply to the combination of envy-freeness and individual stability and to that of envy-freeness and contractual individual stability where individual stability and contractual individual stability are variants of Nash stability (Bogomolnaia and Jackson, 2002).

- For all $r \in R$, $v_{z_1^r}(z_2^r) = v_{z_2^r}(z_1^r) = 3$, $v_{z_1^r}(z_3^r) = 3$ and $v_{z_2^r}(z_3^r) = v_{z_3^r}(z_2^r) = -4$.
- For all $s = \{i, j, k\} \in S$, $v_{z_1^i}(z_1^j) = v_{z_1^i}(z_1^k) = v_{z_1^j}(z_1^k) = v_{y^s}(z_1^i) = v_{y^s}(z_1^j) = v_{y^s}(z_1^k) = 1$.
- For all $a, b \in N$ for which valuations have not been defined, $v_a(b) = v_b(a) = -4$

We note that -4 is a sufficiently large negative valuation to ensure that if $v_a(b) = v_b(a) = -4$, then a and b will get negative utility if they are in the same coalition. We show that there exists an envy-free and Nash stable partition for $(N, \mathcal{P})$ if and only if (R, S) is a 'yes' instance of E3C.

Assume that there exists $S' \subseteq S$ such that S' is a partition of R. Then there exists a partition $\pi = \{\{y^s, z_1^i, z_1^j, z_1^k\} \mid s = \{i, j, k\} \in S'\} \cup \{\{z_2^r\}, \{z_3^r\} \mid r \in R\} \cup \{\{s\} \mid s \in S \setminus S'\}$. It is easy to see that partition π is Nash stable and envy-free. Players z_1^r and z_3^r both had an incentive to be with each other when they are singletons. However, each z_1^r now gets utility 3 by being in a coalition with $z_1^{r'}$, $z_1^{r''}$ and y^s where $s = \{r, r', r''\} \in S$. Therefore z_1^r has no incentive to be with z_3^r and z_3^r has no incentive to join $\{z_1^{r'}, z_1^{r'}, z_1^{r''}, y^s\}$ because $v_{z_3^r}(z_1^{r'}) = v_{z_3^r}(z_1^{r''}) = v_{z_3^r}(y^s) = -4$. Similarly, no player is envious of another player.

Assume that there exists no partition $S' \subseteq S$ of R such that S' is a partition of R. Then, there exists at least one $r \in R$ such that z_1^i is not in the coalition of the form $\{z_1^r, z_1^{r'}, z_1^{r''}, y^s\}$ where $s = \{r, r', r''\} \in S$. Then the only individually rational coalitions which z_1^r can form are the following $\{z_1^r\}$, $\{z_1^r, z_3^r\}$, $\{z_1^r, z_2^r\}$ or $\{z_1^r, z_1^{r'}\}$ where $r, r' \in s$ for some $s \in S$. In the first case, z_1^r wants to deviate to $\{z_3^r\}$. In the second case, z_2^r is envious and wants to replace z_3^r. In the third case, z_3^r is envious and wants to replace z_2^r. In the fourth case, z_3^r is envious and wants to replace $z_1^{r'}$. Therefore, there exists no partition which is both Nash stable and envy-free. □

While the existence of a Pareto optimal partition and an envy-free partition is guaranteed, we show that checking whether there exists a partition which is both envy-free and Pareto optimal is hard (Corollary 1). To prove the result, we first define the resource allocation setting. A *resource allocation problem* is a tuple (I, X, w) where I is a set of players (agents), X is the set of indivisible objects and $w : I \times X \to (R)$ is the weight function. A resource allocation $a : I \to 2^X$ is such that for all i and $j \neq i$, $a(i) \cap a(j) = \emptyset$. A resource allocation a dominate a' if and only if 1) for all $a(i) \succsim_i a'(i)$ and 2) there exists i such that $a(i) \succsim_i a'(i)$. A resource allocation is Pareto optimal if it is not dominated by another resource allocation.

Theorem 6. *(Theorem 2, de Keijzer et al. (2009)) The problem ∃-EEF-ADD of checking the existence of an envy-free and Pareto optimal resource allocation is Σ_2^p-complete.*

We can use the result from de Keijzer et al. (2009) to prove the following.

Corollary 1. *Checking whether there exists a partition which is both Pareto optimal and envy-free is Σ_2^p-complete.*

Proof. The problem has a yes instance if there exists an envy-free partition that Pareto dominates every other partition. Therefore the problem is in the complexity class $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$. We now prove that the problem is Σ_2^p-hard. We provide a polynomial-time reduction ∃-EEF-ADD to the problem of checking whether there exists a partition which is both Pareto optimal and envy-free.

Consider an instance (I, X, w) of a resource allocation problem. Let $W = \sum_{i \in I, x_j \in X} |w(i, x_j)|$. The instance (I, X, w) can be reduced to an instance of an ASHG G where $N = I \cup X$ and

- For all $i \in I$, $x_j \in X$, $v_i(x_j) = w(i, x_j)$ and $v_{x_j}(i) = 0$.
- For all x_j, x_k, $v_{x_i}(x_j) = v_{x_j}(x_i) = 0$.
- For all $i, j \in I$, $v_i(j) = v_j(i) = -W|I \cup X|$.

It is clear that for any Pareto optimal partition π, there exist no $i, j \in I \subset N$ such that $i \neq j$ and $j \in \pi(i)$. Assume that this were not the case and there exist $i, j \in I \subset N$ such that $i \neq j$ and $j \in \pi(i)$. Then i and j both get negative value because $\sum_{k\in\pi(i)} v_i(k) = \sum_{k\in(\pi(i)\setminus\{j\})} v_i(k) - W < 0$ and $\sum_{k\in\pi(i)} v_j(k) = \sum_{k\in(\pi(i)\setminus\{i\})} v_j(k) - W < 0$. Then i and j can be separated to form singletons to get another partition π', where the value of every other player $k \in (N \setminus \{i, j\})$ gets the same value while i and j get at least zero value. Therefore there is a one-to-one correspondence between any such partition π and allocation a where $a(i) = \pi(i) \setminus \{i\}$. It now easy to see that π is Pareto optimal and envy-free in G if and only if a is a Pareto optimal and envy-free allocation. □

The results of this section show that, even though envy-freeness can be trivially satisfied on its own, it becomes much more delicate when considered in conjunction with other desirable properties.

6 Conclusions

In this paper, we studied the complexity of partitions in additively separable hedonic games that satisfy standard criteria of fairness and optimality. We showed that computing a partition with maximum egalitarian or utilitarian social welfare is NP-hard in the strong sense whereas a Pareto optimal partition can be computed in polynomial time when preferences are strict. Interestingly, checking whether a given partition is Pareto optimal is coNP-complete even in the restricted setting of strict and symmetric preferences. We also showed that checking the existence of partition which satisfies not only envy-freeness but an additional property like Nash stability or Pareto optimality is computationally hard. The complexity of computing a Pareto optimal partition for ASHGs with general preferences is still open. Since the grand coalition has special significance in coalitional game theory, it would be interesting to study the complexity of checking whether the grand coalition is Pareto optimal. Other directions for future research include approximation algorithms to compute maximum utilitarian or egalitarian social welfare for different representations of hedonic games.

References

D. J. Abraham, K. Cechlárová, D. Manlove, and K. Mehlhorn. Pareto optimality in house allocation problems. In *Proceedings of the 16th International Symposium on Algorithms and Computation*, pages 1163–1175, 2005.

C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.

S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18:135–153, 2001.

S. Barberà, W. Bossert, and P. K. Pattanaik. Ranking sets of objects. In S. Barberà, P. J. Hammond, and C. Seidl, editors, *Handbook of Utility Theory*, volume II, chapter 17, pages 893—977. Kluwer Academic Publishers, 2004.

A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: logical representation and complexity. *Journal of Artificial Intelligence Research*, 32(1):525–564, 2008.

S. Branzei and K. Larson. Coalitional affinity games and the stability gap. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1319–1320, 2009.

N. Burani and W. S. Zwicker. Coalition formation games with separable preferences. *Mathematical Social Sciences*, 45(1):27–52, 2003.

K. Cechlárová. Stable partition problem. In *Encyclopedia of Algorithms*. 2008.

Bart de Keijzer, S. Bouveret, T. Klos, and Y. Zhang. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In *Proceedings of the First International Conference on Algorithmic Decision Theory*, pages 98–110, 2009.

X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 12(2):257–266, 1994.

D. Dimitrov, P. Borm, R. Hendrickx, and S. C. Sung. Simple priorities and core stability in hedonic games. *Social Choice and Welfare*, 26(2):421–433, 2006.

J. H. Drèze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4): 987–1003, 1980.

E. Elkind and M. Wooldridge. Hedonic coalition nets. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 417–424, 2009.

J. Hajduková. Coalition formation games: A survey. *International Game Theory Review*, 8(4): 613–641, 2006.

O. Hudry. A note on "Banks winners in tournaments are difficult to recognize" by G. J. Woeginger. *Social Choice and Welfare*, 23:113–114, 2004.

R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM conference on Electronic commerce(ACM-EC)*, pages 125–131, 2004.

H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

A. Roth and M. A. O. Sotomayor. *Two-Sided Matching: A Study in Game Theoretic Modelling and Analysis*. Cambridge University Press, 1990.

T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1–2):209–238, 1999.

S. C. Sung and D. Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, 2010.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 2nd edition, 1947.

G. J. Woeginger. A polynomial time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.

Haris Aziz, Felix Brandt, and Hans Georg Seedig
Department of Informatics
Technische Universität München
85748 Garching bei München, Germany
{aziz,brandtf,seedigh}@in.tum.de

Fractional Solutions for NTU-Games

Péter Biró[1] and Tamás Fleiner[2]

Abstract

In this paper we survey some applications of Scarf's Lemma. First, we introduce the notion of fractional core for NTU-games, which is always nonempty by the Lemma. Stable allocation is a general solution concept for games where both the players and their possible cooperations can have capacities. We show that the problem of finding a stable allocation, given a finitely generated NTU-game with capacities, is always solvable by a variant of Scarf's Lemma. Finally, we describe the interpretation of these results for matching games.

1 Introduction

Complex social and economic situations can be described as games where the players may cooperate with each other. Most studies in cooperative game theory focus on the issue of how the participants form disjoint coalitions, and sometimes also on the way the members of coalitions share the utilities of their cooperations among themselves (in case of games with transferable utility). However, in reality, an agent in the market (or any individual in some social situation) may be involved in more than one cooperation at a time, moreover, a cooperation may be performed with different intensities. For instance, an employer can have several employees and their working hours can be different (but within some reasonable limits).

Scarf [20] proved that every balanced NTU-game (i.e, cooperative game with non-transferable utilities) has a nonempty core. His theorem was based on a lemma, which became known as Scarf's Lemma, as its importance has been recognised for its own right.

In this paper, we give a new interpretation of the fractional solutions which are obtained by the Scarf algorithm for different settings. First we consider the original setting of the Lemma for finitely generated NTU-games, and we describe the meaning of the output in terms of *fractional core*. We show the correspondence between this notion and the concept of *fractional stable matchings for hypergraphs*. We conclude Section 2 by explaining how the normality of a hypergraph implies the nonemptiness of the core for the corresponding NTU-games. In Section 3, we define the stable allocation problem for hypergraphs, which corresponds to the problem of finding a fractional core for NTU-games where the players can be involved in more than one coalition and the joint activities can be performed at different intensity levels (up to some capacity constrains). We show that a variant of the Scarf Lemma implies the existence of the latter solution as well. In Section 4, we apply these results for *matching games* and we derive some well-known theorems in this context. Finally, we present some important open problems and new research directions.

2 Fractional core - fractional stable matchings

In this section, first we describe Scarf's Lemma and we give a new interpretation of the fractional results obtained by the Lemma.

[1]Supported by EPSRC grant EP/E011993/1 and by OTKA grant K69027.
[2]Supported by OTKA grant K69027.

2.1 Definitions, preliminaries

We recall the definition of *n-person games with nontransferable utility* (NTU-game for short).

Definition 1. *An* NTU-game *is given by a pair* (N, V)*, where* $N = \{1, 2, \ldots, n\}$ *is the set of players and* V *is a mapping of a set of feasible utility vectors, a subset* $V(S)$ *of* $\mathbb{R}^S$ *to each coalition of players,* $S \subseteq N$*, such that* $V(\emptyset) = \emptyset$*, and for all* $S \subseteq N$*,* $S \neq \emptyset$*:*

a) $V(S)$ *is a closed subset of* $\mathbb{R}^S$

b) $V(S)$ *is comprehensive, i.e. if* $u^S \in V(S)$ *and* $\tilde{u}^S \leq u^S$ *then* $\tilde{u}^S \in V(S)$

c) The set of vectors in $V(S)$ *in which each player in* S *receives no less than the maximum that he can obtain by himself is a nonempty, bounded set.*

One of the most important solution concepts is the core.

Definition 2. *A utility vector* $u^N \in V(N)$ *is in the core of the game, if there exists no coalition* $S \subseteq N$ *with a feasible utility vector* $\tilde{u}^S \in V(S)$ *such that* $u_i^N < \tilde{u}_i^S$ *for every player* $i \in S$*. Such a coalition is called blocking coalition.*

An NTU-game (N, V) is *superadditive* if $V(S) \times V(T) \subseteq V(S \cup T)$ for every pair of disjoint coalitions S and T. In what follows, we restrict our attention to superadditive games.

Partitioning games are special superadditive games. Given a set of *basic coalitions* $\mathcal{B} \subseteq 2^N$, that contain all singletons (i.e. every single player has the right not to cooperate with the others), a partitioning game $(N, V, \mathcal{B})$ is defined as follows: if $\Pi_{\mathcal{B}}(S)$ denotes the set of partitions of S into basic coalitions, then $V(S)$ can be generated as:

$$V(S) = \{u^S \in \mathbb{R}^S | \exists \pi = \{B_1, B_2, \ldots, B_k\} \in \Pi_{\mathcal{B}}(S) : u^S \in V(B_1) \times V(B_2) \times \cdots \times V(B_k)\}$$

This means that u^S is a feasible utility vector of S if there exist a partition π of S into basic coalitions such that each utility vector $u^S|_{B_i}$ can be obtained as a feasible utility vector by basic coalition B_i in π.

Given an NTU-game (N, V), let $U(S)$ be the set of *Pareto optimal* utility vectors of the coalition S, i.e. $u^S \in U(S)$ if there exists no $\tilde{u}^S \in V(S)$, where $u^S \neq \tilde{u}^S$ and $u^S \leq \tilde{u}^S$.

A utility vector $u^S \in V(S)$ is *separable* if there exist a proper partition π of S into subcoalitions $S_1, S_2, \ldots, S_k$ such that $u^S|_{S_i}$ is in $V(S_i)$ for every $S_i \in \pi$. A utility vector that is non-separable, Pareto-optimal and in which each player receives no less than the maximum that he can obtain by himself is called an *efficient* vector. A coalition S is *essential* if $V(S)$ contains an efficient utility vector. In other words, a coalition S is essential, if its members can obtain an efficient utility vector that is not achievable independently by its subcoalitions. The set of essential coalitions is denoted by $\mathcal{E}(N, V)$.

We say that a coalition S is not *relevant* if for every utility vector $u^S \in V(S)$ there exists a proper subcoalition $T \subset S$ such that $u^S|_T$ is in $V(T)$. The set of relevant coalitions is denoted by $\mathcal{R}(N, V)$. The idea behind this notion is that if a non-relevant coalition S is blocking with a utility vector u^S, then one of its subcoalitions, say T_1, must be also blocking with utility vector $u^{T_1} = u^S|_{T_1}$. Moreover, if T_1 is not relevant or u^{T_1} is separable, then we can find another coalition $T_2 \subset T_1$, such that $u^{T_2} = u^{T_1}|_{T_2} = u^S|_{T_2}$, an so on. Continuing this argument, it is clear that there must be a relevant coalition $T_i \subset S$, that is blocking with a non-separable vector $u^{T_i} = u^S|_{T_i}$. This observation implies the following Proposition:

Proposition 1. *A utility vector* $u^N \in V(N)$ *is in the core if and only if it is not blocked by any relevant coalition with an efficient utility vector.*

Obviously, if a coalition is not essential, then it cannot be relevant either. In a partitioning game, the set of essential coalitions must be a subset of the basic coalitions by definition.

Proposition 2. *For every partitioning game* $(N, V, \mathcal{B})$*,* $\mathcal{R}(N, V, \mathcal{B}) \subseteq \mathcal{E}(N, V, \mathcal{B}) \subseteq \mathcal{B}$ *holds.*

Scarf [20] observed that the previously introduced notions are purely ordinal in character: they are invariant under a continuous monotonic transformation of the utility function of any individual. Hence, without loss of generality, we may assume that $U^{\{i\}} = \{0\}$ for every singleton, and all the efficient utility vectors are nonnegative. Moreover, the discussion can be carried out on an abstract level with the outcomes for each individual represented by arbitrary ordered sets, as we describe this in detail below.

Suppose that in order to obtain a particular non-separable vector $u^{S,k}$ in $U(S)$, the members of S have to perform a joint activity, say $\mathfrak{a}^{S,k}$. Let $\mathcal{A}^S$ denote the set of activities that yield efficient utility vectors in $U(S)$. The preference of a player over the possible activities in which he can be involved is determined by the utilities that he obtains in these activities. Formally, we suppose that $\mathfrak{a}^{S,k} \leq_i \mathfrak{a}^{T,l} \iff u_i^{S,k} \leq u_i^{T,l}$ for any pair of activities $\mathfrak{a}^{S,k}$ and $\mathfrak{a}^{T,l}$, where $i \in S$ and $i \in T$.

Considering an efficient utility vector $u^{N,l}$ of the grandcoalition N, the non-separability implies that $u^{N,l}$ corresponds to a joint activity $\mathfrak{a}^{N,l}$ of the entire set of players. Otherwise, if $u^{N,l}$ is separable, then $u^{N,l}$ can be obtained as a direct sum of independent efficient utility vectors of essential subcoalitions that form a partition of the grandcoalition. This can be considered as a set of independent activities of the subcoalitions. An *outcome of the game*, denoted by X then can be regarded as a partition π of the players and a set of activities $\mathcal{A}^\pi$ performed independently by the coalitions in π, so let $X = (\pi, \mathcal{A}^\pi)$. An outcome X is judged by a player i according to the activity he is involved in, denoted by $\mathfrak{a}_i(X)$. An outcome is in the core of the game, or in other words, it is *stable* if there exist no blocking coalition S and an activity $\mathfrak{a}^{S,k}$ that is strictly preferred by all of its members, i.e., $\mathfrak{a}^{S,k} >_i \mathfrak{a}_i(X)$ for every $i \in S$. (This is equivalent to the blocking condition $u_i^{N,l} < \tilde{u}_i^S$, if the outcome X corresponds to the utility vector $u^{N,l}$.)

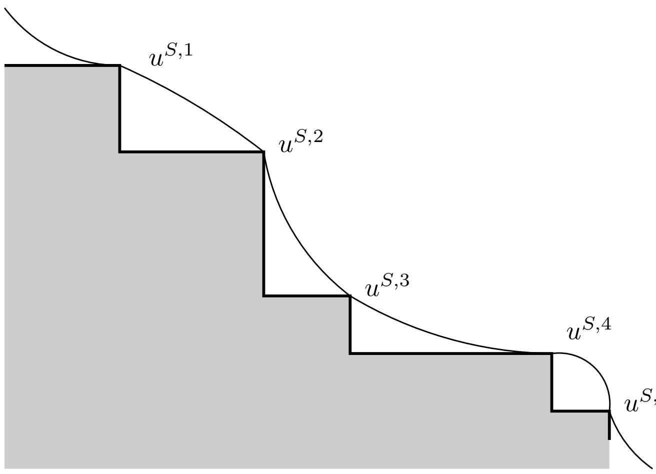
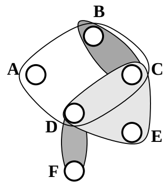


Figure 1: Approximation with finite number of efficient utility vectors.

An NTU-game is *finitely generated* if for every essential coalition S, $U(S)$ contains a finite number of vectors. Here, the preference order of a player over the set of activities, in which he can be involved, can be represented by preference lists. As Scarf observed in [20] and [21], a general NTU-game can be approximated by a finitely generated NTU-game (see an illustration in Figure 1). Here, we will not discuss this question in details.

If for every essential coalition S, in a given NTU-game, $U(S)$ contains only one single vector, u^S then an outcome of the game is simply a partition, since each essential coalition has only one activity to perform. So here, instead of activities, each player has a preference order over the essential coalitions in which he can be a member. These games are called *coalition formation games* (CFG for short), and an outcome that is in the core of the game is called a *core-partition.* The following example illustrates a CFG.

Example 1. *Suppose that we are given 6 players: A, B, C, D, E and F, and 4 possible basic coalitions with corresponding joint activities. The first activity, b (bridge) can be played by A, B, C and D, the second one, p (poker) can be played by C, D and E. Finally, B can play chess with C (denoted by c_1) and D can play chess with F (denoted by c_2). The preferences of the players over the joint activities are as follows.*



Activities	Participants	Players	Preference lists
$b:$	$\{A, B, C, D\}$	$B:$	$b\ c_1$
$p:$	$\{C, D, E\}$	$C:$	$p\ b\ c_1$
$c_1:$	$\{B, C\}$	$D:$	$b\ p\ c_2$
$c_2:$	$\{D, F\}$		

Here, $\{p, \{A\}, \{B\}, \{F\}\}$ is a core-partition, since b is not blocking because C prefers his present coalition p to b, similarly, c_1 is not blocking because C prefers playing poker with D and E to playing chess with B, and c_2 is not blocking because D also prefers playing poker to playing chess with F. One can easily check that $\{b, \{E\}, \{F\}\}$ is also a core-partition, but the partition $\{c_1, c_2, \{A\}, \{E\}\}$ is not in the core, since p and b are both blocking coalitions.

2.2 Fractional core by Scarf's Lemma

First, we present Scarf's Lemma [20] and then we introduce the notion of fractional core. The following description of the Lemma is due to Aharoni and Fleiner [1] (here $[n]$ denotes the set of integers $1, 2, \ldots, n$, and $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise).

Theorem 3 (Scarf, 1967)**.** *Let n, m be positive integers, and b be a vector in $\mathbb{R}^n_+$. Also let $A = (a_{i,j}), C = (c_{i,j})$ be matrices of dimension $n \times (n + m)$, satisfying the following three properties: the first n columns of A form an $n \times n$ identity matrix (i.e. $a_{i,j} = \delta_{i,j}$ for $i, j \in [n]$), the set $\{x \in \mathbb{R}^{n+m}_+ : Ax = b\}$ is bounded, and $c_{i,i} < c_{i,k} < c_{i,j}$ for any $i \in [n], i \neq j \in [n]$ and $k \in [n + m] \setminus [n]$.*

Then there is a nonnegative vector x in $\mathbb{R}^{n+m}_+$ such that $Ax = b$ and the columns of C that correspond to supp(x) form a dominating set, that is, for any column $i \in [n + m]$ there is a row $k \in [n]$ of C such that $c_{k,i} \leq c_{k,j}$ for any $j \in$ supp(x).

Let the columns of A and C correspond to the efficient utility vectors (or equivalently to some activities) of the essential coalitions in a finitely generated NTU-game as follows. If the k-th columns of A and C correspond to the utility vector $u^{S,l}$, then let $a_{i,k}$ be 1 if $i \in S$ and 0 otherwise, (so the k-th column of A is the *membership vector* of coalition S). Furthermore, let $c_{i,k} = u^{S,l}_i$ if $i \in S$ and $c_{i,k} = M$ otherwise, where M is a sufficiently large number. We set $c_{i,i} = u^{\{i\}}_i = 0$ and $c_{i,j} = 2M$ if $i \neq j \leq n$. Finally, let $b = 1^N$. By applying Scarf's Lemma for this setting, we obtain a solution x that we call a *fractional core element* of the game. We refer to the set of fractional core elements as the *fractional core* of the game.

What is the meaning of a fractional core element? Let us suppose first, that a fractional core element x is integer, so $x_i \in \{0,1\}$ for all i. In this case we show that x gives a utility vector u^N that is in the core of the game. Let u^N be the utility vector of N received by summing up those independent essential utility vectors for which $x(u^{S,k}) = 1$, then u^N is obviously in $V(N)$ by superadditivity. To prove that u^N must be in the core of the game, let $u^{S,k}$ be an essential utility vector, with $x(u^{S,k}) = 0$. By the statement of Scarf's Lemma, there must be a player i and an essential utility vector $u^{T,l}$, such that $i \in T$, $x(u^{T,l}) = 1$ and $u_i^{S,k} \leq u_i^{T,l}$, so S cannot be a blocking coalition with the efficient utility vector $u^{S,k}$.

In other words, the $Ax = 1^N$ condition of the solution says that x gives a partition π of N and a set of activities $\mathcal{A}^\pi$ that are performed (we say that $\mathfrak{a}^{S,k}$ is *performed*, i.e. $\mathfrak{a}^{S,k} \in \mathcal{A}^\pi$, if $x(u^{S,k}) = 1$, implying that S is a coalition in partition π). Let $X = (\pi, \mathcal{A}^\pi)$ be the corresponding outcome, and let $\mathfrak{a}^{S,k}$ be an activity not performed, (i.e. $\mathfrak{a}^{S,k} \notin \mathcal{A}^\pi$). Then, by Scarf's Lemma there must be a player i of S for which the performed activity, $\mathfrak{a}_i(X)$ he is involved in is not worse than $\mathfrak{a}^{S,k}$, i.e., $\mathfrak{a}^{S,k} \leq_i \mathfrak{a}_i(X)$, thus S cannot be a blocking coalition with activity $\mathfrak{a}^{S,k}$.

In the non-integer case, we shall regard $x(u^{S,k})$ as the *intensity* at which the activity $\mathfrak{a}^{S,k}$ is performed by coalition S. The $Ax = 1^N$ condition means that each player participates in activities with total intensity 1, including maybe the activity that this player performs alone. The domination condition says that for each activity, which is not performed with intensity 1, there exists a member of the coalition who is not interested in increasing the intensity of this activity, since he is satisfied by some other preferred activities that fill his remaining capacity. Formally, if $x(u^{S,k}) < 1$ then there must be a player i in S such that $\sum_{a^{T,l} \geq_i a^{S,k}} x(u^{T,l}) = 1$.

In Example 1, $x(p) = \frac{1}{3}$, $x(b) = \frac{2}{3}$ is a fractional core element, since for each activity there is at least one player who is not interested in increasing the intensity of that activity. In our corresponding technical report [5] we illustrate with an example that the fractional core of a game may admit a unique fractional core element where the intensities of certain activities can be arbitrary small nonnegative values.

2.3 Fractional stable matching for hypergraphs

For a finitely generated NTU-game, the problem of finding a stable outcome is equivalent to the *stable matching problem* (SM for short) *for a hypergraph*, as defined by Aharoni and Fleiner [1]. Here, the vertices of the hypergraph correspond to the players, the edges correspond to the efficient vectors (or to activities being performed by the players concerned), and the preference of a vertex over the edges it is incident with comes from the preference of the corresponding player over the activities he can be involved in. This is called a *hypergraphic preference system*. A *matching* corresponds to a set of joint activities performed by certain coalitions that form a partition of the grandcoalition together with the singletons (i.e. with the vertices not covered by the matching). A matching M is *stable* if there exist no *blocking edge*, i.e. an edge $e \notin M$ such for that every vertex v covered by e, either v is unmatched in M or strictly prefers e to the edge that covers v in M. The corresponding set of activities gives a stable outcome, since there exist no blocking coalition with an activity that is strictly preferred by all of its members. Note that different activities performed by the same players are represented by *multiple edges* in the corresponding hypergraph. A hypergraph which represents the efficient utility vectors of a CFG is simple (i.e, does not contain multiple edges and loops). [3]

[3] We shall note that Aharoni and Fleiner [1] supposed in their model that the preferences of the players are strict (i.e., no player is indifferent between any pair of activities). In the literature on stable matching, the setting where players may have *ties* in their lists is referred to as *stable matching problem with ties*. In this case, a matching M is (weakly) stable if it does not admit a blocking edge (where the definition of a

The notion of a *fractional stable matching* for an instance of SM for a hypergraph was defined by Aharoni and Fleiner [1] as follows. A function x assigning non-negative weights to edges of the hypergraph is called a *fractional matching* if $\sum_{v\in h} x(h) \leq 1$ for every vertex v. A fractional matching x is called *stable* if every edge e contains a vertex v such that $\sum_{v\in h, e\leq_v h} x(h) = 1$. The existence of a fractional stable matching can be verified by Scarf's Lemma just like the existence of a fractional core element. Actually, these two notions are basically equivalent.

To show the equivalence formally, we consider the polytope of intensity vectors $\{x|Ax = 1^N, x \geq 0\}$ on the one hand, where A is the membership-matrix of the efficient utility vectors (or the corresponding activities) of dimension $n \times (n+m)$ as defined by Scarf's Lemma. On the other hand, the fractional matching polytope is $\{x|Bx \leq 1^N, x \geq 0\}$, where B is the vertex-edge incidence matrix of the hypergraph of dimension $n \times m$. Obviously, $A = (I_n|B)$, so the difference is only the $n \times n$ identity matrix, i.e. the membership-matrix of the singletons. So, there is a natural one-to-one correspondence between the elements of the two polytopes: if x^m is a fractional matching of dimension m, then let $\bar{x}^v = 1^N - Ax^m$ be a vector of dimension n, that gives the unfilled intensities of the players (or in other words, the intensities of the single activities). The direct sum of these two independent vectors, x is an intensity vector of dimension $n+m$, and vice versa. The stability condition is equivalent to the domination condition of Scarf's Lemma.

Aharoni and Fleiner [1] showed that a fractional stable matching can be assumed to be an extremal point of the fractional matching polytope. This fact comes from a statement similar to the following Proposition:

Proposition 4. *If x is a fractional core element of a finitely generated NTU-game, and $x = \sum \alpha_i x^i$, where $\alpha_i > 0$ for all i, $\sum \alpha_i = 1$ and x^i satisfies the $Ax^i = 1^N$ and $x^i \geq 0$ conditions, then each x^i must be a fractional core element.*

The proof of this Proposition is obvious, since $supp(x^i) \subseteq supp(x)$, that implies the dominating property of the fractional core element.

Corollary 5. *For any finitely generated NTU-game, there exists a fractional core element that is an extremal point of the polytope $\{x|Ax = 1^N, x \geq 0\}$.*

Corollary 5 implies that if, for a given finitely generated NTU-game, all the extremal points of the above polytope are integers (or, in other words, the polytope has the *integer property*) then the game has a nonempty core.

2.4 Normality implies the nonemptiness of the core

The definition of a normal hypergraph is due to Lovász [19]. If H is a hypergraph and H' is obtained from H by deleting edges, then H' is called a *partial hypergraph* of H. The *chromatic index* $\chi_e(H)$ of a hypergraph H is the least number of colours sufficient to colour the edges of H so that no two edges with the same colour have a vertex in common. Note that the maximum degree, $\Delta(H)$ (that is, the maximum number of edges containing some one vertex) is a lower bound for the chromatic index. A hypergraph H is *normal* if every partial hypergraph H' of H satisfies $\chi_e(H') = \Delta(H')$. Obviously, the normality is preserved by adding or deleting multiple edges or loops. The following theorem of Lovász [19] gives an equivalent description of normal hypergraphs.

blocking edge is the same as described above). However, an instance of SM with ties can be always derived to another instance of SM (with no ties) by simply breaking the ties arbitrary, and a matching that is stable for the derived instance is (weakly) stable for the original one. The same applies for the core and fractional core in the context of NTU-games. In fact, the Scarf algorithm starts with a perturbation of matrix C in the case that any player is indifferent between two activities she may be involved in (i.e., when her utilities in these two activities are the same for her).

Theorem 6 (Lovász). *The fractional matching polytope of a hypergraph H has the integer property if and only if H is normal.*

Suppose that for a finitely generated NTU-game the set of essential coalitions forms a normal hypergraph. The hypergraph of the corresponding SM must be also normal, since it is obtained by adding multiple edges and by removing the loops. By Theorem 6, the fractional matching polytope, $\{x|Bx \leq 1^N, x \geq 0\}$ has the integer property, and so has the polytope of intensity vectors, $\{x|Ax = 1^N, x \geq 0\}$ as it was discussed previously. This argument and Corollary 5 verify the following Lemma 7.

Lemma 7. *If, for a finitely generated NTU-game, the set of essential coalitions, $\mathcal{E}(N,V)$ forms a normal hypergraph, then the core of the game is nonempty.*

By Lemma 7 and Proposition 2 the following holds.

Theorem 8. *If the set of basic coalitions, $\mathcal{B}$ forms a normal hypergraph, then every finitely generated NTU-game $(N,V,\mathcal{B})$ has a nonempty core.*

Let $A^{\mathcal{B}}$ denote the membership-matrix of the set of basic coalitions $\mathcal{B}$. The fact that the integer property of the polytope $\{x|A^{\mathcal{B}}x = 1^N, x \geq 0\}$ implies the nonemptiness of every NTU-game $(N,V,\mathcal{B})$ was proved first by Vasin and Gurvich [23], and independently, by Kaneko and Wooders [14]. Later, Le Breton *et al.* [18], Kuipers [17] and Boros and Gurvich [8] observed independently that the integer property of the polytope $\{x|A^{\mathcal{B}}x = 1^N, x \geq 0\}$ is equivalent to the integer property of the matching polytope $\{x|A^{\mathcal{B}}x \leq 1^N, x \geq 0\}$, and to the normality of the corresponding hypergraph.

3 Fractional b-core with capacities - stable allocations

In what follows, we introduce the notion of fractional b-core element as a solution of Scarf's Lemma with the original settings. Let the same matrices A and C of dimension $n \times (n+m)$ correspond to the set of effective utility vectors (or activities) in a finitely generated NTU-game as it was described in the previous section. The only modification is that now b is an arbitrary vector of $\mathbb{R}^n_+$ (instead of 1^N). Let $x \in \mathbb{R}^{n+m}_+$ be referred to as a *fractional b-core element* if x is a solution of the Scarf Lemma for the above setting.

Here, $b(i)$ is an upper bound for the total intensity at which player i is capable to perform activities, since $\sum_{i\in S} x(u^{S,l}) = b(i)$. The domination condition of the Lemma says that for every activity $\mathfrak{a}^{T,k}$, there exists some player i who is not interested in increasing the intensity of $\mathfrak{a}^{T,k}$, because his remaining intensity is filled with better activities, i.e., if $u^{T,k}$ corresponds to activity $\mathfrak{a}^{T,k}$, then $\sum_{u_i^{S,l} \geq u_i^{T,k}} x(u^{S,l}) = b(i)$.

In fact, to produce a fractional core element (in other words, a fractional 1^N-core element) with the algorithm of Scarf, we perturb not just matrix C (in case of indifferences), but also the vector 1^N, to avoid the degeneracy. The standard nondegeneracy assumption provides that all variables associated with the n columns of a feasible basis for the equations $A\tilde{x} = \tilde{b} = 1^N + \varepsilon^N$ are strictly positive. Thus, the perturbation uniquely determines the steps of Scarf algorithm. By rounding the final fractional $\tilde{b}$-core element $\tilde{x}$, a fractional core element x is found. The following simple Lemma says that the fractional b-core element has the *scaling property*.

Lemma 9. *Given a finitely generated NTU-game, and a positive constant λ. Suppose that $b' = \lambda \cdot b$, then x is a fractional b-core element if and only if $x' = \lambda \cdot x$ is a fractional b'-core element.*

Let us suppose that the intensities of the activities in the finitely generated NTU-game are constrained by *capacities.* Formally, for each joint activity $\mathfrak{a}^{S,l}$ and for the corresponding utility vector $u^{S,l}$, there may exist a nonnegative capacity $c(u^{S,l})$ for which $x(u^{S,l}) \leq c(u^{S,l})$ is required.

The stable allocation problem can be defined for hypergraphs as follows. Suppose that we are given a hypergraph H and for each vertex v a strict preference order over the edges incident with v (again, this corresponds to the preferences of the players over the activities in which they can be involved). Suppose, that we are given nonnegative *bounds* on the vertices $b : V(H) \to \mathbb{R}_+$ and nonnegative *capacities* on the edges $c : E(H) \to \mathbb{R}_+$. A nonnegative function x on the edges, is an *allocation* if $x(e) \leq c(e)$ for every edge e and $\sum_{v \in h} x(h) \leq b(v)$ for every vertex v. An allocation is *stable* if every *unsaturated* edge e (i.e., every edge e with $x(e) < c(e)$) contains a vertex v such that $\sum_{v \in h, e \leq_v h} x(h) = b(v)$. In this case we say that e is *dominated at* v. If every bound and capacity is integral then we refer to this problem as the *integral stable allocation problem.*

Theorem 10. *Every stable allocation problem for hypergraphs is solvable.*

Proof. Suppose that we are given a given a hypergraph H. Let $V(H) = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices and let $E(H) = \{e_1, e_2, \ldots, e_m\}$ be the set of edges. We define the extended membership-matrix A, and the extended preference-matrix C of size $(n+m) \times (n+2m)$ as follows.

The left part of A is an identity matrix of size $(n+m) \times (n+m)$, (i.e. $a_{i,j} = \delta_{i,j}$ for $i, j \in [n+m]$). At the bottom of the right side there is another identity matrix of size $m \times m$, so $a_{n+i,n+m+j} = \delta_{i,j}$ for $i, j \in [m]$. Finally, at the top of the right side we have the vertex-edge incidence matrix of H (i.e. $a_{i,n+m+j} = 1$ if $v_i \in e_j$ and 0 otherwise for $i \in [n]$ and $j \in [m]$).

The top-right part of C correspond to the preference of the vertices (that is the preference of the players over the activities). We require the following two conditions:

- $c_{i,n+m+j} < c_{i,n+m+k}$ whenever $v_i \in e_j \cap e_k$ and $e_j <_{v_i} e_k$;
- $c_{i,n+m+j} < c_{i,n+m+k}$ whenever $v_i \in e_j \setminus e_k$.

Furthermore, suppose that $c_{n+i,n+m+i} < c_{n+i,n+m+j}$ for every $i \neq j \in [m]$ in the bottom-right part of C. Finally, let the left part of C be such that it satisfies the conditions of Scarf's Lemma. The constant vector, $b \in R_+^{n+m}$ is given by the bounds and capacities, so let $b_i = b(v_i)$ for $i \in [n]$ and $b_{n+j} = c(e_j)$ for $j \in [m]$.

We shall prove that the fractional core element x, obtained by Scarf's Lemma, gives a stable allocation, x^e by simply taking the last m coordinates of x. Here, x_j^e is equal to $x^e(e_j)$ that is the weight of the edge e_j (or equivalently, this is the intensity at which the corresponding activity is performed). If $\bar{x}^v$ and $\bar{x}^e$ are the vectors obtained by taking the $[1, \ldots, n]$ and $[n+1, \ldots, n+m]$ coordinates of x, then these vectors correspond to the remaining weights of the vertices and edges (or the remaining intensities of the players and the activities), respectively.

Obviously, x^e is an allocation by $Ax = b$, since the first n equations preserve the $\sum_{v \in h} x^e(h) \leq b(v)$ condition for every vertex v, and the last m equations preserve $x^e(e) \leq c(e)$ for every edge e.

To prove stability, let us consider an unsaturated edge e_j and let us suppose that the corresponding dominating row by the lemma has index k. First we show, that $i \in [n]$. From $Ax = b$, obviously $\bar{x}^e(e_k) + x^e(e_k) = c(e_k)$ for every edge e_k. Since $x^e(e_j) < c(e_j)$, then $\bar{x}^e(e_j) > 0$, thus the assumptions on C imply that $i \neq n+j$, for other $i \in [n+m] \setminus [n]$ the contradiction is trivial. If $i \in [n]$, then e is dominated at v_i, since $\bar{x}^v(v_i) = 0$ by the assumptions on C, and the $Ax = b$ condition for the i-th row together with the statement of the lemma imply $\sum_{v_i \in h, e_j \leq_{v_i} h} x^e(h) = b(v_i)$. ☐

4 Matching games

Matching games can be defined as partitioning NTU-games, where the cardinality of each basic coalition is at most 2. For simplicity, in this section we suppose that no player is indifferent between two efficient utility vectors, so their preferences over the joint activities are strict. If a matching game is finitely generated, then the problem of finding an outcome that is in the core is equivalent to a SM for a graphic preference system, where the edges of the graph correspond to efficient utility vectors (and to joint activities).

4.1 Stable matching problem

If the graph of a matching game is simple (i.e, if it contains no multiple edges and loops) then the problem of finding a core-partition for the resulting CFG is called *stable roommates problem*. Otherwise, if the graph has multiple edges then we may refer to SM as *stable roommates problem with multiple activities*.

Let us suppose the set of players N can be divided into two parts, say M and W, such that every two-member basic coalition contains one member from each side (so if $\{m, w\} \in \mathcal{B}$ then $m \in M$ and $w \in W$). In this case, we get a *two-sided matching game* (in the general nonbipartite case the matching game is called *one-sided*).

If a two-sided matching game is finitely generated then the corresponding graphic representation of the SM is bipartite. For bipartite graphs, the following Proposition is well-known.

Proposition 11. *Every bipartite graph is normal.*

Proposition 11 and Theorem 8 imply the following result.

Theorem 12. *Every finitely generated two-sided matching game has a nonempty core.*

Theorem 12 was proved for every two-sided matching game, originally called *central assignment game*, by Kaneko [13]. For the corresponding CFG-s, called *stable marriage problems*, this result was proved by Gale and Shapley [11].

A one-sided matching game can have an empty core, even for a CFG, as Gale and Shapley [11] illustrated with an example. However the half-integer property of the fractional matching polytope implies the existence of stable half-solutions. The following statement is due to Balinski [4].

Theorem 13. *The fractional matching polytope for every graph has only half-integer extremal points.*

As Aharoni and Fleiner [1] showed, Theorem 13 and Corollary 5 imply that for every matching game there exists a so-called *half-core element*, that is a fractional core element x with the half-integer property, i.e. $x_i \in \{0, \frac{1}{2}, 1\}$.

Theorem 14. *If a matching game is finitely generated then it always has a half-core element.*

For CFG-s, the fact that for every instance of SM there exists a stable half-matching was proved by Tan [22]. Finally we note that an easy consequence of Theorem 13 and Lemma 9 is that for every finitely generated matching game, there always exists an integer 2^N-core element.

4.2 Stable allocation problem for graphs

The stable allocation problem was introduced by Baïou and Balinski [3] for bipartite graphs. The integer version, where the allocation x is required to be integer on every edge for integer bounds and capacities, was called the *stable schedule problem* by Alkan and Gale [2] (however they considered a more general model, the case of so-called *substitutable* preferences).

Biró and Fleiner [6] generalised the algorithm of Baïou and Balinski [3] for nonbipartite graphs, resulting in a weakly polynomial algorithm that finds a half-integral stable allocation for any given instance of integral stable allocation problem. Dean and Munshi [9] strengthened this result by giving a strongly polynomial algorithm for the same problem. But we shall note that the existence of a stable half-integer allocation is a consequence of Theorem 10.

Theorem 15. *For every integral stable allocation problem in a graph there exists a half-integral stable allocation. If the graph is bipartite, then every integral stable allocation problem is solvable.*

Proof. Suppose that we have a stable allocation x that has some weights that are not half-integers. We create another stable allocation x' with half-integer weights as follows. If $x(e)$ is not integer then let v be the vertex where e is dominated. Since $b(v)$ is integer, there must be another edge, f that is incident with v and has non-integer weight. Moreover, f cannot be dominated at v. By this argument, it can be verified that the edges with non-integer weights form vertex-disjoint cycles, moreover, in each such a cycle the fractional parts of the weights are ε and $1-\varepsilon$ alternately. If a cycle is odd, then ε must be $\frac{1}{2}$. If a cycle is even, then ε can be modified to be 0 (or 1) in such a way that the obtained allocation x' remains stable and has only half-integer weights.

If the graph is bipartite, thus has no odd cycle, then x' has only integer weights, so x' is an integral stable allocation . $\square$

In [5] we give an integral stable allocation problem for a graph and we illustrate how a half-integer stable allocation can be obtained with the Scarf algorithm.

5 Further directions

Guarantees for solvability. The original goal of Scarf [20] was to give a necessary condition for the nonemptiness of the core for general NTU-games (and this condition was the balancedness of the game). As we described in Section 3, if the coalition structure of an NTU-game can be represented with a normal hypergraph then the core of the game is always nonempty (regardless of the players' preferences). The bipartite graph is an easy example for normal hypergraphs, and so every two-sided matching game has nonempty core. But what other games have this property? Our claim is that certain network games also have a coalition structure where the underlying hypergraph is normal.

Understanding Scarf's results. Scarf proved his Lemma in an algorithmic way. Is there some deeper reasons for the correctness of the Lemma (and a more general interpretation of the algorithm)? What is the relation of this result to other fundamental theorems, such as the Sperner Theorem? There are some recent papers [15, 10] attempting to answer this question, but yet, there still are many open problems regarding this issue.

Also, it would be interesting to know how the Scarf algorithm works for special games. For instance, does the Scarf algorithm run in polynomial time for matching games?

At the beginning of the Scarf algorithm we perturb matrix C and vector b. By doing so, the steps in the algorithm and the final output are fully determined. Can we output every core element of a given NTU-game by using a suitable perturbation? How does the perturbation effect the solution, we obtain by the algorithm? For stable marriage problem we observed that using small epsilons for men and larger epsilons for women we always get the man-optimal stable matching. Can we output each stable matching by a suitable perturbation? Is that true that the smaller epsilon we give to a woman the better partner she is going to get in the resulting stable matching?

Further application of Scarf Lemma. It is possible that the contribution of the participants are not equal in a cooperation. Imagine an internal project of a company where the hours allocated to the employees involved can be different (e.g., a project manager may have less work load than an engineer in terms of working hours). We can facilitate this option easily for any stable matching or stable allocation problem (that we may call *stable allocation problem with contributions*). We only need to use *contribution vectors* rather than membership vectors when defining matrix A in Scarf's Lemma, and the existence of a stable solution is guaranteed. But can we find a stable integral solution in polynomial time for, say, two-sided matching games?

Practical applications. As Gale and Shapley [11] envisaged, stable matching problems turned out to be very useful models for real applications in two-sided markets. Centralised matching schemes have been established worldwide to allocate residents to hospitals, students to schools, and so on. In most cases, a stable solution can be found by the classical Gale-Shapley algorithm. However, there are some special features, such as the presence of *couples* in the residence allocation program, that can make the problem unsolvable (or even if a stable matching exists, the problem of finding one can be NP-hard). Although if the ratio of the couples is relatively small in a large market then a stable matching exists with high probability and sophisticated heuristics may be able to find such solutions (see e.g., [16] and [7]). A new heuristic for this problem could be based on the Scarf algorithm for a stable allocation problem, where a hyperedge would represent an application from a couple to a pair of hospitals. If the solution obtained by the Scarf algorithm is integral then it would correspond to a stable matching. We illustrate this application with an example in [5].

References

[1] R. Aharoni and T. Fleiner. On a lemma of Scarf. *J. Combin. Theory Ser. B*, 87(1):72–80, 2003. Dedicated to Crispin St. J. A. Nash-Williams.

[2] A. Alkan and D. Gale. Stable schedule matching under revealed preference. *J. Econom. Theory*, 112(2):289–306, 2003.

[3] M. Baïou and M. L. Balinski. The stable allocation (or ordinal transportation) problem. *Math. Oper. Res.*, 27(3):485–503, 2002.

[4] M. L. Balinski. Integer programming: Methods, uses, computation. *Management Sci.*, 12:253–313, 1965.

[5] P. Biró and T. Fleiner. Fractional solutions for NTU-games. Technical report, no. TR-2010-315, University of Glasgow, Department of Computing Science, 2010.

[6] P. Biró and T. Fleiner. The integral stable allocation problem on graphs. *Discrete Opt.*, 7(1-2):64–73, 2010.

[7] P. Biró, R. W. Irving. Stable matching with couples – an empirical study. *Working paper*, 2010.

[8] E. Boros and V. Gurvich. Perfect graphs are kernel solvable. *Discrete Math.*, 159(1-3):35–55, 1996.

[9] B. C. Dean and S. Munshi. Faster algorithms for Stable Allocation Problems. To appear in *Algorithmica*, 2010.

[10] J. Edmonds and S. Gaubert and V. Gurvich. On Scarf and Sperner Oiks. *Rutcor Research Report*, RRR 18-2009, 2009.

[11] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *Amer. Math. Monthly*, 69(1):9–15, 1962.

[12] R. W. Irving and S. Scott. The stable fixtures problem—a many-to-many extension of stable roommates. *Discrete Appl. Math.*, 155(16):2118–2129, 2007.

[13] M. Kaneko. The central assignment game and the assignment markets. *J. Math. Econom.*, 10(2-3):205–232, 1982.

[14] M. Kaneko and M. H. Wooders. Cores of partitioning games. *Math. Social Sci.*, 3(4):313–327, 1982.

[15] T. Király and J. Pap. Kernels, stable matchings, and Scarf's Lemma. *Egerváry Research Group*, TR-2008-13, 2008.

[16] F. Kojima, P. A. Pathak and A. E. Roth. Matching with couples: Stability and incentives in large markets. *Working paper*, 2010.

[17] J. Kuipers. *Combinatorial methods in cooperative game theory.* Maastricht University, 1994. PhD Thesis.

[18] M. Le Breton, G. Owen, and S. Weber. Strongly balanced cooperative games. *Internat. J. Game Theory*, 20(4):419–427, 1992.

[19] L. Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Math.*, 2(3):253–267, 1972.

[20] H. E. Scarf. The core of an N person game. *Econometrica*, 35:50–69, 1967.

[21] H. E. Scarf. *The computation of economic equilibria*. Yale University Press, New Haven, Conn., 1973. With the collaboration of Terje Hansen, Cowles Foundation Monograph, No. 24.

[22] J. J. M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *J. Algorithms*, 12(1):154–178, 1991.

[23] A. A. Vasin and V. A. Gurvič. Reconcilable sets of coalitions. In *Questions in applied mathematics (Russian)*, pages 20–30. Sibirsk. Ènerget. Inst., Akad. Nauk SSSR Sibirsk. Otdel., Irkutsk, 1977.

Péter Biró
Department of Computing Science
University of Glasgow
Glasgow G12 8QQ, UK
Email: `pbiro@dcs.gla.ac.uk`

Tamás Fleiner
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Budapest
Magyar tudósok körútja 2.
H-1117, Hungary
Email: `fleiner@cs.bme.hu`

Dependence in Games and Dependence Games

Davide Grossi and Paolo Turrini

Abstract

The paper provides a formal analysis of a notion of dependence between players in a game. We will show: first, how this notion of dependence allows for an elegant characterization of a property of reciprocity for the outcomes of a game; and second, how it can be used to ground new cooperative solution concepts for strategic games, where coalitions can force outcomes only in the presence of reciprocal dependences.

1 Introduction

The paper outlines a theory of dependence for strategic games. It moves from the following definition of dependence, inspired by foundational literature on multi-agent systems (see for instance [2, p.4]): player i depends on player j for reaching outcome s, within a given game, if and only if j plays a strategy, in the profile determining s, which is a best response (or a dominant strategy) not for j itself, but instead for i (Definition 8).

The aim of the paper is to provide a thorough analysis of the above definition. Concretely, it presents two results. First, it shows that this notion of dependence allows for the characterization of an original notion of *reciprocity* for strategic games (Theorem 1). Second, it shows that this notion of dependence can be fruitfully applied to ground cooperative solution concepts. These solution concepts are characterizable as the core of a specific class of coalitional games—here called *dependence games*—where coalitions can force outcomes only in the presence of reciprocity (Theorems 2 and 3). The paper generalizes and extends results presented in [3].

2 Dependence in games

The section introduces some preliminary notions and notation from game theory and proceeds to the definition and analysis of the notion of dependence.

2.1 Preliminary definitions and notation

The present section introduces the basic game-theoretic notions used in the paper. All definitions will be based on an ordinal notion of preference. Our main sources are [5] and [4].

Definition 1 (Game) *A (strategic form) game is a tuple $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ where: N is a set of players; S is a set of outcomes; Σ_i is a set of strategies for player $i \in N$; $\succeq_i$ is a total preorder on S; $o : \bigtimes_{i \in N} \Sigma_i \to S$ is a bijective function from the set of strategy profiles to S. Strategy profiles will be denoted $\sigma, \sigma', \ldots$.*

We will also make use of the notion of sub-game.

Definition 2 (Sub-game of a strategic game) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game, σ be a strategy profile, and $C \subseteq N$. The subgame of $\mathcal{G}$ defined by σ_C is a game $\mathcal{G} \downarrow \sigma_c = (N', S', \Sigma'_i, \succeq'_i, o')$ such* *that: $N' = N - C$; $S' = S - \{s \mid \exists\sigma'$ s.t. $s = o(\sigma')$ and $\sigma'_C \neq \sigma_C\}$; for all $i \in N - C$, $\Sigma'_i = \sigma_i$; for all $i \in N - C$, $\Sigma'_i = \succeq_i$; $o' : \bigtimes_{i \in N-C} \Sigma_i \to S'$ is a bijection such that for all $\sigma' \in \bigtimes_{i \in N-C} \Sigma_i$, $o'(\sigma') = o(\sigma', \sigma_C)$.*

To put it in words, a subgame of $\mathcal{G}$ is nothing but what it is obtained from $\mathcal{G}$ once the strategies of a set of players in C are fixed, or what is still 'left to play' once the players in C have made their choice.

As to the solution concepts, we will work with Nash equilibrium, which we will refer to also as best response equilibrium (*BR*-equilibrium), and the dominant strategy equilibrium (*DS*-equilibrium).

Definition 3 (Equilibria) *Let $\mathcal{G}$ be a game. A strategy profile σ is: a BR-equilibrium (Nash equilibrium) iff $\forall i \in N, \forall \sigma'_i \in \Sigma_i : o(\sigma) \succeq_i o(\sigma'_i, \sigma_{-i})$; it is a DS-equilibrium iff $\forall i \in N, \sigma' \in \bigtimes_{i \in N} \Sigma_i : o(\sigma_i, \sigma'_{-i}) \succeq_i o(\sigma')$.*

In addition to the games in strategic form (Definition 1) we will also work with coalitional games, i.e., cooperative games with non-transferable pay-offs abstractly represented by effectivity functions [4].

Definition 4 (Coalitional game) *A coalitional game is a tuple $\mathcal{C} = (N, S, E, \succeq_i)$ where: N is a set of players; S is a set of outcomes; E is function $E : 2^N \to 2^{2^S}$; $\succeq_i$ is a total preorder on S.*

An effectivity function associates to a coalition a set of sets of outcomes and the fact that $X \in E(C)$ is usually understood as the coalition C being able to force the interaction to end up in an outcome in X. This intuition can be given a concrete semantics in terms of strategic games, from which a coalitional games can be obtained in a canonical way (cf. [4]). These games in particular will be object of study in Section 3.

Definition 5 (Coalitional games from strategic ones) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game. The coalitional game $\mathcal{C}^{\mathcal{G}} = (N, S, E^{\mathcal{G}}, \succeq_i)$ of $\mathcal{G}$ is a coalitional game where the effectivity function $E^{\mathcal{G}}$ is defined as follows:*

$$X \in E^{\mathcal{G}}(C) \Leftrightarrow \exists\sigma_C \forall\sigma_{\overline{C}}\, o(\sigma_C, \sigma_{\overline{C}}) \in X.$$

As it can be observed from the translation, the effectivity function of $\mathcal{C}^{\mathcal{G}}$ contains those sets in which a coalition C can force the game to end up, no matter what strategies $\overline{C}$ decides to play.

Finally we consider the standard solution concept for coalitional games.

Definition 6 (The Core) *Let $\mathcal{C} = (N, S, E, \succeq_i)$ be a coalitional game. We say that a state s is dominated in $\mathcal{C}$ if for some C and $X \in E(C)$ it holds that $x \succ_i s$ for all $x \in X, i \in C$. The core of $\mathcal{C}$, in symbols $CORE(\mathcal{C})$ is the set of undominated states.*

Intuitively, the core is the set of those states in the game that are stable, i.e., for which there is no coalition that is at the same time able and interested to deviate from them.

2.2 Dependence

We will work with the following notion of dependence: a player i depends on a player j for the realization of an outcome s, i.e., of the strategy profile σ such that $o(\sigma) = s$, when, in order for σ to occur, j has to favour i, that is, it has to play in i's interest. To put it otherwise, i depends on j for σ when, in order to achieve σ, j has to do a favour to i by playing σ_j (which is obviously not under i's control). This intuition is made clear in the following definition:

	g	$\neg g$
g	3, 3, 3	2, 4, 2
$\neg g$	4, 2, 2	1, 1, 0

g

	g	$\neg g$
g	2, 2, 4	0, 1, 1
$\neg g$	1, 0, 1	1, 1, 1

$\neg g$

Figure 1: A three person Prisoner's dilemma.

Definition 7 (Best for someone else) *Assume a game $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ and let $i, j \in N$. 1) Player j's strategy in σ is a* best response *for i iff $\forall \sigma'$, $o(\sigma) \succeq_i o(\sigma'_j, \sigma_{-j})$. 2) Player j's strategy in σ is a* dominant strategy *for i iff $\forall \sigma'$, $o(\sigma_j, \sigma'_{-j}) \succeq_i o(\sigma')$.*

Definition 7 generalizes the standard definitions of best response and dominant strategy by allowing the player holding the preference to be different from the player whose strategies are considered.

Definition 8 (Dependence) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game and $i, j \in N$. 1) Player i* BR-depends *on j for strategy σ—in symbols, $iR^{BR}_{\sigma}j$—if and only if σ_j is a best response for i in σ. 2) Player i* DS-depends *on j for strategy σ—in symbols, $iR^{DS}_{\sigma}j$—if and only if σ_j is a dominant strategy for i.*

Intuitively, i depends on j for profile σ in a best response sense if, in σ, j plays a strategy which is a best response for i given the strategies in σ_{-j} (and hence given the choice of i itself), and similarly for dominant strategy dependence.

In general, relations R^{BR}_{σ} and R^{DS}_{σ} do not enjoy any particular structural property. However, the following simple fact is of interest as it shows a direct connection between dependence graphs and underlying games.

Fact 1 (Reflexive dependencies) *Let $\mathcal{G}$ be a game and (N, R^x_{σ}) be its dependence structure for outcome σ with $x \in \{BR, DS\}$. It holds that R^x_{σ} is reflexive iff σ is an x-equilibrium.*

The proof is omitted for space reasons. The relation of dependence acquires interest for cooperative interaction when a structural property, namely the presence of cycles, suggests the possibility of players acting for each other. The following three sections study this property.

2.3 Cycles

As also emphasized by related contributions (see for instance [1]), cycles in dependence graphs represent the possibility of social interaction between players of a *do-ut-des* (give-to-get) type. In a cycle, the first player of the cycle could be prone to do what the last player asks since it can obtain something from the second player who, in turn, can obtain something from the third and so on.

Definition 9 (Dependence cycles) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game, (N, R^x_{σ}) be its dependence structure for profile σ with $x \in \{BR, DS\}$, and let $i, j \in N$. An R^x_{σ}-dependence cycle $\mathfrak{c}$ of length $k-1$ in $\mathcal{G}$ is a tuple $(a_1, \ldots, a_k)$ such that: $a_1, \ldots, a_k \in N$; $a_1 = a_k$; $\forall a_i, a_j$ with $1 \leq i \neq j < k$, $a_i \neq a_j$; $a_1 R^x_{\sigma} a_2 R^x_{\sigma} \ldots R^x_{\sigma} a_{k-1} R^x_{\sigma} a_k$. Given a cycle $\mathfrak{c} = (a_1, \ldots, a_k)$, its orbit $O(\mathfrak{c}) = \{a_1, \ldots, a_{k-1}\}$ denotes the set of its elements.*

In other words, cycles are sequences of pairwise different players, except for the first and the last which are equal, such that all players are linked by a dependence relation. Note that the definition allows for cycles of length 1, whose orbit is a singleton, i.e., loops. Those are the cycles occurring at reflexive points in the graph.

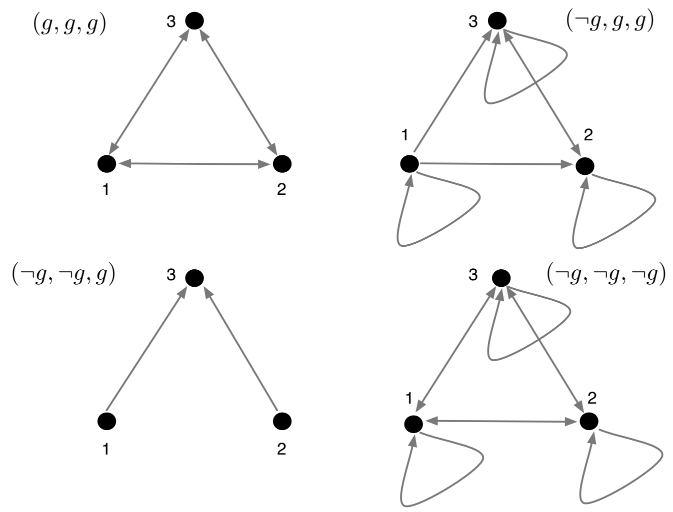


Figure 2: Some BR-dependences of Example 1.

Cycles become of particular interest in games with more than two players, so let us illustrate the definition by the following example.

Example 1 (Cycles in three person games.) *Consider the following three-person variant of the Prisoner's dilemma. A committee of three juries has to decide whether to declare a defendant in a trial guilty or not. All the three juries want the defendant to be found guilty, however, all three prefer that the others declare her guilty while she declares her innocent. Also, they do not want to be the only ones declaring her guilty if the other two declare her innocent. They all know each other's preferences. Figure 1 gives a payoff matrix for such game. Figure 2 depicts some cyclic BR-dependencies inherent in the game presented. Player 1 is Row, player 2 Column, and player 3 picks the right or left table. Among the ones depicted, the reciprocal profiles are clearly* (g, g, g)*,* $(\neg g, \neg g, \neg g)$ *(which is also universal) and* $(\neg g, g, g)$*, only the last two of which are Nash equilibria (reflexive). Looking at the cycles present in these BR-reciprocal profiles, we notice that* (g, g, g) *contains the* 2×3 *cycles of length 3, all yielding the partition* $\{\{1, 2, 3\}\}$ *of the set of players* $\{1, 2, 3\}$*. Profile* $(\neg g, g, g)$*, instead, yields two partitions:* $\{\{1\}, \{2\}, \{3\}\}$ *and* $\{\{1\}, \{2, 3\}\}$*. The latter is determined by the cycles* $(1, 1)$ *and* $(2, 3, 2)$ *or* $(1, 1)$ *and* $(3, 2, 3)$*. Finally, profile* $(\neg g, \neg g, g)$ *is such that both 1 and 2 depend on 3. Yet, neither of them plays a best response strategy.*

The notion of reciprocity obtains a formal definition in the following section.

2.4 Reciprocity

Depending on the properties of the dependence cycles of a given profile, we can distinguish between several notions of reciprocity capturing different ways in which players are interconnected via a dependence structure.

Definition 10 (Types of reciprocity in profiles) *Let* $\mathcal{G}$ *be a game and* (N, R^x_σ) *be its dependence structure with* $x \in \{BR, DS\}$ *and* σ *be a profile, and* $C \subseteq N$*.*

i) A profile σ *is* x-reciprocal *if and only if there exists a partition* $P(N)$ *of* N *such that each element* p *of the partition is the orbit of some* R^x_σ*-cycle, i.e., a cycle in the directed graph* (N, R^x_σ)*;*

ii) *A profile σ is* partially x-reciprocal in C *(or C-x-reciprocal) if and only if C is the orbit of some R^x_σ-cycle, i.e., a cycle in the directed graph (N, R^x_σ);*

iii) *A profile σ is* trivially x-reciprocal *if and only if it yields only x-cycles whose orbits are singletons;*

iv) *A profile σ is* fully x-reciprocal *if and only if it yields an x-cycle with orbit N (i.e., a Hamiltonian cycle) or, equivalently, if and only if it is N-x-reciprocal.*

Let us explain the above definitions by referring to BR-dependence. A profile σ is BR-reciprocal if all players belong to some cycle of BR-dependence. Along the same lines, a profile σ is partially BR-reciprocal in coalition C (or C-BR-reciprocal) if the all the members of C are linked by a cycle of BR-dependence. This means, intuitively, that independently on whether the players outside of coalition C are linked by dependencies or not, the members of C are in a situation of reciprocity in which everybody plays a dominant strategy for somebody else in the coalition. To put it yet otherwise, a profile is reciprocal when the corresponding dependence relation, be it a *BR*- or *DS*-dependence, clusters the players into non-overlapping groups whose members are all part of some cycle of dependencies (including degenerate ones such as reflexive links). It is partially reciprocal if its dependence graph contains at least one cycle. Finally, trivial and full *BR*-reciprocity refer to two extreme cases of *BR*-reciprocity. In the first case the cycles are loops, that is, all players play their own dominant strategy, in the second case there exists one Hamiltonian cycle, that is, all players are connected to one another by a path of *BR*-dependence. For example, inspecting the BR-dependencies in the Prisoner Dilemma (Figure 3) it can be observed that: (U, L) is fully BR-reciprocal, (D, R) is trivially *BR*-reciprocal, (U, R) is $\{2\}$-BR-reciprocal and (D, L) is $\{1\}$-BR-reciprocal.

It is worth noting that x-reciprocity is a more demanding requirement than C-x-reciprocity as it is easy to see that if σ is x-reciprocal, then for each $C \in P(N)$ σ is C-x-reciprocal. Also, here below we report a few simple but relevant facts concerning the logical relationship between DS- and BR-reciprocity.

Fact 2 (DS- vs. BR-reciprocity) *Let $\mathcal{G}$ be a game and (N, R^x_σ) be its dependence structure with $x \in \{BR, DS\}$, σ be a profile, and $C \subseteq N$. The following holds:*

i) *σ is C-BR-reciprocal iff σ_C is BR-reciprocal in $\mathcal{G} \downarrow \sigma_{\overline{C}}$;*

ii) *σ is C-BR-reciprocal iff σ_C is DS-reciprocal in $\mathcal{G} \downarrow \sigma'_{\overline{C}}$ for any σ';*

iii) *if σ is C-DS-reciprocal, then σ is C-BR-reciprocal, but not vice versa;*

iv) *if σ is DS-reciprocal, then σ is BR-reciprocal, but not vice versa.*

The proof is omitted for space reasons. The first claim suggests that a *C-DS*-reciprocal profile σ can be referred to simply by the partial profile σ_C without loss of information. The second and third claims point out, as expected, that DS-reciprocity is a stronger notion than BR-reciprocity.

2.5 Reciprocity and equilibrium

We provide a characterization of reciprocity as defined in Definition 10 in terms of standard solution concepts. However, we first have to complement the set of notions provided in Section 2.1 with the notion of permuted game.

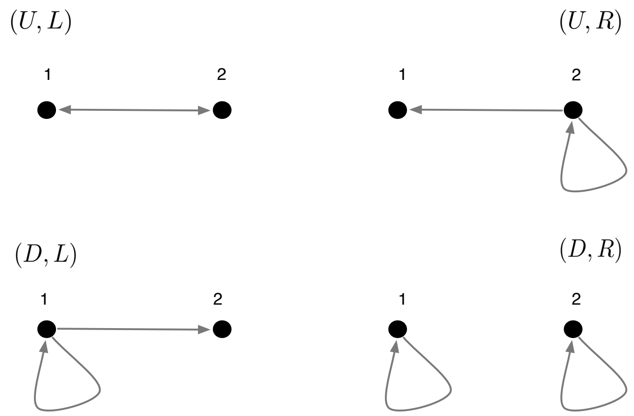


Figure 3: BR-dependences in the Prisoner Dilemma: 1 and 2 denote row and column.

Definition 11 (Permuted games) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game and $\mu : N \mapsto N$ a bijection on N. The μ-permutation of game $\mathcal{G}$ is the game $\mathcal{G}^\mu = (N^\mu, S^\mu, \Sigma^\mu, \succeq_i^\mu, o^\mu)$ such that: $N^\mu = N$; $S^\mu = S$; for all $i \in N$, $\Sigma_i^\mu = \Sigma_{\mu(i)}$; for all $i \in N$, $\succeq_i^\mu = \succeq_i$; $o_\mu : \bigtimes_{i \in N} \Sigma_{\mu(i)} \rightarrow S$ is such that $o_\mu(\mu(\sigma)) = o(\sigma)$ with $\mu(\sigma)$ denoting the permutation of σ according to μ.*

Intuitively, a permuted game $\mathcal{G}^\mu$ is therefore a game where the strategies of each player are redistributed according to μ in the sense that i's strategies become $\mu(i)$'s strategies, where players keep the same preferences over outcomes, and where the outcome function assigns same outcomes to same profiles.

Example 2 (Two horsemen [6]) *"Two horsemen are on a forest path chatting about something. A passerby M, the mischief maker, comes along and having plenty of time and a desire for amusement, suggests that they race against each other to a tree a short distance away and he will give a prize of \$100. However, there is an interesting twist. He will give the \$100 to the owner of the slower horse. Let us call the two horsemen Bill and Joe. Joe's horse can go at 35 miles per hour, whereas Bill's horse can only go 30 miles per hour. Since Bill has the slower horse, he should get the \$100. The two horsemen start, but soon realize that there is a problem. Each one is trying to go slower than the other and it is obvious that the race is not going to finish. [...] Thus they end up [...] with both horses going at 0 miles per hour. [...] However, along comes another passerby, let us call her S , the problem solver, and the situation is explained to her. She turns out to have a clever solution. She advises the two men to switch horses. Now each man has an incentive to go fast, because by making his competitor's horse go faster, he is helping his own horse to win!" [6, p. 195-196].*

Once the game of the example is depicted as the left-hand side game in Figure 4, it is possible to view the second passerby's solution as a bijection μ which changes the game to the right-hand side version. Now Row can play Column's moves and Column can play Row's moves. The result is a swap of (D, L) with (U, R), since (D, L) in $\mathcal{G}^\mu$ corresponds

	L	R
U	0, 0	1, 0
D	0, 1	1, 0

$\mathcal{G}$

L	R
0, 0	0, 1
1, 0	1, 0

$\mathcal{G}^\mu$

Figure 4: The two horsemen game and its transposition.

to (U, R) in $\mathcal{G}$ and vice versa. On the other hand, (U, L) and (D, R) stay the same, as the exchange of strategies do not affect them. As a consequence, profile (D, R), in which both horsemen engage in the race, becomes a dominant strategy equilibrium.

On the ground of these intuitions, it is possible to obtain a simple characterization of the different notions of reciprocity given in Definition 10 as the existence of equilibria in appropriately permuted games.

Theorem 1 (Reciprocity in equilibrium) *Let $\mathcal{G}$ be a game and (N, R^x_σ) be its dependence structure with $x \in \{BR, DS\}$ and σ be a profile. It holds that:*

i) *σ is x-reciprocal iff there exists a bijection $\mu : N \mapsto N$ s.t. σ is a x-equilibrium in the permuted game $\mathcal{G}^\mu$;*

ii)
- *σ is partially BR-reciprocal in C (or C-BR-reciprocal) iff there exists a bijection $\mu : C \mapsto C$ s.t. σ_C is a BR-equilibrium in the permuted subgame $(\mathcal{G} \downarrow \sigma_{\overline{C}})^\mu$;*
- *σ is partially DS-reciprocal in C (or C-DS-reciprocal) iff there exists a bijection $\mu : C \mapsto C$ s.t. σ_C is a DS-equilibrium in all permuted subgames $(\mathcal{G} \downarrow \rho_{\overline{C}})^\mu$, for $\rho_{\overline{C}} \in \bigtimes_{j \in \overline{C}} \rho_j$ and $\rho_j \in \Sigma_j$;*

iii) *σ is trivially x-reciprocal iff σ is an x-equilibrium in $\mathcal{G}^\mu$ where μ is the identity over N;*

iv) *σ is fully x-reciprocal iff there exists a bijection $\mu : N \mapsto N$ s.t. σ is a x-equilibrium in the permuted game $\mathcal{G}^\mu$ and μ is such that $\{(i, j) \mid i \in N \;\&\; j = \mu(i)\}$ is a Hamiltonian cycle in N.*

The proof is omitted for space reasons. From the foregoing result, it follows that permutations can be fruitfully viewed as ways of *implementing*—in a social software sense [6]—a reciprocal profile. This is terminology is worth casting in the following definition.

Definition 12 (Implementation as game permutation) *Let $\mathcal{G}$ be a game, (N, R^x_σ) be its dependence structure in σ with $x \in \{BR, DS\}$, and $\mu : N \mapsto N$ and $\mu' : C \mapsto C$ with $C \subseteq N$ be two bijections. We say that:*

i) *μ x-implements σ iff σ is an x-equilibrium in $\mathcal{G}^\mu$;*

ii)
- *μ' C-BR-implements σ iff σ_C is an BR-equilibrium in $(\mathcal{G} \downarrow \sigma_{\overline{C}})^{\mu'}$;*
- *μ' C-DS-implements σ iff σ_C is a DS-equilibrium in all $(\mathcal{G} \downarrow \rho_{\overline{C}})^{\mu'}$.*

Intuitively, implementation is here understood as a way of transforming a game in such a way that the desirable outcomes, in the transformed game, are brought about at an equilibrium point. In this sense we talk about *BR*- or *DS*-implementation. The difference between the two arises in the implementation of partial agreements where the locality of partial BR-reciprocity becomes apparent vis-à-vis the global character of partial DS-reciprocity.

3 Solving dependencies: dependence games

The previous section has shown how reciprocity can be given two corresponding formal characterization: existence of cycles in a dependence structure, and existence of equilibria in a suitably permuted game (Theorem 1). In the present section we take the notion of reciprocity as the basis upon which to define two new solution concepts, of a cooperative kind, for games in strategic form.

	L	R
U	2,2	0,3
D	3,0	1,1

$\mathcal{G}^{\mu}$

L	R
2,2	3,0
0,3	1,1

$\mathcal{G}^{\nu}$

Figure 5: Agreements between the prisoners

3.1 Agreements

The intuition is that, given a reciprocal profile (of some sort according to Definition 10), the players can fruitfully *agree* to transform the game by some suitable permutation of strategy sets.

Definition 13 (Agreements and partial agreements) *Let $\mathcal{G}$ be a game, (N, R^x_σ) be its dependence structure in σ with $x \in \{BR, DS\}$, and let $i, j \in N$. A pair (σ, μ) is:*

i) an x-agreement for $\mathcal{G}$ if σ is an x-reciprocal profile, and $\mu : N \mapsto N$ a bijection which x-implements σ;

ii) a partial x-agreement in C (or a C-x-agreement) for $\mathcal{G}$, if σ is a C-x-reciprocal profile and $\mu : C \mapsto C$ a bijection which C-x-implements σ.

The set of x-agreements of a game $\mathcal{G}$ is denoted x-AGR($\mathcal{G}$) and the set of partial x-agreements, that is the set of pairs (σ, μ) for which there exists a C such that μ C-x-implements σ, is denoted x-pAGR($\mathcal{G}$).

Intuitively, a (partial) agreement, of BR or DS type, can be seen as the result of coordination (endogenous, via the players themselves, or exogenous, via a third party like in Example 2) selecting a desirable outcome and realizing it by an appropriate exchange of strategies.

Example 3 (Agreements in PD) *In the game Prisoner's Dilemma two DS-agreements can be observed, whose permutations give rise to the games depicted in Figure 5. Agreement $((D, R), \mu)$ with $\mu(i) = i$ for all players, is the standard DS-equilibrium of the strategic game. But there is another possible agreement, where the players swap their strategies: it is $((U, L), \nu)$, for which $\nu(i) = N \backslash \{i\}$. Here Row plays cooperatively for Column and Column plays cooperatively for Row. Of the same kind is the agreement arising in Example 2. Notice that in such example, the agreement is the result of coordination mediated by a third party (the second passerby). Analogous considerations can also be done about Example 1 where, for instance, $((g, g, g), \mu)$ with $\mu(1) = 2, \mu(2) = 3, \mu(3) = 1$ is a BR- agreement.*

As we might expect, BR- and DS-agreements are related in the same way as BR- and DS-reciprocity (Fact 2). In what follows we will focus only on DS-agreements and partial DS-agreements so, whenever we talk about agreements and partial agreements, we mean DS-agreements and partial DS-agreements, unless stated otherwise.

3.2 Dominance

As there can be several possible agreements in a game, the natural issue arises of how to order them. We will do that by defining a natural notion of dominance between agreements, but first we need some auxiliary notions.

Definition 14 (C-candidates and C-variants) *Let $\mathcal{G} = (N, S, \Sigma_i, \geq_i, o)$ be a game and C a non-empty subset of N. An agreement (σ, μ) for $\mathcal{G}$ is a* C-candidate *if C is the union of some members of the partition induced by μ, that is: $C = \bigcup X$ where $X \subseteq P_\mu(N)$. An agreement (σ, μ) for $\mathcal{G}$ is a*

C-variant *of an agreement* (σ', μ') *if* $\sigma_C = \sigma'_C$ *and* $\mu_C = \mu'_C$*, where* μ_C *and* μ'_C *are the restrictions of* μ *to* C*. As a convention we take the set of* $\emptyset$*-candidate agreements to be empty and an agreement* (σ, ν) *to be the only* $\emptyset$*-variant of itself.*

In other words, an agreement (σ, μ) is a C-candidate if $\{C, \overline{C}\}$ is a bipartition of $P_\mu(N)$, and it is a C-variant of (σ', μ') if it differs from this latter at most in its C-part. We can now define the following notions of dominance between agreements and between partial agreements.

Definition 15 (Dominance) *Let* $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ *be a game and* $C \subseteq N$ *be a coalition. We say that:*

i) An agreement (σ, μ) *is dominated iff for some coalition* C *there exists a* C*-candidate agreement* (σ', μ') *for* $\mathcal{G}$ *such that for all agreements* (ρ, ν) *which are* $\overline{C}$*-variants of* (σ', μ')*,* $o(\rho) \succ_i o(\sigma)$ *for all* $i \in C$*.*

ii) A partial agreement (σ_C, μ) *in* C *is dominated iff for some coalition* $D \subseteq N$ *there exists* (τ_D, ν) *which is a* D*-agreement such that for all* σ', τ'*,* $o(\tau_D, \tau'_{\overline{D}}) \succ_i o(\sigma_C, \sigma'_{\overline{C}})$ *for all* $i \in D$*.*

The set of undominated agreements of $\mathcal{G}$ *is denoted* $DEP(\mathcal{G})$ *and the set of undominated partial agreements is denoted* $pDEP(\mathcal{G})$*.*

Intuitively, an agreement is undominated when a coalition C can force all possible agreements to yield outcomes which are better for all the members of the coalition, regardless of what the rest of the players can agree to do, that is, regardless of the $\overline{C}$-variants of their agreements. A partial agreement in coalition C is undominated when C can, by means of a partial permutation, force the game to end up in a set of states which are better for the member of the coalition no matter what the players in $\overline{C}$ do.

It is worth stressing the critical difference between the two notions of dominance. This difference resides in the fact that while dominance between agreements only considers deviations which are the results of agreements, dominance between partial agreements considers any form of possible deviation.

Example 4 (Dominance between partial agreements) *In the three persons Prisoner Dilemma (see Figure 1),* $((g_1, g_2), (\mu(1) := 2, \mu(2) := 1))$ *is a partial DS-agreement in* $\{1, 2\}$*. This agreement, which represents a form of dependence-based cooperation between* 1 *and* 2 *dominates the partial DS-agreement in* N*—on a trivially DS-reciprocal profile—*$((\neg g_1, \neg g_2, \neg g_3), (\mu(1) := 1, \mu(2) := 2, \mu(3) := 3))$*. In fact, it is undominated, since even the partial DS-agreement in* N $((g_1, g_2, g_3), (\mu(1) := 2, \mu(2) := 3, \mu(3) := 1))$ *(which is also a DS-agreement) does not dominate it.*

3.3 Dependence-based coalitional games

Now the question is, can we characterize the notion of dominance for agreements and partial agreements (Definition 15) in terms of a suitable notion of stability in appropriately defined games?

In order to answer this question we proceed as follows. First, starting from a game $\mathcal{G}$, we consider its representation $C^{\mathcal{G}}$ as a coalitional game as illustrated in Section 2.1 (Definition 5). As Definition 5 abstracts from dependence-theoretic considerations we refine it in two ways, corresponding to the two different sorts of dependence upon which we want to build the coalitional game:

1. The first refinement is obtained by defining a coalitional game $C^{\mathcal{G}}_{DEP}$ capturing the intuition that coalitions form only by means of *agreements* (Definition 13). Such games are called *dependence games*.

2. The second one is obtained by defining a coalitional game $C^{\mathcal{G}}_{pDEP}$ capturing the intuition that coalitions form only by means of *partial agreements* (Definition 13). Such games are called *partial dependence games*.

Having done this, we show that the core of $C^{\mathcal{G}}_{DEP}$ coincides with the set of undominated agreements of $\mathcal{G}$ (Theorem 2) and, respectively, that the core of $C^{\mathcal{G}}_{pDEP}$ coincides with the set of undominated partial agreements of $\mathcal{G}$ (Theorem 3). We thereby obtain a game-theoretical characterization of Definition 15.

3.3.1 Dependence games

Definition 16 (Dependence games from strategic ones) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game. The dependence game $C^{\mathcal{G}}_{DEP} = (N, S, E^{\mathcal{G}}_{DEP}, \succeq_i)$ of $\mathcal{G}$ is a coalitional game where the effectivity function $E^{\mathcal{G}}_{DEP}$ is defined as follows:*

$$
\begin{aligned}
X \in E^{\mathcal{G}}_{DEP}(C) \quad \Leftrightarrow \quad & \exists \sigma_C, \mu_C \text{ S.T.} \\
& \exists \sigma_{\overline{C}}, \mu_{\overline{C}} : [((\sigma_C, \sigma_{\overline{C}}), (\mu_C, \mu_{\overline{C}})) \in AGR(\mathcal{G})] \\
& \text{AND } [\forall \sigma_{\overline{C}}, \mu_{\overline{C}} : [((\sigma_C, \sigma_{\overline{C}}), (\mu_C, \mu_{\overline{C}})) \in AGR(\mathcal{G}) \\
& \text{IMPLIES } o(\sigma_C, \sigma_{\overline{C}}) \in X]].
\end{aligned}
$$

where $\mu : N \rightarrow N$ is a bijection.

This somewhat intricate formulation states nothing but that the effectivity function $E^{\mathcal{G}}_{DEP}(C)$ associates with each coalition C the states which are outcomes of agreements (and hence of reciprocal profiles), and which C can force via partial agreements (σ_C, μ_C) regardless of the partial agreements $(\sigma_{\overline{C}}, \mu_{\overline{C}})$ of $\overline{C}$.

We have the following theorem.

Theorem 2 (*DEP* vs. *CORE*) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game. It holds that, for all agreements (σ, μ):*

$$(\sigma, \mu) \in DEP(\mathcal{G}) \ \Leftrightarrow \ o(\sigma) \in CORE(C^{\mathcal{G}}_{DEP}).$$

where $\mu : N \rightarrow N$.

The proof is omitted for space reasons. Put it otherwise, here is what Theorem 2 states. Given a game $\mathcal{G}$, a profile σ which is partially DS-implemented by μ (Definition 12) forms an undominated partial agreement (σ, μ) if and only if σ is in the core of the dependence game of $\mathcal{G}$. By taking Definition 10 and Theorem 1 into the picture, we thus see that Theorem 2 connects three apparently rather different properties of a strategic game $\mathcal{G}$: the existence of reciprocal profiles, the existence of DS-equilibria in permutations of $\mathcal{G}$, and the core of the dependence game built on $\mathcal{G}$.

3.3.2 Partial dependence games

Definition 17 (Partial dependence games from strategic ones) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game. The partial dependence game $C^{\mathcal{G}}_{pDEP} = (N, S, E^{\mathcal{G}}_{pDEP}, \succeq_i)$ of $\mathcal{G}$ is a coalitional game where the effectivity function $E^{\mathcal{G}}_{pDEP}$ is defined as follows:*

$$
\begin{aligned}
X \in E^{\mathcal{G}}_{pDEP}(C) \quad \Leftrightarrow \quad & \exists \sigma_C, \mu_C \text{ S.T.} \\
& (\sigma_C, \mu_C) \in pAGR(\mathcal{G}) \\
& \text{AND } [\forall \sigma_{\overline{C}} : o(\sigma_C, \sigma_{\overline{C}}) \in X]].
\end{aligned}
$$

where $\mu_C : C \rightarrow C$ is a bijection.

Partial dependence games are defined by just looking at the set of outcomes that each coalition can force by means of a partial agreement. Unlike Definition 16, Definition 17 is much closer to the standard definition of coalitional games based on strategic ones (Definition 5).

Like for dependence games, we have a characterization of the set of undominated partial agreements.

Theorem 3 (*pDEP* vs. *CORE*) *Let $\mathcal{G} = (N, S, \Sigma_i, \succeq_i, o)$ be a game. It holds that, for all agreements (σ, μ):*

$$(\sigma, \mu) \in pDEP(\mathcal{G}) \;\Leftrightarrow\; o(\sigma) \in CORE(C^{\mathcal{G}}_{pDEP}).$$

where $\mu : C \to C$ is a bijection with $C \subseteq N$.

The proof is omitted for space reasons. Like Theorem 2, Theorem 3 establishes a precise connection between the notions of partial reciprocity in a strategic game $\mathcal{G}$, the existence of DS-equilibria in all permuted subgames of $\mathcal{G}$, and the core of the partial dependence game built on $\mathcal{G}$.

3.4 Coalitional, dependence, partial dependence effectivity

The coalitional game $C^{\mathcal{G}}$ built on a strategic game $\mathcal{G}$ and its dependence-based counterparts $C^{\mathcal{G}}_{DEP}$ and $C^{\mathcal{G}}_{pDEP}$ are clearly related. The following fact shows how.

Fact 3 (Effectivity functions related) *The following relations hold:*

i) For all $\mathcal{G}$: $E^{\mathcal{G}}_{pDEP} \subseteq E^{\mathcal{G}}$;

ii) It does not hold that for all $\mathcal{G}$: $E^{\mathcal{G}}_{DEP} \subseteq E^{\mathcal{G}}_{pDEP}$; nor it holds that for all $\mathcal{G}$: $E^{\mathcal{G}}_{pDEP} \subseteq E^{\mathcal{G}}_{DEP}$;

iii) It does not hold that for all $\mathcal{G}$: $E^{\mathcal{G}}_{DEP} \subseteq E^{\mathcal{G}}$; nor it holds that for all $\mathcal{G}$: $E^{\mathcal{G}} \subseteq E^{\mathcal{G}}_{DEP}$.

The proof is omitted for space reasons. The fact shows that dependence games are not just a refinement of coalitional ones, which instead holds for partial dependence games. In other words dependence-based effectivity function considerably modify the powers assigned to coalitions by the standard definition of coalitional games on strategic ones (Definition 5).

4 Conclusions

The contribution of the paper is two-fold. On the one hand it has been shown that central dependence-theoretic notions such as the notion of cycle are amenable to a game-theoretic characterization (Theorem 1). On the other hand dependence theory has been demonstrated to give rise to types of cooperative games where solution concepts such as the core can be applied. The relation between the various forms of cooperative games where coalitions undertake agreements (dependence and partial dependence) have been analyzed, together with the dominance they induce on agreements (Theorem 2 and 3).

Acknowledgments

The authors would like to thank the anonymous reviewers of AAMAS'10, where a previous version of the work has been published, for the helpful remarks which have been taken up in the present version of the paper. Davide Grossi acknowledges support by the *Nederlandse Organisatie voor Wetenschappelijk Onderzoek* under the NWO VENI grant 639.021.816.

References

[1] G. Boella, L. Sauro, and L. van der Torre. Strengthening admissible coalitions. In *Proceeding of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence*, pages 195–199. ACM, 2006.

[2] C. Castelfranchi, A. Cesta, and M. Miceli. Dependence relations among autonomous agents. In E. Werner and Y. Demazeau, editors, *Decentralized A.I.3*. Elsevier, 1992.

[3] D. Grossi and P. Turrini. Dependence theory via game theory. In W. van der Hoek and G. Kaminka, editors, *Proceedings of AAMAS 2010*, 2010.

[4] H. Moulin and B. Peleg. Cores of effectivity functions and implementation theory. *Journal of Mathematical Economics*, 10:115–145, 1982.

[5] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[6] R.Parikh. Social software. *Synthese*, 132(3):187–211, 2002.

Davide Grossi
Institute for Logic, Language and Computation
University of Amsterdam
Science Park 904
Amsterdam, 1098 XH, The Netherlands
Email: `d.grossi@uva.nl`

Paolo Turrini
Department of Information and Computing Sciences
Utrecht University
Padualaan 14, De Uithof
3584 CH, Utrecht
Email: `paolo@cs.uu.nl`

Optimization of Payments in Dominant Strategy Mechanisms for Single-Parameter Domains[1]

Victor Naroditskiy, Maria Polukarov, and Nicholas R. Jennings

Abstract

The paper studies dominant strategy mechanisms in anonymous single-parameter allocation domains with monetary payments. Given a mechanism design problem with a fixed allocation function (e.g., efficient allocation), we seek an optimal payment function. Restricting attention to "constant-dependent" allocation functions, we present a general technique for finding an optimal payment function for any mechanism design problem. By construction, the optimal payment function is piecewise linear, proving the existence of piecewise linear optimal payment functions.

1 Introduction

Mechanism design, traditionally studied in economics, is now a rapidly growing field in computer science (e.g., see [12, 16]). Generally, it deals with problems where multiple self-interested participants take actions to optimize individual utilities based on the *incentives* provided to them. The overlap of computer science and mechanism design is natural: reasoning about incentives is unavoidable in many fundamental computer science problems—e.g., peer-to-peer networks, packet routing, and resource allocation. To this end, the term *algorithmic mechanism design* was introduced by Nisan and Ronen [11] to refer to the study of incentives in computer science scenarios. At the same time, some classic mechanism design solutions rely on computationally efficient approximations and implementations to be of practical use (e.g., computationally efficient VCG-like combinatorial auctions [5]).

This paper interfaces computer science and mechanism design in yet another way: instead of considering a particular domain and designing a mechanism with certain properties, we are looking for a general, unified technique that takes a mechanism design problem as an input, and outputs an optimal mechanism to this problem. While this goal, in general, may sound unrealistic, this work shows it can be effectively achieved for a wide class of problems in *single-parameter domains* where agents have private types expressed by single numbers. Examples include recently studied problems of surplus-maximizing allocation of free resources [10, 7] and fair task imposition [13] as well as problems of surplus-maximization and fairness in more general models where resources are not free. Furthermore, our—at first glance, purely algorithmic—approach enables derivation of theoretical results that provide a base for the following contributions:

- **Characterization.** We formulate sufficient conditions for the existence of *(piecewise) linear* optimal mechanisms in single-parameter domains (Theorem 2).
- **Existence.** We identify a class of mechanism design problems characterized by a *constant-dependent* allocation function (to be defined), and prove the above conditions hold for each problem in this class (Theorem 3).
- **Construction.** We develop an algorithm that finds an optimal mechanism for any given problem in this class (Theorem 2 and Figure 4).

[1] Helpful discussions with Geoffroy de Clippel, Sergey Kushnarev, Lyle Ramshaw, Warren Schudy, and Meinolf Sellmann are gratefully acknowledged.

We start with preliminaries and related work in Section 2, and present our main theorems in Section 3. These results provide a general and powerful tool for analysis of mechanism design problems in single-parameter domains: as the existence and construction results apply to *any* problem with constant-dependent allocation, the algorithm for finding an optimal mechanism is unified within this class of problems. In Section 4, we demonstrate the strength of this approach on two central mechanism design problems of (i) surplus-maximizing resource allocation and (ii) fair task imposition. First, we re-derive the mechanisms for surplus-maximizing allocation of free items by Moulin [10] and Guo and Conitzer [7] and fair imposition of a single task by Porter *et al.* [13]. We note that our solution is not analytical but algorithmic; that is, we provide systems of linear equations whose—*unique*—solutions coincide with the mechanisms obtained in [10, 7] and [13], for each particular problem instance. However, in the latter case, this approach also allowed us to easily find an analytical solution. Second, we use our method to obtain an optimal mechanism for fair imposition of multiple tasks, for which no closed form has been previously found. Furthermore, in Section 5 we apply our technique to open problems. Specifically, we extend the consideration to scenarios where objects have costs and provide first algorithms for computing optimal mechanisms for surplus maximizing allocation and fair imposition in these generalized settings. Finally, our work suggests several directions for future research outlined in Section 6.

2 Preliminaries and Related Work

Informally, a *mechanism* refers to a procedure for making decisions (or, choices) involving multiple agents. Suppose one item needs to be allocated among a group of agents. A mechanism might collect bids from each agent, give the item to the highest bidder, and charge him his bid: this mechanism is a first-price auction, and the choice made defines an allocation.

Mechanism design is concerned with finding the best way of making decisions in a given scenario (e.g., allocation). The "best" way is specified by properties the decision must satisfy: e.g., the mechanism should be fair, the agent with the highest value should be allocated, the revenue of the seller should be maximized. Crucially, quality of a decision depends on private information called *types* of the agents (e.g., their values for the item). Therefore, a mechanism must ensure the agents have the incentive to reveal their types truthfully: without knowing the true types, there is no way to know how good a decision is. We study *strategy-proof* mechanisms that make it in each agent's best interest to truthfully reveal his type—regardless of whether the other agents do so or not. This—the strongest—concept of truthfulness is called *dominant strategy*.

Implementation in dominant strategies is virtually impossible without restrictions on agents' types. In fact, for unrestricted types (i.e., different value for each possible choice) only dictatorial choice functions are implementable [6]. One way to get out of the impossibility is by introducing monetary payments which are added to agent's valuation of the chosen alternative: in this case, agents are said to have *quasi-linear utilities*. A mechanism is therefore defined by a choice rule f and a payment scheme t—both are functions of the agents' types.

However, even with money (a.k.a. transferable utilities), the set of implementable mechanisms is rather limited—the only such mechanisms are weighted VCG [14], which motivates further restrictions on agents' types. In this work, we focus on single-parameter domains: we restrict our attention to allocation domains where agent's type represents the value for being allocated—in this context, it is intuitive to refer to the choice rule as the *allocation function*. It is known that in these settings, any allocation function that is monotone[2] in the agent's report, is implementable:

Theorem 1 (e.g., see [12] p. 229) *A mechanism (f, t) is implementable if and only if for each agent i:* (i) *f_i is monotone in v_i;* (ii) *$t_i = h(v_{-i}) - \tau(v_{-i})$ if $f_i = 1$ (i.e., i is allocated) and $t_i = h(v_{-i})$*

[2] In words, monotonicity of f_i in v_i means that if an agent is allocated when he reports v_i he is also allocated when he reports $v'_i \geq v_i$.

otherwise ($f_i = 0$), where $\tau(v_{-i}) = \sup_{v_i | f_i(v_i, v_{-i})=0} v_i$ defines the critical value.[3,4]

Thus, any pair of functions (f, t) that satisfy the conditions in Theorem 1, defines a strategy-proof (i.e., truthful) mechanism. In this work, we take a monotone allocation function as an input and look for an optimal (according to provided properties) payment function of the form above. We develop a general algorithmic method for finding optimal payments for an important class of allocation functions we term "constant-dependent": in particular, *efficient*[5] allocation functions in settings of consideration fall in this class.

A dominant strategy mechanism does not make any assumptions about the values of the agents: desirable properties (e.g., efficiency, individual rationality, no subsidy) of the mechanism must hold for all possible values the agents may have. These properties can be expressed as a system of constraints to be met for each possible profile v of agents' valuations and (optionally) an objective function (e.g., revenue maximization). In this work, we take the allocation function as an input and optimize the payment function. Using the characterization above, the only degree of freedom in designing payments is the function $h : \mathbb{R}^{n-1} \to \mathbb{R}$. Thus, the problem of finding optimal payments can be stated as

$$\begin{aligned} &\text{optimize}_{h:\mathbb{R}^{n-1}\to\mathbb{R}} \text{ objective value} \quad \text{s.t.} \quad \forall\, v \in \mathbb{R}^n \\ &\quad \text{objective value is achieved} \\ &\quad \text{properties hold} \end{aligned}$$

At the first glance, this problem is hard: optimization is over functions and there is an infinite number of constraints. However, in this paper we propose an algorithmic approach that makes it possible to effectively tackle such problems. Our technique exploits and makes explicit the linearity structure present in many standard mechanism design problems. As a result, the existence of (piecewise) linear optimal payment functions follows immediately. For some important allocation problems, this method offers an easy way to find the optimal payment function analytically, by solving a simple system of linear equations (see, for example, the results in 4.2). For a general problem in this class, it provides an algorithm for finding an optimal mechanism computationally.

Most related to our work is the literature on optimizing rebates in VCG mechanisms. In [10] and [7], the authors independently discover the optimal VCG redistribution mechanism for allocating free homogenous items. VCG redistribution schemes have also been designed for a public good domain [1]. A similar result has been derived in [2] in the context of allocating a single item. An alternative objective of fairness was considered in [13] for task imposition scenarios. In this paper, we provide a general approach for addressing all of these problems.

The model of allocating homogeneous items that have costs was considered in [3, 9], although for a different purpose—to compare "random priority" and "average cost" mechanisms. We are the first to obtain (algorithmic) solutions for surplus-maximization and fairness in this setting.

We are aware of only one other attempt to approach mechanism design problems algorithmically—that of Automated Mechanism Design (AMD) [15]. However, AMD applies when the space of agents' types is finite and a prior over the types is available. In contrast, we deal with infinite types spaces and no priors. Our method is based on partitioning the space of value profiles into a finite number of convex regions, on each of which, as we prove, a linear optimal payment function can be defined. A similar idea was exploited in [4]; however, there the partitioning is heuristic and does not result in an optimal mechanism.

[3] An agent i is allocated if and only if his report is above the critical value $\tau(v_{-i})$.

[4] In stating the theorem, we restricted attention to *anonymous* payment functions: payment functions that do not depend on agent's identity.

[5] An allocation is efficient if the items are assigned to the agents who value them the most.

3 Optimization, Linearity and Partition

In this section, we present our main results: Theorem 2 provides sufficient conditions for the existence of a piecewise linear optimal mechanism, and Theorem 3 constructively proves these conditions hold for the class of constant-dependent allocations (to be defined). We start by formally stating the problem in 3.1 and explaining the idea of our solution in 3.2. The theorems appear in 3.3 and 3.4.

3.1 Setting

We consider single-parameter domains where each of n agents desires one unit of a (homogenous) good, and $v \in \mathbb{R}^n_+$ represents the agents' valuations for consuming the good (or, *item*). Monetary transfers are possible, and agents' utilities are quasi-linear. The value profiles are such that $v_1 \geq v_2 \geq \ldots \geq v_n$ (this is without loss of generality for anonymous mechanisms), and since the values are non-negative, one can scale all vectors to be in the interval $[0, 1]$. We denote the space of value profiles by $V = \{v \in \mathbb{R}^n \mid 1 \geq v_1 \geq v_2 \geq \ldots \geq v_n \geq 0\}$ and define vector $v_{-i} \in \mathbb{R}^{n-1}$ to indicate values of the agents other than i. The space of all such $(n-1)$-dimensional vectors is the same for each i, and is denoted by W.

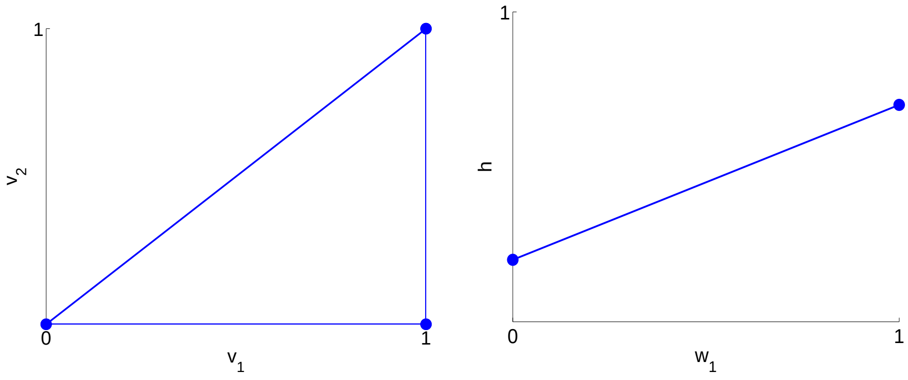
An *outcome* is a pair $(\mathrm{f}, \mathrm{t}) \in \{0, 1\}^n \times \mathbb{R}^n$, where f_i indicates whether agent i is *allocated* (gets the item), and t_i represents the *payment* he receives (t_i can be negative, in which case agent i pays that amount); the total utility of agent i from the outcome (f, t) is given by $u_i = \mathrm{f}_i v_i + \mathrm{t}_i$. A *mechanism* is defined by a pair of functions $f : \mathbb{R}^n_+ \rightarrow \{0, 1\}^n$ and $t : \mathbb{R}^n_+ \rightarrow \mathbb{R}^n$ that determine the allocation and payments for each possible report from the agents regarding their value for the item. We take as an input an allocation function f satisfying the monotonicity condition in Theorem 1. This determines the critical value, $\tau(v_{-i})$, for each agent i, and the only remaining degree of freedom is the function $h(v_{-i})$ that adjusts payments to the agents. In some applications, it is intuitive to view τ as the price for being allocated and h as the rebate distributed back to all agents; henceforth, we refer to h as the *rebate function*.

Our goal is to find *optimal rebates* that guarantee the best possible value of a given objective function and satisfy given constraints for each possible vector of agents' valuations—that is, provide a dominant strategy implementation. The objective of optimization may be, for example, maximization of social surplus (i.e., redistributing back as much of the budget surplus as possible when there is no auctioneer), some measure of fairness (e.g., maximizing the lowest utility), or minimization of budget deficit. Desirable properties of mechanisms (e.g., no subsidy, individual rationality, k-fairness) are specified as constraints in the optimization problem. Some combinations of properties (e.g., no subsidy and 2-fairness) may be impossible to implement: this is identified by the lack of a feasible solution to the optimization problem.

3.2 Linear properties

Our approach exploits the linear structure which characterizes standard mechanism design problems. Typical constraints (e.g., individual rationality, no subsidy, k-fairness) and objectives (e.g., utility maximization, deficit minimization) are linear in values and payments of the agents. For example, the no subsidy (or weak budget balance) constraint requires the *sum* of payments to the agents to be non-positive; utilitarian objective function maximizes the *sum* of agents' values and payments. This linearity structure lies at the heart of the idea presented next.

Consider the following illustrative example. Recall that we are after an optimal rebate function $h : \mathbb{R}^{n-1} \rightarrow \mathbb{R}$. In the simplest case with 2 agents, the domain of the rebate function is the real interval between 0 and 1. The space of values is a *triangle* given by the extreme points $(0, 0)$, $(1, 0)$, and $(1, 1)$ shown in Figure 1(a)—recall that the value vectors are non-decreasing, and thus only the bottom half of the unit square is relevant. Suppose that allocation is fixed for all profiles of values



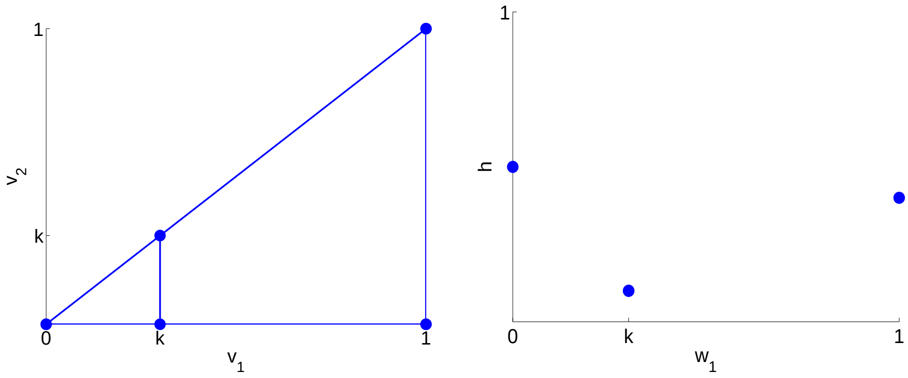
(a) Allocation region.

(b) Rebate function.

Figure 1: Single allocation region and optimal linear rebate function.

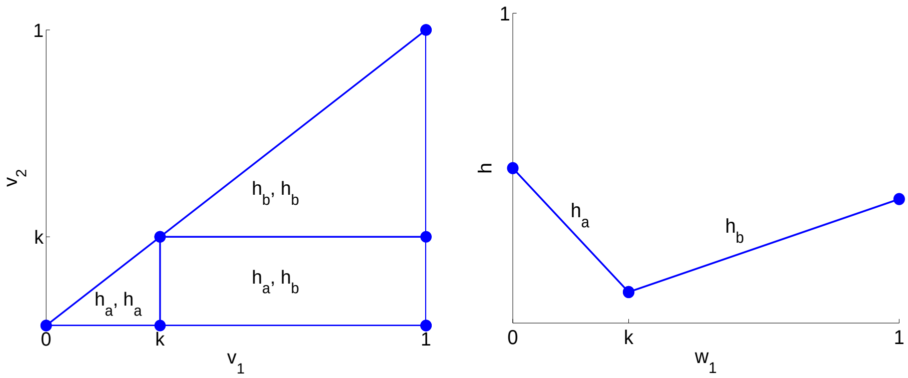
(e.g., agent 1 is always allocated and agent 2 is never allocated) and that constraints are linear in v and h. Indeed, h is a function of v itself, but it is a variable in our problem. For a finite set of points $v \in V$ we can talk of a finite set of values of the rebate function $h(w)$, where w corresponds to v_{-i} for agent i. It is easy to show that a linear constraint is satisfied everywhere on a convex region if and only if it is satisfied on its extreme points: in our example, enforcing linear constraints on the profiles $(0, 0)$, $(1, 0)$ and $(1, 1)$ guarantees that they are satisfied for all profiles $v \in \mathbb{R}^2 \mid 1 \geq v_1 \geq v_2 \geq 0$. Note that constraints for these profiles involve exactly two rebates $h(0)$ and $h(1)$: thereby, restricting the optimization problem to constraints for these extreme profiles gives a linear program with two variables, which we call $\hat{h}(0)$ and $\hat{h}(1)$. Also, since this restricted linear program includes only a subset of constraints from the original mechanism design problem, its optimal objective value provides an upper (in the case of maximization) bound on the objective value of the original problem (in problems with no objectives, if the original problem has a feasible solution, so does the restricted problem.) Now, having solved the *restricted problem* by computing the rebates $\hat{h}(0)$ and $\hat{h}(1)$, the equation of the line on which these two points lie provides us with a—linear—rebate function h: that is, for an arbitrary point $w \in W$, we can define $h(w) = a_1 w_1 + b$, where the coefficients a_1, b are obtained by solving the system of two linear equations: $\hat{h}(0) = a_1 0 + b$ and $\hat{h}(1) = a_1 1 + b$. The rebate function is the line segment connecting points $(0, \hat{h}(0))$ and $(1, \hat{h}(1))$ (see Figure 1(b)). *This function is linear in v, so all constraints remain linear.* These constraints are satisfied on the extreme points of a convex region, and therefore hold everywhere on this region. Thus, we can "expand" an optimal solution to the restricted problem to a feasible solution to the original problem, and achieve the same objective value. Since the objective value of the restricted problem was an upper bound on the objective of the original problem, the constructed solution is optimal and the upper bound it tight. Finally, note that we were able to linearly combine rebate values $h(0)$ and $h(1)$ in the rebate space because there were exactly two (i.e., n) of them.

In more general cases, allocation may not be linear on the whole value space. For instance, consider the allocation rule that allocates to agent 1 if his value is above $k \in (0, 1)$ and never allocates to agent 2. The value space is partitioned into 2 allocation regions: agent 1 is not allocated in the region to the left of $v_1 = k$ and is allocated in the region to the right (see Figure 2(a)). Constraints for the extreme points $(0, 0)$, $(k, 0)$, $(1, 0)$, (k, k), and $(1, 1)$ of the allocation regions include three rebates $\hat{h}(0)$, $\hat{h}(k)$, and $\hat{h}(1)$. Proceeding as we did in the previous example we would have to linearly connect the values of these rebates. However, in general, three points do not lie on the same line, as is illustrated in Figure 2(b). A natural idea is to define two linear rebate functions: one connecting $\hat{h}(0)$ to $\hat{h}(k)$ and the other connecting $\hat{h}(k)$ to $\hat{h}(1)$ (see Figure 3(b)). We refer to these functions as h_a and h_b: thus, $h(w) = h_a(w)$ if $0 \leq w_1 \leq k$ and $h(w) = h_b(w)$ if $k \leq w_1 \leq 1$.



(a) Two allocation regions. (b) Relevant rebate values.

Figure 2: Two allocation regions and rebates in corresponding extreme points.



(a) Value regions. (b) Rebate function.

Figure 3: Refinement of allocation regions and optimal piecewise linear rebate function.

In order for constraints to be linear on a region, for each agent i the choice of the rebate function (h_a or h_b) must be constant throughout the region. The allocation region to the right of $v_1 = k$ does not satisfy this condition: the rebate for agent 1 is given by h_a for $v_2 \leq k$ and by h_b for $v_2 \geq k$. However, we can *refine* the allocation regions along $v_2 = k$ to fix this problem. In Figure 3(a), the regions are labeled with the rebate function used by each agent. Partitioning along $v_2 = k$ introduced a new extreme point: $(1, k)$. However, $\hat{h}(0)$, $\hat{h}(k)$, and $\hat{h}(1)$ are still the only rebates used by constraints on the extreme points. Two line segments passing through the points $(0, \hat{h}(0))$, $(k, \hat{h}(k))$ and $(k, \hat{h}(k))$, $(1, \hat{h}(1))$, respectively, define the—piecewise linear—rebate function (see Figure 3(b)). As before, this implies that the constraints are linear in v on each region of this—refined—partition, and since they are satisfied on the extreme points of each region, they hold for all points of each region.

Next, we generalize this idea and formalize conditions on partitions into regions of the value space V and the rebate space W, which we prove to be sufficient for the existence of an optimal mechanism with piecewise linear rebates.

3.3 Linearly consistent partitions

We need the following definitions.

Definition 1 *A set P_X of polytopes is called a* partition *of the polytope X if the polytopes do not overlap: $p \cap q = \varnothing,\ \forall\, p, q \in P_X$, and cover exactly the polytope X: $\bigcup_{p \in P_X} p = X$.*

Definition 2 *The partition P_X* refines *the set of polytopes Q if for all $p \in P_X,\ q \in Q$, their intersection is either empty or p: $p \cap q = \varnothing \vee p \cap q = p$.*

We are interested in partitions that consist of convex polytopes. A convex d-dimensional polytope p can be defined as a finite intersection of halfspaces: $p = \{x \in \mathbb{R}^d \mid Ax \geq b\}$, where $A \in \mathbb{R}^{k \times d}$, $b \in \mathbb{R}^k$, k is the number of halfspaces.[6] In the examples we provided, the rebate space is 1-dimensional and intersections of halfspaces specify line segments. The partition P_W in Figure 3(b) is given by 2 polytopes $k \geq w_1 \geq 0$ and $1 \geq w_1 \geq k$. Recall the corresponding partition P_V of the value space V in Figure 3(a). Crucially, for each agent i, the choice of the rebate function is fixed on each *value region* $q \in P_V$ (see Figure 3(a)). Stating the property mathematically, we obtain $\forall q \in P_V,\ \forall i \in \{1, \ldots, n\}$ there exists $p \in P_W \mid v_{-i} \in p,\ \forall v \in q$. Observation 1 below characterizes the partitions of the value space that satisfy this property: it notes that P_V must refine a set of polytopes *lift*(P_W) which is obtained by "lifting" the partition P_W of the $n-1$ dimensional space W to the n-dimensional space V.

Definition 3 *A set of polytopes* $\mathrm{lift}(P_W)$ *in the value space V is said to be obtained by* lifting *the partition P_W of the rebate space W if*

$$\mathrm{lift}(P_W) = \bigcup_{(A,b) \in P_W} \bigcup_{i=1}^{n} V \cap (Av_{-i} \geq b)$$

Note that each polytope $(A, b) \in P_W$ adds n (possibly overlapping) polytopes to *lift*(P_W). In our example in Figures 3(a) and 3(b), lifting the polytope $k \geq w_1 \geq 0$ yields overlapping polytopes $k \geq v_1 \geq v_2 \geq 0$ and $1 \geq v_1 \geq v_2 \geq 0;\ k \geq v_2$.

Observation 1 *Let P_V and P_W be partitions of the value and the rebate space, respectively. The condition $\forall q \in P_V,\ \forall i \in [1, n],\ \exists p \in P_W \mid v_{-i} \in p,\ \forall v \in q$ is satisfied when P_V refines* $\mathrm{lift}(P_W)$.

Next we derive additional conditions that would let us define rebate functions h_p for *rebate regions* $p \in P_W$ so that each such function is linear and $\{h_p \mid p \in P_W\}$ is optimal. For linearity, we need each polytope p in the rebate partition to have exactly n extreme points. For optimality, we need to guarantee the linearity of constraints on each of the value regions. This is formally stated in Definition 4 and below. Finally, Theorem 2 shows the sufficiency of these conditions.

We refer to the union of extreme points of the partition P_X of a polytope X as $\hat{P}_X$. Given the partition P_V of the value space, the *projection* of its extreme points into the rebate space W is defined as follows:

$$\Pi_W(\hat{P}_V) = \bigcup_{v \in \hat{P}_V} \bigcup_{i=1}^{n} v_{-i}$$

Definition 4 *Partitions P_V and P_W are called* linearly consistent *if:* (i) *all polytopes $q \in P_V$ and $p \in P_W$ are convex;* (ii) $\Pi_W(\hat{P}_V) = \hat{P}_W$*;* (iii) *$P_V$ refines both the set of polytopes* $\mathrm{lift}(P_W)$ *and the allocation partition P_V^a as defined by the allocation function f[7]; and* (iv) *each polytope in P_W has n extreme points.*

Consider the graph of a rebate function $(w, h(w))$. Note that any n rebate values can be described by a linear rebate function: indeed, there exists an $(n-1)$ dimensional hyperplane passing through n points in $\mathbb{R}^n$. Therefore, partitioning the rebate space into polytopes each having n extreme points lets us define a linear rebate function on each polytope. Finally, setting the rebates on extreme points $\hat{P}_W$ accordingly to the optimal solution to the optimization problem, restricted to $\hat{P}_V$, implies the optimality of rebates for linearly consistent partitions.

[6] Thus, a pair (A, b) defines a polytope in $\mathbb{R}^d$.

[7] That is, for any v^1, v^2 in the same allocation region $q^a \in P_V^a$, $f(v^1) = f(v^2)$.

Theorem 2 *Given linearly consistent partitions P_V and P_W, let $\{\hat{h}(w) \mid w \in \hat{P}_W\}$ denote the set of rebates from an optimal solution to the restricted problem and let $\hat{p}$ denote the set of n extreme points of a polytope $p \in P_W$. For each polytope, define a linear rebate function $h_p(w) = \sum_{i=1}^{n-1} a_i^p w_i + b^p$ with coefficients $a^p \in \mathbb{R}^{n-1}, b^p \in \mathbb{R}$ given by a solution to the system of linear equations $\{\hat{h}(w) = \sum_{i=1}^{n-1} a_i^p w_i + b^p \mid w \in \hat{p}\}$. Then, the following rebate function is* optimal*: for $w \in p$, $h(w) = h_p(w)$.*

By Theorem 2, if one can partition spaces V and W in a linearly consistent way, an optimal, piecewise linear, mechanism follows immediately. Next, we present an algorithm for finding such partitions for an important class of, what we call, *constant-dependent* allocation functions: these, in particular, include commonly desirable efficient allocations.

3.4 Constant-dependent allocations

We start with a definition.

Definition 5 *An allocation function is called constant-dependent if there exists a finite set of constants $C = \{c_1, \ldots, c_q\}$, such that the allocation is constant on each of the regions defined by hyperplanes of the form $v_i = c \mid c \in C$. For $C = \varnothing$, the allocation is constant on the whole space of agents' valuations.*

In Figure 4, we present the *partition* algorithm and show that it defines linearly-consistent partitions of the value and rebate spaces for a given constant-dependent allocation function (Theorem 3).

Algorithm *partition*
Input: polytope X
1. partition X along $x_i = c \quad \forall c \in C,\ i \in \{1, \ldots, dim(X)\}$ /* denote the partition by P_X^g */
2. **for** each hyperrectangle $p \in P_X^g$
 for each pair (i, j) of dimensions $i, j \in \{1, \ldots, dim(X)\},\ i \neq j$,
  partition p along $x_i = ax_j + b$ where $a, b \in \mathbb{R}$ define the diagonal
  from the lower left to the upper right corner of projection onto the i-j plane

Figure 4: Linearly-consistent partitions.

Theorem 3 *For a constant-dependent allocation, the partitions $P_V =$ partition(V) and $P_W =$ partition(W) are linearly-consistent.*

Constant-dependent allocation functions may not be monotone. Since a dominant-strategy implementation is possible only for monotone allocation functions (see Theorem 1), we only consider the ones that are.

In the following sections, we demonstrate that the algorithmic technique described in Theorems 2 and 3 can be applied to a wide class of mechanism design problems. In particular, we consider the surplus-maximizing allocation and fair imposition problems and show that our method provides an easy way of obtaining mechanisms for the (previously studied) case with free objects. Moreover, uniqueness of the mechanisms follows immediately from the uniqueness of the optimal solution to the restricted problem. These results are presented in Section 4. Finally, in Section 5, we extend the consideration to the open problem where items have costs.

4 "Free" Homogeneous Objects

In this section, we apply our technique to two central mechanism design problems in single-parameter domains. We start by re-deriving the results on surplus-maximizing allocation of free items by Moulin [10] and Guo and Conitzer [7] and fair imposition of a single task by Porter *et al.* [13]. We then proceed to show that an optimal mechanism for fair imposition of multiple tasks, for which no closed form has been previously derived, can be easily obtained using our method.

4.1 Surplus-maximizing allocation

See online appendix at http://users.ecs.soton.ac.uk/vn/comsoc.pdf

4.2 Fair imposition

See online appendix at http://users.ecs.soton.ac.uk/vn/comsoc.pdf

5 Allocation with Costs

In this section, we apply our technique to solve open mechanism design problems. Specifically, we consider more realistic scenarios where items are not free. This generalization significantly complicates the setting for both surplus-maximizing allocation and fair imposition problems, which have not been previously tackled for items with costs. We observe that the generalized model still falls in the framework of single-parameter domains with constant-dependent allocation, and Theorem 3 holds. Given this, we provide the first algorithm for computing optimal mechanisms for these scenarios.

5.1 Motivation

We consider a setting where (identical) items must be assigned to the agents, assuming each agent wants exactly one item, and the items have (increasing marginal) costs. The goal, as before, may be either to maximize the social surplus or to achieve k-fairness.

The allocation problem with increasing marginal costs is a simple and fundamental example of the problem of the commons [8], in which multiple participants, acting independently to optimize their own objectives, will ultimately deplete a shared limited resource even when it is clear that it is not in anyone's long-term interest for this to happen. Increasing marginal costs model decreasing returns to every agent as the number of allocated items increases. For instance, consider membership in a free gym. As the gym becomes more crowded, the utility each member derives from exercising there decreases. Membership in the gym corresponds to an item in our model. Cost of item i represents the marginal disutility of the members, which increases as the gym becomes more crowded.

Allocating items with increasing unit costs also arises in other familiar contexts, such as scheduling and disaster management. For example, consider multiple teams willing to be deployed in a disaster response. Each team has information (i.e., private value) about different regions of the affected area and can judge how much their region needs help. For teams to operate, they need communication frequencies for intra-team communication. The number of frequencies is limited and the more frequencies are allocated, the higher is the noise. The goal of a disaster response manager is to solicit truthful evaluations of situations in each team's region and to allocate frequencies to teams in the regions that need help the most. Additional frequencies should be allocated as long as the benefit derived from deploying an extra team outweighs the cost corresponding to the increase in noise on the communication channel. More generally, agents could be either emergency responders or sensor networks. The important part is that each agent is self-interested and maximizes its own utility, which is the case, for example, when agents are owned by different companies.

5.2 Setting

The setting of *allocation with costs* is defined by a triple $\langle n, c, v\rangle$, where n is the number of agents each desiring one unit of a homogenous good, c is the vector of marginal costs for producing each additional unit (item), and $v \in \mathbb{R}^n_+$ represents the agents' valuations for consuming the item. The marginal cost is increasing in the number of items, i.e. $c_1 \leq c_2 \leq \ldots \leq c_n$, and value profiles are such that $1 \geq v_1 \geq v_2 \geq \ldots \geq v_n \geq 0$. Monetary transfers are possible, and agents' utilities are quasi-linear.

In contrast to the case with free items, the number of allocated agents is *not* fixed but depends on c and v: we do not assign the item to an agent whose value for the item is lower than its cost. An efficient mechanism in this setting will maximize the total value of agents minus the total cost; the number of items allocated this way is $m(v, c) = \max_i(i \mid v_i \geq c_i)$ and the value of the efficient allocation is $\sum_{i \leq m(v,c)}(v_i - c_i)$.

Finally, we assume that at least one, but no more than $n-1$ items, are allocated: $c_1 < v_1$ and $c_n = 1$. It is easy to see that the efficient allocation in this setting is constant-dependent and defined by set $C = \{c_1, \ldots, c_{n-1}\}$. Hence, Theorem 3 implies.

5.3 Mechanisms

We now formulate the surplus-maximizing allocation and fair imposition problems in this domain. First, we modify the surplus ratio as follows:

$$S(c) = \min_{v \in V} \frac{\sum_{i=1}^{m(v,c)} v_i - m(v,c)\tau^a + \sum_{i=1}^n h(v_{-i})}{\sum_{i=1}^{m(v,c)}(v_i - c_i)}$$

where τ^a is the critical value of an allocated agent. Note that we fix the cost vector c and consider the worst ratio over all possible value profiles: we do not take the minimum over costs as that would obviously result in zero ratio—when the first $n-1$ costs are the same, the ratio is zero. The surplus-maximizing allocation problem is then defined by the following optimization program:

$$\max_{S \in \mathbb{R},\ r:\mathbb{R}^{n-1}\to\mathbb{R}} S \quad \text{s.t.} \quad \forall\, v \in V \tag{1}$$

$$m = \text{argmax}_i(v_i \geq c_i) \tag{2}$$

$$\tau^a = \max\{v_{m+1}, c_m\} \tag{3}$$

$$\sum_{i=1}^n h(v_{-i}) - m\tau^a \leq -\sum_{i \leq m} c_i \tag{4}$$

$$h(v_{-i}) \geq 0 \quad \forall i \tag{5}$$

$$\sum_{i \leq m}(v_i) - m\tau^a + \sum_{i=1}^n h(v_{-i}) \geq S \sum_{i \leq m}(v_i - c_i) \tag{6}$$

Here, (2) determines the number of items in an efficient allocation for the profile v, and corresponding critical values are defined by (3). The no-deficit property is enforced in (4): the payments collected from the agents must cover the costs of the allocated items. Constraint (5) guarantees that the utility of each agent is non-negative (recall from the previous section that enforcing the non-negativity of rebates is equivalent). Finally, as before, (6) ensures that the ratio is satisfied under all value profiles.

Similarly, we modify the fair imposition problem as follows: $\forall\, v \in V$,

$$m = \text{argmax}_i (v_i \geq c_i) \tag{7}$$
$$\tau^a = \max\{v_{m+1}, c_m\} \tag{8}$$
$$h(v_{-i}) \geq \frac{m v_k}{n} \quad \forall i \tag{9}$$
$$\sum_{i=1}^{n} h(v_{-i}) - m\tau^a \leq -\sum_{i \leq m} c_i \tag{10}$$

We have observed that the efficient allocation is constant-dependent in this model (as defined by the set of costs). Therefore, a *piecewise linear* surplus-maximizing and a k-fair mechanisms are obtained by solving (1)-(6) and (7)-(10), respectively, for the subset of profile values $\hat{V}$ as defined in 3, and linearly combining the rebate values in these—extreme—points on each of the regions of partition they define on space W.

6 Open Questions

Our work suggests several directions for future research. First, the characterization result in Theorem 2 can potentially be used to conclude the existence of linear optimal mechanisms in classes of problems, other than those with constant-dependent allocations: here, combinatorial auctions with single-minded bidders may be of particular interest; another extension is to public good settings. Second, a more general question in this context is about the necessity of conditions in Theorem 2. These conditions imply the existence of a partition of the space of agents' types, with certain properties: is it the case that if no such partition exists, an optimal linear mechanism does not exist either? Finally, an optimal partition may be complicated: in the setting of allocation with costs, the space is partitioned into $\binom{2n-2}{n-1}n!$ regions, where n is the number of agents. For small n, we empirically observed that most of the regions are required for an optimal mechanism, but it is likely that merging some of the regions does not decrease the solution quality too much. The tradeoff between efficiency and optimality remains open for further study.

References

[1] Martin J Bailey. The demand revealing process: To distribute the surplus. *Public Choice*, 91(2):107–26, April 1997.

[2] Ruggiero Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of vcg payments. In *AAMAS'06*, Hakodate, Japan, 2006.

[3] H. Crès, H. Moulin, and HEC Groupe. Commons with increasing marginal costs: random priority versus average cost. *Int. econ. review*, 44:1097–1115, 2003.

[4] Geoffroy de Clippel, Victor Naroditskiy, and Amy Greenwald. Destroy to save. In *EC'09*, 2009.

[5] S. De Vries, R.V. Vohra, Center for Mathematical Studies in Economics, and Management Science. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.

[6] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–601, 1973.

[7] Mingyu Guo and Vincent Conitzer. Worst-case optimal redistribution of vcg payments. In *EC'07*, pages 30–39, New York, NY, USA, 2007. ACM.

[8] G. Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, December 1968.

[9] R. Juarez. The worst absolute surplus loss in the problem of commons: random priority versus average cost. *Economic Theory*, 34(1):69–84, 2008.

[10] Herve Moulin. Almost budget-balanced vcg mechanisms to assign multiple objects. *Journal of Economic Theory*, 144(1):96–119, 2009.

[11] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140, New York, NY, USA, 1999. ACM.

[12] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[13] Ryan Porter, Yoav Shoham, and Moshe Tennenholtz. Fair imposition. *Journal of Economic Theory*, 118(2):209 – 228, 2004.

[14] Kevin Roberts. The characterization of implementable social choice rules. *In Jean-Jacques Laffont, editor, Aggregation and Revelation of Preferences*, 1979.

[15] Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *In Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP 03), Kinsale, County*. Springer, 2003.

[16] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, December 2008.

Victor Naroditskiy
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO17 1BJ, United Kingdom
Email: `vn@ecs.soton.ac.uk`

Maria Polukarov
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO17 1BJ, United Kingdom
Email: `mp3@ecs.soton.ac.uk`

Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO17 1BJ, United Kingdom
Email: `nrj@ecs.soton.ac.uk`

Strong Implementation of Social Choice Functions in Dominant Strategies

Sven O. Krumke and Clemens Thielen

Abstract

We consider the classical mechanism design problem of strongly implementing social choice functions in a setting where monetary transfers are allowed. In contrast to *weak* implementation, where only *one* equilibrium of a mechanism needs to yield the desired outcomes given by the social choice function, *strong* implementation (also known as *full* implementation) means that a mechanism is sought in which *all* equilibria yield the desired outcomes. For strong implementation, one cannot restrict attention to incentive compatible direct revelation mechanisms via the Revelation Principle, so the question whether a given social choice function is strongly implementable cannot be answered as easily as for weak implementation.

When considering Bayes Nash equilibria, the *Augmented Revelation Principle* states that it suffices to consider mechanisms in which the set of types of each agent is a subset of the set of her possible bids. Moreover, given some additional data, such a mechanism can be constructed by an iterative procedure via *selective elimination* of undesired equilibria in finitely (but possible exponentially) many steps. For dominant strategies as the equilibrium concept, however, no such results have been known so far. We close this gap by showing a variant of the Augmented Revelation Principle for dominant strategies and a selective elimination procedure for constructing the desired mechanisms in *polynomially many* steps. Using these results, we then show that strong implementability in dominant strategies can be decided in nondeterministic polynomial time. This complements the results obtained in the companion paper by Thielen and Westphal [7], where an efficient polynomial time algorithm for the problem is given when one restricts to strong implementation by incentive compatible direct revelation mechanisms.

1 Introduction

Mechanism design is a classical area of noncooperative game theory and microeconomics, which studies how privately known preferences of several people can be aggregated towards a social choice. Applications include the design of procedures for elections and for deciding upon public projects. Recently, the study of the Internet has fostered the interest in algorithmic aspects of mechanism design [5].

In the classical social choice setting considered in this paper, there are n selfish agents, which must make a collective decision from some finite set X of possible social choices. Each agent i has a private value $\theta_i \in \Theta_i$ (called the agent's *type*), which influences the preferences of all agents over the alternatives in X. Formally, this is modeled by a *valuation function* $V_i : X \times \Theta \to \mathbb{Q}$ for each agent i, where $\Theta = \Theta_1 \times \cdots \times \Theta_n$. Every agent i reports some information s_i from a set S_i of possible *bids* of i to the mechanism designer who must then choose an alternative from X based on these bids. The goal of the mechanism designer is to implement a given social choice function $f : \Theta \to X$, that is, to make sure that the alternative $f(\theta)$ is always chosen in equilibrium when the vector of true types is $\theta = (\theta_1, \ldots, \theta_n)$. To achieve this, the mechanism designer hands out a payment $P_i(\theta)$ to each agent i, which depends on the bids. Each agent then tries to maximize the sum of her valuation and payment by choosing an appropriate bid depending on her type. A *mechanism* $\Gamma = (S_1, \ldots, S_n, g, P)$ is defined by the sets $S_1, \ldots, S_n$ of possible bids of the agents, an *outcome function* $g : S_1 \times \cdots \times S_n \to X$, and the payment scheme $P = (P_1, \dots, P_n)$.

In the most common concept called *weak* implementation, the mechanism Γ is said to *implement* the social choice function f if *some* equilibrium of the noncooperative game defined by the mechanism yields the outcomes specified by f. An important result known as the *Revelation Principle* (cf. [2, p. 884]) states that a social choice function is weakly implementable if and only if it can be *truthfully implemented* by an incentive compatible *direct revelation mechanism*, which means that f can be implemented by a mechanism with $S_i = \Theta_i$ for all i and truthful reporting as an equilibrium that yields the outcome specified by f. As a result, the question whether there exists a mechanism that weakly implements a given social choice function f can be easily answered in time polynomial in $|\Theta|$ by checking for negative cycles in complete directed graphs on the agents' type spaces with changes of valuations as edge weights (cf. [1, 4, 6]).

The more robust concept of implementation called *strong implementation* (also known as *full implementation*) requires that not only one, but *all* equilibria of a mechanism yield the desired outcomes. Hence, a strong implementation does not rely on the implicit assumption that the agents always play the "desired" equilibrium if there is more than one. For strong implementation, the Revelation Principle does not hold, so one cannot, in general, restrict attention to direct revelation mechanisms and truthful implementations when trying to decide whether a social choice function is strongly implementable.

When considering Bayes Nash equilibria as the equilibrium concept, a generalization of the Revelation Principle called the *Augmented Revelation Principle* [3] states that it suffices to consider *augmented revelation mechanisms*, in which the set Θ_i of types of each agent i is a subset of the set S_i of her possible bids. Moreover, it was shown in [3] that one can always obtain an augmented revelation mechanism that strongly implements a strongly implementable social choice function f via the *selective elimination procedure* that starts with an incentive compatible direct revelation mechanism and some additional data on its equilibria and iteratively eliminates all the finitely many equilibria that do not yield the outcomes specified by f. To do so, one of the agents is given a new bid, so her set of possible bids is enlarged by one element. Since the procedure always stops after finitely many iterations, this also implies that the sets S_i can always be chosen to be finite.

For dominant strategies as the equilibrium concept, however, no such results have been known so far and it has not even been clear that one can restrict to finite sets of possible bids or polynomially sized payments. Hence, also the complexity of deciding whether a given social choice function f is strongly implementable in dominant strategies has remained open.

2 Our Contribution

We prove a variant of the Augmented Revelation Principle for dominant strategy equilibria. Our result implies that, as in the case of Bayes Nash equilibria, one can always restrict to augmented revelation mechanisms when trying to decide strong implementability of social choice functions in dominant strategies. Moreover, we present a selective elimination procedure for constructing augmented revelation mechanisms in finitely many steps when dominant strategies are considered. In contrast to the case of Bayes Nash equilibria, where the number of steps needed for selective elimination of all undesired equilibria of an incentive compatible direct revelation mechanism can be exponential, we show that our procedure for dominant strategies always terminates after *polynomially many* steps, which implies that only a polynomial number of possible bids for each agent is needed. Based on this result, we show that the payments in a strong implementation can always be chosen to be of polynomial encoding length and present a method for deciding strong implementability of a given social choice function in nondeterministic polynomial time. Doing so, we prove the first upper bound on the computational complexity of this classical mechanism design problem. We suspect that a matching lower bound can be proved as well, i.e., that deciding strong implementability of a social choice function in dominant strategies is NP-complete.

3 Problem Definition

We are given n *agents* identified with the set $N = \{1, \ldots, n\}$ and a finite set X of possible *social choices*. For each agent i, there is a finite set Θ_i of possible *types* and we write $\Theta = \Theta_1 \times \cdots \times \Theta_n$. The true type θ_i of agent i is known only to the agent herself. Each agent i has a *valuation function* $V_i : X \times \Theta \to \mathbb{Q}$, where $V_i(x, \theta)$ specifies the value that agent i assigns to alternative $x \in X$ when the types of the agents are $\theta \in \Theta$. A *social choice function* in this setting is a function $f : \Theta \to X$ that assigns an alternative $f(\theta) \in X$ to every vector θ of types.

Definition 1. *A* mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$ *consists of a set S_i of possible* bids *for each agent i, an* outcome function $g : S \to X$ *and a* payment scheme $P : S \to \mathbb{Q}^n$*, where $S := S_1 \times \cdots \times S_n$.*

A strategy *for agent i in the mechanism Γ is a function $\alpha_i : \Theta_i \to S_i$ that defines a bid $\alpha_i(\theta_i) \in S_i$ for every possible type θ_i of agent i. A* strategy profile *(in the mechanism Γ) is an n-tuple $\alpha = (\alpha_1, \ldots \alpha_n)$ containing a strategy α_i for each agent i.*

Definition 2. *Given a mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$, a vector $\theta \in \Theta$ of types of all agents, and a vector $s_{-i} \in S_{-i}$ of bids of all agents except i, the* utility *from a bid $s_i \in S_i$ for agent i is defined as*

$$U_i^\Gamma(s_{-i}, s_i|\theta) := V_i(g(s_{-i}, s_i), \theta) + P_i(s_{-i}, s_i).$$

A bid $\bar{s}_i \in S_i$ of an agent i is called a dominant bid *for type $\theta_i \in \Theta_i$ if it maximizes the utility of an agent i of type θ_i for every possible vector $s_{-i} \in S_{-i}$ of bids of the other agents and every possible vector $\theta_{-i} \in \Theta_{-i}$ of types of the other agents, i.e., if*

$$U_i^\Gamma(s_{-i}, \bar{s}_i|\theta) \geq U_i^\Gamma(s_{-i}, s_i|\theta) \quad \forall s_{-i} \in S_{-i}, \theta_{-i} \in \Theta_{-i}, s_i \in S_i.$$

A pair $(\theta, s) \in \Theta \times S$ of a type vector $\theta \in \Theta$ and bid vector $s \in S$ is called a dominant pair *if s_i is a dominant bid for θ_i for every $i \in N$. The strategy profile α is a* dominant strategy equilibrium *of Γ if $(\theta, \alpha(\theta))$ is a dominant pair for every $\theta \in \Theta$.*

Definition 3. *The mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$* strongly implements *the social choice function f if Γ has at least one equilibrium and* every *equilibrium α of Γ satisfies $g \circ \alpha = f$. The social choice function f is called* strongly implementable *if there exists a mechanism Γ that strongly implements f.*

Definition 4. *A mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$ is called a* direct revelation mechanism *if $S_i = \Theta_i$ for all $i \in N$. The direct revelation mechanism $(\Theta_1, \ldots, \Theta_n, f, P)$ defined by a social choice function f and a payment scheme P will be denoted by $\Gamma_{(f,P)}$. A direct revelation mechanism $\Gamma_{(f,P)}$ is called* incentive compatible *if truthful reporting is a dominant strategy equilibrium of $\Gamma_{(f,P)}$.*

Definition 5. *A mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$ is called* augmented revelation mechanism *if $S_i = \Theta_i \cup T_i$ for all $i \in N$ and arbitrary sets T_i.*

Definition 6 (Strong Implementability Problem)**.**

INSTANCE:	*The number n of agents, the set X of possible social choices, the sets Θ_i of possible types of the agents, the valuation functions $V_i : X \times \Theta \to \mathbb{Q}$, and the social choice function $f : \Theta \to X$.*
QUESTION:	*Is f strongly implementable in dominant strategies?*

To encode an instance of Strong Implementability, we need to do the following: For every valuation function $V_i : X \times \Theta \to \mathbb{Q}$, we need to store $|X| \cdot |\Theta|$ rational numbers. The social choice function $f : \Theta \to X$ has encoding length $|\Theta| \cdot \log(|X|)$. Thus, the encoding length of an instance of Strong Implementability is in $\Omega(|X| \cdot |\Theta| \cdot n)$.

4 The Augmented Revelation Principle for Dominant Strategies

In this section, we prove the Augmented Revelation Principle for dominant strategies and present our selective elimination procedure that, given an incentive compatible direct revelation mechanism $\Gamma_{(f,P)}$ and some data on it equilibria, constructs an augmented revelation mechanism that strongly implements f by an iterative procedure that stops after polynomially many steps.

Theorem 1 (Augmented Revelation Principle for dominant strategies)**.** *If a social choice function $f : \Theta \to X$ is strongly implementable in dominant strategies, then f can be strongly implemented in dominant strategies by an augmented revelation mechanism in which truthful reporting is an equilibrium.*

Proof. Given a mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$ that strongly implements f in dominant strategies, we construct an augmented revelation mechanism $\bar{\Gamma} = (\bar{S}_1, \ldots, \bar{S}_n, \bar{g}, \bar{P})$ that strongly implements f similar to the proof of the Augmented Revelation Principle for Bayes Nash equilibria given in [3]. Additionally, we have to define the new payment scheme $\bar{P}$ in terms of the given payment scheme P since the proof in [3] focused on the case without payments.

Given an arbitrary equilibrium $\alpha = (\alpha_1, \ldots, \alpha_n)$ of Γ, we define $\bar{S}_i := \Theta_i \cup T_i$, where

$$T_i := \{s_i \in S_i \mid s_i \notin \text{image}(\alpha_i)\},$$

and $\text{image}(\alpha_i) = \{\alpha_i(\theta_i) \mid \theta_i \in \Theta_i\}$ denotes the image of the function $\alpha_i : \Theta_i \to S_i$. We consider the functions $\phi_i : \bar{S}_i \to S_i$ given by

$$\phi_i(\bar{s}_i) := \begin{cases} \alpha_i(\theta_i) & \text{if } \bar{s}_i = \theta_i \text{ for } \theta_i \in \Theta_i \\ \bar{s}_i & \text{if } \bar{s}_i \in T_i \end{cases}$$

and define the outcome function $\bar{g} : \bar{S} \to X$ as $\bar{g} := g \circ \phi$, where $\phi = (\phi_1, \ldots, \phi_n)$. The payment scheme $\bar{P} : \bar{S} \to \mathbb{Q}$ is defined analogously as $\bar{P} := P \circ \phi$.

To show that $\bar{\Gamma}$ strongly implements f in dominant strategies, suppose that $\bar{\alpha} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_n)$ is an equilibrium of $\bar{\Gamma}$ and again consider the strategy profile $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ in Γ given by $\alpha_i^* := \phi_i \circ \bar{\alpha}_i$. As before, we then have $g \circ \alpha^* = g \circ \phi \circ \bar{\alpha} = \bar{g} \circ \bar{\alpha}$ and $P \circ \alpha^* = P \circ \phi \circ \bar{\alpha} = \bar{P} \circ \bar{\alpha}$ and claim that α^* is an equilibrium of Γ.

Since every $\phi_j : \bar{S}_j \to S_j$ is surjective, we can choose $\bar{s}_j \in \bar{S}_j$ with $\phi_j(\bar{s}_j) = s_j$ for each $j \in N$ and each $s_j \in S_j$. Then, for all $i \in N, \theta \in \Theta, s_{-i} \in S_{-i}$, and $s_i \in S_i$,

$$\begin{aligned} U_i^{\Gamma}(s_{-i}, \alpha_i^*(\theta_i)|\theta) &= V_i(g(s_{-i}, \alpha_i^*(\theta_i)), \theta) + P_i(s_{-i}, \alpha_i^*(\theta_i)) \\ &= V_i(\bar{g}(\bar{s}_{-i}, \bar{\alpha}_i(\theta_i)), \theta) + \bar{P}_i(\bar{s}_{-i}, \bar{\alpha}_i(\theta_i)) \\ &\geq V_i(\bar{g}(\bar{s}_{-i}, \bar{s}_i), \theta) + \bar{P}_i(\bar{s}_{-i}, \bar{s}_i) \\ &= V_i(g(s_{-i}, s_i), \theta) + P_i(s_{-i}, s_i) \\ &= U_i^{\Gamma}(s_{-i}, s_i|\theta), \end{aligned}$$

where the inequality follows since $\bar{\alpha}$ is an equilibrium of $\bar{\Gamma}$. Thus, α^* is an equilibrium of Γ as claimed. So since Γ strongly implements f, it follows that $f = g \circ \alpha^* = \bar{g} \circ \bar{\alpha}$, i.e., the equilibrium $\bar{\alpha}$ yields the outcomes specified by f. Hence, it just remains to show that truthful bidding is an equilibrium of $\bar{\Gamma}$. But this follows easily since, for every $\theta \in \Theta$, we have $\bar{g}(\theta) = (g \circ \phi)(\theta) = g(\alpha(\theta))$ and $\bar{P}(\theta) = (P \circ \phi)(\theta) = P(\alpha(\theta))$ and α is an equilibrium of Γ. $\square$

We now present our selective elimination procedure for dominant strategies. To this end, we need the following definition:

Definition 7. *A dominant bid $\bar{\theta}_i \in \Theta_i$ for type $\tilde{\theta}_i \in \Theta_i$ of agent $i \in N$ in a direct revelation mechanism $\Gamma_{(f,P)}$ can be* selectively eliminated *if there exists a nonempty subset $\bar{N} \subseteq N \setminus \{i\}$ of the other agents such that the following holds: For $\bar{S}_j := \Theta_j \cup \{\bar{s}_j\}$ for $j \in \bar{N}$, $\bar{S}_j := \Theta_j$ for $j \in N \setminus \bar{N}$, and $\bar{S} := \bar{S}_1 \times \cdots \times \bar{S}_n$, there exist functions $h : \bar{S} \to X$ and $\bar{P}_j : \bar{S} \to \mathbb{Q}, j \in N$, with $h_{|\Theta} = f$ and $(\bar{P}_j)_{|\Theta} = P_j$ such that:*

1. *For some $\tilde{\theta}_{-i} \in \Theta_{-i}$ and some bid vector $\bar{\theta}_{-(\bar{N}\cup\{i\})} \in \Theta_{-(\bar{N}\cup\{i\})}$ of the agents not in $\bar{N} \cup \{i\}$*
$$\begin{aligned} & V_i(h(\bar{s}_{\bar{N}}, \bar{\theta}_{-(\bar{N}\cup\{i\})}, \tilde{\theta}_i), \tilde{\theta}) + \bar{P}_i(\bar{s}_{\bar{N}}, \bar{\theta}_{-(\bar{N}\cup\{i\})}, \tilde{\theta}_i) \\ > \; & V_i(h(\bar{s}_{\bar{N}}, \bar{\theta}_{-(\bar{N}\cup\{i\})}, \bar{\theta}_i), \tilde{\theta}) + \bar{P}_i(\bar{s}_{\bar{N}}, \bar{\theta}_{-(\bar{N}\cup\{i\})}, \bar{\theta}_i). \end{aligned}$$

2. *For all $j \in N, \theta \in \Theta, s \in \bar{S} \setminus \Theta$*
$$V_j(h(s_{-j}, \theta_j), \theta) + \bar{P}_j(s_{-j}, \theta_j) \geq V_j(h(s_{-j}, s_j), \theta) + \bar{P}_j(s_{-j}, s_j).$$

A dominant pair $(\theta, \theta') \in \Theta^2$ can be selectively eliminated if the dominant bid $\theta'_i \in \Theta_i$ for type $\theta_i \in \Theta_i$ can be selectively eliminated for some $i \in N$.

Here, each agent $j \in \bar{N}$ is given a new bid $\bar{s}_j$. The function h extends f to the enlarged set $\bar{S}$ of possible bids by specifying the outcomes chosen when at least one agent chooses a non-type message. Similarly, the functions $\bar{P}_j$ extend the payment functions P_j to $\bar{S}$. The first condition says that, for some type vector $\tilde{\theta}_{-i} \in \Theta_{-i}$ of the other agents and some bid vector $\bar{\theta}_{-(\bar{N}\cup\{i\})}$ of the agents not in $\bar{N} \cup \{i\}$, agent i can increase her utility by bidding her true type $\tilde{\theta}_i$ instead of $\bar{\theta}_i$ in the case that the agents in $\bar{N}$ choose their new non-type messages. Thus, $\bar{\theta}_i$ is not a dominant bid for type $\tilde{\theta}_i$ anymore. On the other hand, the second condition ensures that all pairs $(\theta, \theta) \in \Theta^2$ stay dominant pairs, so truthful reporting is preserved as an equilibrium.

Definition 8. *A dominant pair $(\theta, \theta') \in \Theta^2$ in the direct revelation mechanism $\Gamma_{(f,P)}$ is called* bad *if $f(\theta) \neq f(\theta')$. $\Gamma_{(f,P)}$ satisfies the* selective elimination condition *if every bad dominant pair can be selectively eliminated.*

The idea behind Definition 8 is the following observation, which follows immediately from the definitions:

Observation 1. *A direct revelation mechanism $\Gamma_{(f,P)}$ with at least one dominant strategy equilibrium has a* bad *dominant strategy equilibrium if and only if there exists a bad dominant pair $(\theta, \theta') \in \Theta^2$ in $\Gamma_{(f,P)}$. In particular, an* incentive compatible *direct revelation mechanism $\Gamma_{(f,P)}$ has a bad dominant strategy equilibrium if and only if there exists a bad dominant pair $(\theta, \theta') \in \Theta^2$ in $\Gamma_{(f,P)}$.*

Hence, selectively eliminating all bad dominant pairs will lead to elimination of all bad equilibria. The difference to the case of Bayes Nash equilibria discussed in [3] is that one does not need to consider complete equilibria α and check whether they can be selectively eliminated. Here, one has to consider only bad dominant pairs $(\theta, \theta') \in \Theta^2$. While there are potentially exponentially many bad equilibria, the number of bad dominant pairs $(\theta, \theta') \in \Theta^2$ is bounded by $|\Theta|^2$, which is *polynomial* in the encoding length of the input. This observation will play a crucial role when we show that Strong Implementability is in NP when considering dominant strategies.

Theorem 2. *Suppose that the social choice function $f : \Theta \to X$ is strongly implementable in dominant strategies. Then there exists an incentive compatible direct revelation mechanism $\Gamma_{(f,P)}$ that satisfies the selective elimination condition.*

Proof. Theorem 1 states that there exists an augmented revelation mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$ that strongly implements f in dominant strategies and in which truthful reporting is an equilibrium. In particular, this implies that $g_{|\Theta} = f$, and we claim that the direct revelation mechanism $\Gamma_{(f,P_{|\Theta})}$ is as required.

Incentive compatibility of $\Gamma_{(f,P_{|\Theta})}$ follows directly from the fact that truthful reporting is an equilibrium in Γ. To show that $\Gamma_{(f,P_{|\Theta})}$ satisfies the selective elimination condition, consider a bad dominant pair $(\theta, \theta') \in \Theta^2$ in $\Gamma_{(f,P_{|\Theta})}$ (if none exists, we are done). Since Γ strongly implements f, it can have no bad equilibria and, thus, (θ, θ') cannot be a dominant pair in Γ. Hence, there must be an agent i, a vector $\tilde{\theta}_{-i} \in \Theta_{-i}$ of types of the other agents, and a vector $\bar{s} \in S$ of bids such that

$$U_i^\Gamma(\bar{s}_{-i}, \bar{s}_i | \theta_i, \tilde{\theta}_{-i}) > U_i^\Gamma(\bar{s}_{-i}, \theta'_i | \theta_i, \tilde{\theta}_{-i}).$$

Moreover, since truthful reporting is an equilibrium in Γ, we know that

$$U_i^\Gamma(\bar{s}_{-i}, \theta_i | \theta_i, \tilde{\theta}_{-i}) \geq U_i^\Gamma(\bar{s}_{-i}, \bar{s}_i | \theta_i, \tilde{\theta}_{-i}),$$

so we obtain

$$U_i^\Gamma(\bar{s}_{-i}, \theta_i | \theta_i, \tilde{\theta}_{-i}) > U_i^\Gamma(\bar{s}_{-i}, \theta'_i | \theta_i, \tilde{\theta}_{-i}). \tag{1}$$

Moreover, since (θ, θ') is a dominant pair in $\Gamma_{(f,P_{|\Theta})}$, we know that $\bar{s}_j \notin \Theta_j$ for at least one $j \neq i$. Hence, the set $\bar{N} := \{j \neq i : \bar{s}_j \notin \Theta_j\}$ is nonempty. We now set $\bar{S}_j := \Theta_j \cup \{\bar{s}_j\}$ for $j \in \bar{N}$, $\bar{S}_j := \Theta_j$ for $j \in N \setminus \bar{N}$, and $\bar{S} := \bar{S}_1 \times \cdots \times \bar{S}_n$ as in the definition of selective elimination. The function $h : \bar{S} \to X$ is defined as the restriction of g to $\bar{S} \subseteq S$ and it satisfies $h_{|\Theta} = g_{|\Theta} = f$. Analogously, the functions $\bar{P}_j : \bar{S} \to \mathbb{Q}$ are defined as the restrictions of the P_j to $\bar{S} \subseteq S$ and we have $(\bar{P}_j)_{|\Theta} = (P_j)_{|\Theta}$. Defining $\bar{\theta}_j := \bar{s}_j \in \Theta_j$ for all $j \notin (\bar{N} \cup \{i\})$, it is now immediate that the dominant bid θ'_i for type θ_i of agent i can be selectively eliminated, i.e., that Conditions 1 and 2 in the definition of selective elimination are satisfied: Condition 1 follows directly from (1) and the definitions, and Condition 2 follows since truthful reporting is an equilibrium in Γ. Thus, the bad dominant pair $(\theta, \theta') \in \Theta^2$ can be selectively eliminated. ☐

We are now ready to present our selective elimination procedure for constructing augmented revelation mechanisms. This procedure is used to prove the following theorem, which is states that the selective elimination condition is also sufficient for strong implementability:

Theorem 3. *Suppose that there exists an incentive compatible direct revelation mechanism $\Gamma_{(f,P)}$ that satisfies the selective elimination condition. Then the social choice function $f : \Theta \to X$ is strongly implementable in dominant strategies.*

Proof. We start with the direct revelation mechanism $\Gamma_{(f,P)}$ and proceed inductively to selectively eliminate all bad dominant pairs $(\theta, \theta') \in \Theta^2$ in $\Gamma_{(f,P)}$ one by one without introducing any new dominant pairs by augmenting the mechanism appropriately. Since there can only be finitely many bad dominant pairs, the procedure stops after a finite number of steps with an augmented revelation mechanism without bad dominant pairs and, thus, without bad equilibria. In fact, the procedure stops after a *polynomial* number of steps since there can only be $|\Theta|^2$ many bad dominant pairs in $\Gamma_{(f,P)}$.

We describe a representative stage of this iterative procedure. From the previous iteration, we are given an augmentation $\Gamma = (S_1, \ldots, S_n, g, P')$ of $\Gamma_{(f,P)}$ with $S_i = \Theta_i \cup T_i$ for all $i \in N$, and $\theta' \in \Theta$ for every dominant pair (θ, θ') of Γ. Let (θ, θ') be a bad dominant pair of Γ. Let $i \in N$ be such that the dominant bid θ'_i for type θ_i of agent i can be selectively eliminated, and suppose that $\emptyset \neq \bar{N} \subseteq N \setminus \{i\}$, $\bar{S} = \bar{S}_1 \times \cdots \times \bar{S}_n$ with $\bar{S}_j = \Theta_j \cup \{\bar{s}_j\}$ for $j \in \bar{N}$ and $\bar{S}_j = \Theta_j$ for $j \in N \setminus \bar{N}$, $h : \bar{S} \to X$, and $\bar{P}_j : \bar{S} \to \mathbb{Q}$ are as in the definition of selective elimination. Consider the mechanism $\tilde{\Gamma} = (\tilde{S}_1, \ldots, \tilde{S}_n, \tilde{g}, \tilde{P})$ with

$$\begin{aligned} \tilde{S}_j &:= S_j \cup \{\bar{s}_j\} \text{ for } j \in \bar{N}, \\ \tilde{S}_j &:= S_j \text{ for } j \in N \setminus (\bar{N} \cup \{i\}), \\ \tilde{S}_i &:= S_i \cup \{\text{CFL}\}. \end{aligned}$$

Hence, each agent $j \in \bar{N}$ is given a new bid $\bar{s}_j$ (a *flag*), and agent i is given a new *counterflag* CFL. We set $\tilde{g}_{|S} := g$ and $\tilde{P}_{|S} := P'$, i.e., outcomes and payments associated with bids from the previous stages are left unchanged. Outcomes and payments associated with the new bids are defined as follows:

1. If the bid vector is in $\bar{S}$, the outcome and the payments are given by h and the $\bar{P}_j$, respectively, i.e., $\tilde{g}(s) := h(s)$ for $s \in \bar{S}$ and $\tilde{P}_j(s) := \bar{P}_j(s)$ for $s \in \bar{S}, j \in N$. Note that this definition agrees with the outcomes and payments of the previous stages when the bid vector is in Θ since $\tilde{g}_{|\Theta} = f$ and $(\tilde{P}_j)_{|\Theta} = P$.

2. If some agents $\emptyset \neq \tilde{N} \subseteq \bar{N}$ choose their new bids, agent i does *not* choose her new counterflag CFL, but some agents $j \in \tilde{\tilde{N}} \subseteq N \setminus \tilde{N}$ choose bids in $T_j = S_j \setminus \Theta_j$, then outcome and payments are as if each agent $k \in \tilde{N}$ had chosen a fixed type $\theta^0_k \in \Theta_k$, but each agent $k \in \tilde{N}$ is charged $\epsilon > 0$ for choosing her new bid $\bar{s}_k$ if agent i chooses a bid in $T_i = S_i \setminus \Theta_i$.

3. If *at least two* agents $\tilde{N} \subseteq \bar{N}, |\tilde{N}| \geq 2$, choose their new bids and agent i chooses CFL, then outcome and payments are as if agent i had reported a fixed type $\theta^0_i \in \Theta_i$.

4. If *no* agent in $\bar{N}$ chooses her new bid, but agent i chooses CFL, then outcome and payments are as if agent i had reported $\theta^0_i \in \Theta_i$, but agent i is charged $\epsilon > 0$ for choosing CFL.

5. If *exactly one* agent $k \in \bar{N}$ chooses her new bid and agent i chooses CFL, then outcome and payments are as if agent i had reported $\theta^0_i \in \Theta_i$ and agent k had reported $\theta^0_k \in \Theta_k$, but agent k is charged $\epsilon > 0$.

We now have to show that truthful reporting is still an equilibrium in $\tilde{\Gamma}$, (θ, θ') is not a dominant pair anymore, and there are no new dominant bid pairs in $\tilde{\Gamma}$ (or, equivalently, no new dominant bids). Note that, since the outcomes and payments associated with bids from the previous stages are left unchanged, any new dominant bid of an agent would have to be one of the agent's new bids.

Claim 1. *Truthful reporting is an equilibrium in* $\tilde{\Gamma}$.

Proof. We consider a fixed agent $j \in N$ and show that truthful reporting is a dominant strategy for j in $\tilde{\Gamma}$.

As long as the other agents bid a vector in $S_{-j} \cup \bar{S}_{-j}$, truthful reporting is always optimal for agent j among all bids in $S_j \cup \bar{S}_j$ by Condition 2 in the definition of selective elimination and since truthful reporting is an equilibrium in Γ. When agent j bids a new bid (if she has one), this can only lead to some agents being charged ϵ as long as the other agents still bid a vector in $S_{-j} \cup \bar{S}_{-j}$. Hence, truthful reporting is always optimal among *all* bids of agent j in this case.

If the vector of bids of the other agents is *not* in $S_{-j} \cup \bar{S}_{-j}$, some agents choose new bids and some agents choose previously added bids. Hence, Case 2 in the definition of $\tilde{\Gamma}$ applies when agent i does not choose CFL, and Case 3,4, or 5 applies when agent i chooses CFL. But outcomes and payments in each of these cases are equivalent to the outcome and payments resulting from some bid vector in $S \cup \bar{S}$, except that some agents are possibly charged ϵ. Hence, truthful reporting is optimal for agent j by the case considered above. $\square$

Claim 2. *(θ, θ') is not a dominant pair in $\tilde{\Gamma}$.*

Proof. Follows immediately from Case 1 in the definition of $\tilde{\Gamma}$ and the definition of selective elimination. $\square$

Claim 3. *There is no dominant bid $\bar{s}_k$ for any type $\theta_k \in \Theta_k$ of any agent $k \in \bar{N}$ in $\tilde{\Gamma}$.*

Proof. If agent k has type θ_k and chooses the bid $\bar{s}_k$, consider the situation in which no other agent $j \in \bar{N} \setminus \{k\}$ chooses her new bid and agent i chooses CFL. Then, by Cases 4 and 5 in the definition of $\tilde{\Gamma}$, agent k could increase her utility by $\epsilon > 0$ by bidding θ_k^0 instead of $\bar{s}_k$ for every possible vector θ_{-k} of types of the other agents. $\square$

Claim 4. *CFL is not a dominant bid for any type $\theta_i \in \Theta_i$ of agent i in $\tilde{\Gamma}$.*

Proof. If agent i has type θ_i and chooses the bid CFL, consider the situation in which no agent $j \in \bar{N}$ chooses her new bid. Then, by Case 4 in the definition of $\tilde{\Gamma}$, agent i could increase her utility by $\epsilon > 0$ by bidding θ_i^0 instead of CFL for every possible vector θ_{-i} of types of the other agents. $\square$

By inductive application of the claims, the final mechanism obtained after eliminating all bad dominant pairs (θ, θ') in $\Gamma_{(f,P)}$ has no bad dominant pairs, but truthful reporting is still an equilibrium. Hence, this mechanism strongly implements f in dominant strategies, which proves the theorem. $\square$

Theorem 4. *The social choice function $f : \Theta \to X$ is strongly implementable in dominant strategies if and only if there exists an incentive compatible direct revelation mechanism $\Gamma_{(f,P)}$ that satisfies the selective elimination condition.* $\square$

Theorem 4 is the main ingredient needed for the proof of our complexity result on Strong Implementability in the next section. The following lemma resolves one last formal problem resulting from the definition of selective elimination: Giving all agents in $\bar{N}$ a new bid could yield an exponentially large space $\bar{S}$ of possible bids in the definition of selective elimination. This can, however, only happen if some of the agents have only one possible type, and the behavior of such agents cannot impose restrictions on the implementability of a social choice function since the types of these agents are common knowledge.

Lemma 1. *Let $Z \subseteq N$ denote the set of agents whose type space consists of only one element, i.e., $|\Theta_j| = 1$ for every $j \in Z$. Consider the instance of Strong Implementability for the agents in $N \setminus Z$ given by the valuations $V_i^{-Z} : X \times \Theta_{-Z} \to \mathbb{Q}$ defined by $V_i^{-Z}(x, \theta_{-Z}) := V_i(x, \theta_{-Z}, \theta_Z)$ and the social choice function $f^{-Z} : \Theta_{-Z} \to X$ defined by $f^{-Z}(\theta_{-Z}) := f(\theta_{-Z}, \theta_Z)$, where θ_Z is the unique type vector of the agents in Z. Then f is* *strongly implementable in dominant strategies if and only if f^{-Z} is strongly implementable in dominant strategies.*

Proof. Suppose that f is strongly implementable in dominant strategies. Then, by Theorem 1, there exists an augmented revelation mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$ that strongly implements f in dominant strategies and in which truthful reporting is an equilibrium. Without loss of generality, we assume that $N \setminus Z = \{1, \ldots, z\}$ with $z := |N \setminus Z|$ and consider the mechanism $\Gamma^{-Z} = (S_1^{-Z}, \ldots, S_z^{-Z}, g^{-Z}, P^{-Z})$ defined by

$$\begin{aligned} S_i^{-Z} &:= S_i \text{ for } i = 1, \ldots, z \\ g^{-Z}(s_1, \ldots, s_z) &:= g(s_1, \ldots, s_z, \theta_Z) \\ P_i^{-Z}(s_1, \ldots, s_z) &:= P_i(s_1, \ldots, s_z, \theta_Z). \end{aligned}$$

Then $\alpha^{-Z} = (\alpha_1^{-Z}, \ldots \alpha_z^{-Z}) \longmapsto (\alpha_1^{-Z}, \ldots, \alpha_z^{-Z}, id_{\Theta_Z}) = \alpha$ defines an injective map from the set of strategy profiles in Γ^{-Z} to the set of strategy profiles in Γ, and α^{-Z} is an equilibrium in Γ^{-Z} if and only if α is an equilibrium in Γ (here, we use that truthful bidding is a dominant strategy in Γ for each agent in Z). Moreover, we have $g^{-Z} \circ \alpha^{-Z} = g^{-Z}$ if and only if $g \circ \alpha = g$. Hence, since Γ strongly implements f, it follows that Γ^{-Z} strongly implements f^{-Z}.

Conversely, assume that f^{-Z} is strongly implementable and denote an augmented revelation mechanism that strongly implements it in dominant strategies and in which truthful reporting is an equilibrium by $\Gamma^{-Z} = (S_1^{-Z}, \ldots, S_z^{-Z}, g^{-Z}, P^{-Z})$. We define a mechanism $\Gamma = (S_1, \ldots, S_n, g, P)$ for all agents as follows:

$$\begin{aligned} S_i &:= S_i^{-Z} \text{ for } i = 1, \ldots, z \\ S_i &:= \Theta_i = \{\theta_i\} \text{ for } i \in Z \\ g(s_1, \ldots, s_z, \theta_Z) &:= g^{-Z}(s_1, \ldots, s_z) \\ P_i(s_1, \ldots, s_z, \theta_Z) &:= P_i^{-Z}(s_1, \ldots, s_z) \text{ for } i = 1, \ldots, z \\ P_i(s_1, \ldots, s_z, \theta_Z) &:= 0 \text{ for } i \in Z \end{aligned}$$

Then $\alpha = (\alpha_1, \ldots \alpha_z, id_{\Theta_Z}) \longmapsto (\alpha_1, \ldots, \alpha_z) = \alpha^{-Z}$ defines a bijection between the set of strategy profiles in Γ and the set of strategy profiles in Γ^{-Z}, and α is an equilibrium in Γ if and only if α^{-Z} is an equilibrium in Γ^{-Z}. Again, we have $g^{-Z} \circ \alpha^{-Z} = g^{-Z}$ if and only if $g \circ \alpha = g$. Hence, since Γ^{-Z} strongly implements f^{-Z}, Γ strongly implements f. $\square$

Lemma 1 shows that, when trying to decide strong implementability of a social choice function f in dominant strategies, one can disregard all agents that have only one possible type by considering the equivalent problem of strong implementability of the social choice function f^{-Z}. Hence, we may from now on assume that $|\Theta_j| \geq 2$ for every agent $j \in N$. With this assumption, the cardinality of the set $\bar{S}$ in the definition of selective elimination is only quadratic in $|\Theta|$:

$$|\bar{S}| = \prod_{j \in \bar{N}} \underbrace{(|\Theta_j| + 1)}_{\leq 2|\Theta_j|} \cdot \prod_{j \in N \setminus \bar{N}} |\Theta_j| \leq \underbrace{2^{|N|-1}}_{\leq \prod_{j=1}^{n} |\Theta_j|} \prod_{j=1}^{n} |\Theta_j| \leq (\prod_{j=1}^{n} |\Theta_j|)^2 = |\Theta|^2.$$

5 Solving Strong Implementability in Nondeterministic Polynomial Time

In this section, we use our results on augmented revelation mechanisms and selective elimination to show that Strong Implementability can be decided in nondeterministic polynomial time when dominant strategies are considered.

Suppose we are given a yes-instance of Strong Implementability, i.e., an instance with a strongly implementable social choice function f. Theorem 4 then tells us that there exists an incentive compatible direct revelation mechanism $\Gamma_{(f,P)}$ satisfying the selective elimination condition. We denote the set of all bad dominant pairs (θ, θ') by $D \subseteq \Theta^2$. Similarly, for each $i \in N$, we denote the set of all pairs $(\tilde{\theta}_i, \bar{\theta}_i) \in \Theta_i^2$ such that $\bar{\theta}_i$ is a dominant bid for type $\tilde{\theta}_i$ of agent i by $D_i \subseteq \Theta_i^2$.

Since $\Gamma_{(f,P)}$ satisfies the selective elimination condition, we know that each bad dominant pair $(\theta, \theta') \in D$ can be selectively eliminated. Suppose that, for every $(\theta, \theta') \in D$, $\left(i^{(\theta,\theta')}, \bar{N}^{(\theta,\theta')}, h^{(\theta,\theta')}, \tilde{\theta}^{(\theta,\theta')}_{-i^{(\theta,\theta')}}, \bar{\theta}^{(\theta,\theta')}_{-(\bar{N}\cup\{i^{(\theta,\theta')}\})}\right)$ is the data which, together with appropriate payment functions $\bar{P}_j^{(\theta,\theta')}$ for $j \in N$, can be used to selectively eliminate the bad dominant pair (θ, θ').

Similarly, suppose that, for every $i \in N$ and every pair $(\tilde{\theta}_i, \bar{\theta}_i) \in \Theta_i^2 \setminus D_i$ (i.e., for every pair $(\tilde{\theta}_i, \bar{\theta}_i)$ of types of agent i such that $\bar{\theta}_i$ is *not* a dominant bid for type $\tilde{\theta}_i$), $\left(\tilde{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \bar{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}\right)$ is a pair of a type vector and a bid vector of the other agents such that

$$U_i^{\Gamma_{(f,P)}}\left(\bar{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \tilde{\theta}_i | \tilde{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \tilde{\theta}_i\right) > U_i^{\Gamma_{(f,P)}}\left(\bar{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \bar{\theta}_i | \tilde{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \tilde{\theta}_i\right).$$

The possible payment functions $P_j : \Theta \to \mathbb{Q}$ of the mechanism $\Gamma_{(f,P)}$ and the functions $\bar{P}_j^{(\theta,\theta')} : \bar{S}^{(\theta,\theta')} \to \mathbb{Q}$ are then given by the solutions of the system of linear inequalities in the variables $P_j(\theta)$ for $j \in N, \theta \in \Theta$ and $\bar{P}_j^{(\theta,\theta')}(s)$ for $j \in N, (\theta,\theta') \in D$, $s \in \bar{S}^{(\theta,\theta')} \setminus \Theta$ displayed on Page 11. Note that the values $\bar{P}_j^{(\theta,\theta')}(s)$ for $s \in \Theta$ do not need to appear in the system since we require that $(\bar{P}_j^{(\theta,\theta')})_{|\Theta} = P_j$ for all $j \in N, (\theta,\theta') \in D$.

Inequalities (2) and (3) encode exactly which bids $\bar{\theta}_i \in \Theta_i$ are dominant bids for any type $\tilde{\theta}_i$ of an agent i in $\Gamma_{(f,P)}$ (in particular, (3) encodes incentive compatibility of $\Gamma_{(f,P)}$) and (4) corresponds to Condition 1 in the definition of selective elimination. Inequalities (5) and (6) correspond to Condition 2, where (6) is stated separately since it involves the variable $P_j(s_{-j}, \theta_j)$ instead of $\bar{P}_j^{(\theta,\theta')}(s_{-j}, \theta_j)$ as in (5).

Note that there are only *polynomially many* variables and inequalities in this system and all coefficients have polynomial encoding length. Hence, we can find a relative interior point of the polyhedron defined by the system, which corresponds to a solution of the original system with strict inequalities in (2) and (4), in polynomial time (e.g., by using the ellipsoid method). In particular, this shows that all the values $P_j(\theta)$ and $\bar{P}_j^{(\theta,\theta')}(s)$ can be chosen to have polynomial encoding length, which proves the following Theorem:

Theorem 5. *The social choice function $f : \Theta \to X$ is strongly implementable in dominant strategies if and only if there exists an incentive compatible direct revelation mechanism $\Gamma_{(f,P)}$* of polynomial encoding length *that satisfies the selective elimination condition. In this case, for every (fixed) bad dominant pair (θ, θ') of $\Gamma_{(f,P)}$, the data $(i, \bar{N}, h, \tilde{\theta}_{-i}, \bar{\theta}_{-(\bar{N}\cup\{i\})})$ needed to selectively eliminate (θ, θ') can be chosen to have polynomial encoding length.* □

Using Theorem 5, we can now state our nondeterministic polynomial time algorithm for Strong Implementability and, thus, prove the main result of this section:

Theorem 6. *Strong Implementability $\in$ NP.*

Proof. Assume that the given social choice function f is strongly implementable in dominant strategies. Then, by Theorem 5, there exists an incentive compatible direct revelation mechanism $\Gamma_{(f,P)}$ of polynomial encoding length that satisfies the selective elimination condition. Moreover, for every bad dominant pair (θ, θ') of $\Gamma_{(f,P)}$, the

For all $i \in N, (\tilde{\theta}_i, \bar{\theta}_i) \in \Theta_i^2 \setminus D_i$:

$$\begin{aligned} V_i\left(f(\bar{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \tilde{\theta}_i), \tilde{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \tilde{\theta}_i\right) + P_i\left(\bar{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \tilde{\theta}_i\right) & \\ -V_i\left(f(\bar{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \bar{\theta}_i), \tilde{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \tilde{\theta}_i\right) - P_i\left(\bar{\theta}_{-i}^{(\tilde{\theta}_i,\bar{\theta}_i)}, \bar{\theta}_i\right) & > \quad 0 \end{aligned} \tag{2}$$

For all $i \in N, (\tilde{\theta}_i, \bar{\theta}_i) \in D_i$ and all $\tilde{\theta}_{-i}, \bar{\theta}_{-i} \in \Theta_{-i}, \theta_i \in \Theta_i$:

$$\begin{aligned} V_i\left(f(\bar{\theta}_{-i}, \bar{\theta}_i), \tilde{\theta}\right) + P_i\left(\bar{\theta}_{-i}, \bar{\theta}_i\right) & \\ -V_i\left(f(\bar{\theta}_{-i}, \theta_i), \tilde{\theta}\right) - P_i\left(\bar{\theta}_{-i}, \theta_i\right) & \geq \quad 0 \end{aligned} \tag{3}$$

For all $(\theta, \theta') \in D$:

$$\begin{aligned} V_{i^{(\theta,\theta')}}\left(h^{(\theta,\theta')}(\bar{s}_{\bar{N}^{(\theta,\theta')}}, \bar{\theta}^{(\theta,\theta')}_{-(\bar{N}^{(\theta,\theta')}\cup\{i^{(\theta,\theta')}\})}, \theta_{i^{(\theta,\theta')}}), \tilde{\theta}^{(\theta,\theta')}_{-i^{(\theta,\theta')}}, \theta_{i^{(\theta,\theta')}}\right) + \bar{P}^{(\theta,\theta')}_{i^{(\theta,\theta')}}\left(\bar{s}_{\bar{N}^{(\theta,\theta')}}, \bar{\theta}^{(\theta,\theta')}_{-(\bar{N}\cup\{i^{(\theta,\theta')}\})}, \theta_{i^{(\theta,\theta')}}\right) & \\ -V_{i^{(\theta,\theta')}}\left(h^{(\theta,\theta')}(\bar{s}_{\bar{N}^{(\theta,\theta')}}, \bar{\theta}^{(\theta,\theta')}_{-(\bar{N}^{(\theta,\theta')}\cup\{i^{(\theta,\theta')}\})}, \theta'_{i^{(\theta,\theta')}}), \tilde{\theta}^{(\theta,\theta')}_{-i^{(\theta,\theta')}}, \theta_{i^{(\theta,\theta')}}\right) - \bar{P}^{(\theta,\theta')}_{i^{(\theta,\theta')}}\left(\bar{s}_{\bar{N}^{(\theta,\theta')}}, \bar{\theta}^{(\theta,\theta')}_{-(\bar{N}\cup\{i^{(\theta,\theta')}\})}, \theta'_{i^{(\theta,\theta')}}\right) > & \quad 0 \end{aligned} \tag{4}$$

For all $j \in N, (\theta, \theta') \in D$ and all $\theta \in \Theta, s_{-j} \in \bar{S}_{-j}^{(\theta,\theta')} \setminus \Theta_{-j}, s_j \in \bar{S}_j^{(\theta,\theta')}$:

$$\begin{aligned} V_j\left(h^{(\theta,\theta')}(s_{-j}, \theta_j), \theta\right) + \bar{P}_j^{(\theta,\theta')}\left(s_{-j}, \theta_j\right) & \\ -V_j\left(h^{(\theta,\theta')}(s_{-j}, s_j), \theta\right) - \bar{P}_j^{(\theta,\theta')}\left(s_{-j}, s_j\right) & \geq \quad 0 \end{aligned} \tag{5}$$

For all $j \in N, (\theta, \theta') \in D$ and all $\theta \in \Theta, s_{-j} \in \Theta_{-j}, s_j \in \bar{S}_j^{(\theta,\theta')} \setminus \Theta_j$:

$$\begin{aligned} V_j\left(h^{(\theta,\theta')}(s_{-j}, \theta_j), \theta\right) + P_j\left(s_{-j}, \theta_j\right) & \\ -V_j\left(h^{(\theta,\theta')}(s_{-j}, s_j), \theta\right) - \bar{P}_j^{(\theta,\theta')}\left(s_{-j}, s_j\right) & \geq \quad 0 \end{aligned} \tag{6}$$

data $\left(i^{(\theta,\theta')}, \bar{N}^{(\theta,\theta')}, h^{(\theta,\theta')}, \tilde{\theta}^{(\theta,\theta')}_{-i^{(\theta,\theta')}}, \bar{\theta}^{(\theta,\theta')}_{-(\bar{N}\cup\{i^{(\theta,\theta')}\})}\right)$ needed to selectively eliminate (θ,θ') can be chosen to have polynomial encoding length. Now consider the following nondeterministic algorithm for verifying that f is strongly implementable:

Algorithm 1.

1. *Guess the (polynomially many) values $P_j(\theta)$.*
2. *For every $i \in N$, guess the set D_i of all pairs $(\tilde{\theta}_i, \bar{\theta}_i)$ of types $\tilde{\theta}_i \in \Theta_i$ and dominant bids $\bar{\theta}_i \in \Theta_i$ for type $\tilde{\theta}_i$ in $\Gamma_{(f,P)}$.*
3. *Guess the set $D \subseteq \Theta^2$ of all bad dominant pairs in $\Gamma_{(f,P)}$.*
4. *For every $i \in N$ and every pair $(\tilde{\theta}_i, \bar{\theta}_i) \in \Theta_i^2 \setminus D_i$, guess the pair $(\tilde{\theta}^{(\tilde{\theta}_i,\bar{\theta}_i)}_{-i}, \bar{\theta}^{(\tilde{\theta}_i,\bar{\theta}_i)}_{-i})$ of a type vector and a bid vector of the other agents such that*
$$U_i^{\Gamma_{(f,P)}}\left(\bar{\theta}^{(\tilde{\theta}_i,\bar{\theta}_i)}_{-i}, \tilde{\theta}_i | \tilde{\theta}^{(\tilde{\theta}_i,\bar{\theta}_i)}_{-i}, \tilde{\theta}_i\right) > U_i^{\Gamma_{(f,P)}}\left(\bar{\theta}^{(\tilde{\theta}_i,\bar{\theta}_i)}_{-i}, \bar{\theta}_i | \tilde{\theta}^{(\tilde{\theta}_i,\bar{\theta}_i)}_{-i}, \tilde{\theta}_i\right).$$
5. *For every $(\theta,\theta') \in D$, guess the data $\left(i^{(\theta,\theta')}, \bar{N}^{(\theta,\theta')}, h^{(\theta,\theta')}, \tilde{\theta}^{(\theta,\theta')}_{-i^{(\theta,\theta')}}, \bar{\theta}^{(\theta,\theta')}_{-(\bar{N}\cup\{i^{(\theta,\theta')}\})}\right)$ needed to selectively eliminate the bad dominant pair (θ,θ').*
6. *Check all the (polynomially many) inequalities in the system displayed on Page 11.*

Since all the values $P_j(\theta)$ and the data needed for selective elimination of each of the polynomially many bad dominant pairs $(\theta,\theta') \in D$ have polynomial encoding length, Algorithm 1 runs in polynomial time, which proves the claim. □

References

[1] H. Gui, R. Müller, and R. V. Vohra: *Dominant strategy mechanisms with multidimensional types.* In: *Computing and Markets* (2005).

[2] A. Mas-Colell, M. D. Whinston, and J. R. Green: *Microeconomic Theory.* Oxford University Press (1995).

[3] D. Mookherjee and S. Reichelstein: *Implementation via augmented revelation mechanisms.* Review of Economic Studies 57(3): pp. 453–475 (1990).

[4] R. Müller, A. Perea, and S. Wolf: *Weak monotonicity and Bayes-Nash incentive compatibility.* Games and Economic Behavior 61: pp. 344–358 (2007).

[5] N. Nisan and A. Ronen: *Algorithmic mechanism design.* Games and Economic Behavior 35: pp. 166–196 (2001).

[6] J.-C. Rochet: *A necessary and sufficient condition for rationalizability in a quasilinear context.* Journal of Mathematical Economics 16: pp. 191–200 (1987).

[7] C. Thielen and S. Westphal: *A combinatorial algorithm for strong implementation of social choice functions.* COMSOC 2010.

Sven O. Krumke and Clemens Thielen
Department of Mathematics
University of Kaiserslautern
Paul-Ehrlich-Str. 14
67663 Kaiserslautern, Germany
Email: `{krumke,thielen}@mathematik.uni-kl.de`

A Combinatorial Algorithm for Strong Implementation of Social Choice Functions

Clemens Thielen and Stephan Westphal

Abstract

We consider algorithmic aspects of the classical mechanism design problem of implementing social choice functions. We show how an adaption of the well-known negative cycle criterion for weak implementability can be used to decide the question of implementability in the strong sense when one restricts to incentive compatible direct revelation mechanisms. We derive an efficient combinatorial algorithm that computes the payments of an incentive compatible direct revelation mechanism that strongly implements a given social choice function in dominant strategies or decides that none exist.

Our result complements the results obtained in the companion paper of Krumke and Thielen [3], where a nondeterministic polynomial time algorithm for the more general problem of deciding of strong implementability via *indirect* mechanisms is given. This more general problem is expected to be NP-complete.

1 Introduction

One of the central problems considered in mechanism design is the implementation of social choice functions. In this problem, there are n selfish agents, which must make a collective decision from some finite set X of possible social choices (or *outcomes*). Each agent i has a private value θ_i (called the agent's *type*) that belongs to a finite set Θ_i (called the agent's *type space*) and influences the preferences of all agents over the alternatives in X. Formally, this is modelled by a *valuation function* $V_i : X \times \Theta \to \mathbb{Q}$ for each agent i, which specifies a valuation $V_i(x, \theta)$ that agent i assigns to outcome $x \in X$ when the vector of types of all agents is $\theta \in \Theta = \Theta_1 \times \cdots \times \Theta_n$. The type space Θ_i of agent i is public knowledge, but only agent i knows the true value of θ_i. Every agent i reports a claimed value $\theta'_i \in \Theta_i$ (a *bid*) for her type, and the resulting collective decision is given by a social choice function $f : \Theta \to X$ that maps vectors of bids of the agents to outcomes in X. A *mechanism* $\Gamma_{(f,P)}$ in this setting is given by a payment $P_i(\theta')$ to each agent i that depends on the vector θ' of bids and is used to motivate the agents to report their types truthfully.

When the concept of *weak* implementation is used, a mechanism is said to *implement* the social choice function f if truthfully reporting her type is a *dominant strategy* for every agent, i.e., it maximizes the sum of the agent's valuation and her payment for every possible behavior of the other agents and for every possible vector θ of true types.† The more robust concept of implementation called *strong implementation* (also known as *full implementation*) additionally requires that all other dominant strategy equilibria of the mechanism yield the same outcomes as truthful reporting, so the desired social choices are obtained independently of the equilibrium that is actually played by the agents.

It is easy to see that weak implementation of a given social choice function can be expressed as a system of linear inequalities, in which the variables correspond to the payments. Rochet [6] observed that this system can be interpreted as the problem of finding node potentials in complete, directed graphs on the agents' type spaces with changes of valuations as arc weights. Hence, an implementation exists if and only if there is no negative cycle in

†Note that there are also other notions of implementation, e.g., implementation in Bayes Nash equilibrium. In this paper, we only consider implementation in dominant strategies.

these graphs. Later, it was shown by Gui et al.[2] that it suffices to consider node potentials in smaller graphs on the set X of possible outcomes.

In this paper, we show how the above node potential interpretations of the weak implementability problem can be adapted for deciding also *strong* implementation. Here, some of the inequalities in the linear system have to be fulfilled with strict inequality, i.e., a point in the relative interior of the corresponding polyhedron is sought. We show how such a point can be found by an efficient combinatorial algorithm that perturbates a node potential corresponding to a weak implementation such that the reduced cost of some arcs in the graphs becomes strictly positive, which corresponds to the strict inequalities in the system. To do so, all arcs whose inequalities are already strictly fulfilled are deleted and depth first search is used to find nodes with no outgoing arcs, whose potential can then be perturbated. Furthermore, we use contraction techniques to handle cycles of weight zero. Using these methods, our algorithm computes the payments of a strong implementation of the given social choice function or decides that none exist. The running time is linear in $|\Theta|$, which usually is the largest part of the input. In public project settings, for example, $|\Theta|$ can be quite large, whereas $|X|$ is usually two (the project is either done or not).

We remark that there is also a more general definition of a mechanism, where each agent i is allowed to bid a value s_i from an arbitrary set S_i of bids instead of just reporting a claimed value for her type. A classical result known as the *Revelation Principle* (cf. [4, p. 871]) states that, when considering *weak* implementation, it imposes no loss of generality to restrict to *incentive compatible direct revelation mechanisms* as defined above. For *strong* implementation, it is known that it suffices to consider *augmented revelation mechanisms*, in which the set Θ_i of types of each agent i is a subset of the set S_i of her possible bids (cf. [5] for Bayesian equilibria and the companion paper of Krumke and Thielen [3] for dominant strategies). Most strongly implementable social choice functions can, however, be strongly implemented via incentive compatible direct revelation mechanisms. Thus, since the general problem of deciding strong implementability via augmented revelation mechanisms is expected to be computationally intractable (until recently, it was not even known to belong to NP and it is suspected to be NP-complete, cf. [3]), it makes sense to restrict to incentive compatible direct revelation mechanisms also for strong implementation.

2 The Algorithm

We now present our algorithm for strong implementation of social choice functions. Formally, the problem is defined as follows:

Definition 1 (The Strong Implementability Problem)**.**

INSTANCE: *The number n of agents, the set X of possible social choices, the sets Θ_i of possible types of the agents, the valuation functions $V_i : X \times \Theta \to \mathbb{Q}$, and the social choice function $f : \Theta \to X$.*

TASK: *Compute payments $P_i : \Theta \to \mathbb{Q}$, $i = 1, \ldots, n$, to the agents such that the mechanism $\Gamma_{(f,P)}$ strongly implements f, or decide that none exist.*

The encoding length of an instance of Strong Implementability can be calculated as follows: For every valuation function $V_i : X \times \Theta \to \mathbb{Q}$, we need to store $|X| \cdot |\Theta|$ rational numbers. The social choice function $f : \Theta \to X$ has encoding length $|\Theta| \cdot \log(|X|)$. Thus, the encoding length of an instance of Strong Implementability is in $\Omega(|X| \cdot |\Theta| \cdot n)$.

We start our analysis by formulating a system of linear inequalities whose solutions correspond to the values of payment functions P_i needed to implement a social choice function f. Denoting the $(n-1)$-dimensional vector resulting from an n-vector v when the i-th component is deleted by $v_{-i} := (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$, this linear system in the variables $P_i(\theta')$, $i = 1, \ldots, n$, $\theta' \in \Theta$, can be written as follows:

For all $i \in N$, $(\theta_i, \theta'_i) \in \Theta_i^2$, and $\theta_{-i}, \theta'_{-i} \in \Theta_{-i}$:

$$V_i(f(\theta'_i, \theta'_{-i}), \theta) + P_i(\theta'_i, \theta'_{-i}) \leq V_i(f(\theta_i, \theta'_{-i}), \theta) + P_i(\theta_i, \theta'_{-i}) \tag{1}$$

For all $i \in N$ and $(\theta_i, \theta'_i) \in \Theta_i^2$ with $f(\theta_i, \bar{\theta}_{-i}) \neq f(\theta'_i, \bar{\theta}_{-i})$ for *some* $\bar{\theta}_{-i} \in \Theta_{-i}$, there exists $\theta_{-i}, \theta'_{-i} \in \Theta_{-i}$ such that:

$$V_i(f(\theta'_i, \theta'_{-i}), \theta) + P_i(\theta'_i, \theta'_{-i}) < V_i(f(\theta_i, \theta'_{-i}), \theta) + P_i(\theta_i, \theta'_{-i}) \tag{2}$$

Here, the Inequalities (1) encode that truthfully reporting her type is a dominant strategy for each agent i: For every type θ_i of agent i, reporting θ_i truthfully is at least as good as reporting any other possible type θ'_i, no matter what the type vector θ_{-i} and the bid vector θ'_{-i} of the other agents are. Hence, the first half of the system encodes that the social choice function f is *weakly* implemented by the mechanism $\Gamma_{(f,P)}$. For *strong* implementation, the payments must additionally satisfy the strict Inequalities (2), which encode that there are no dominant strategies for any agent that yield outcomes different from the ones obtained by truthful reporting: If the second condition in the system was violated for some pair (θ_i, θ'_i), bidding θ'_i would always be optimal for agent i when her type is θ_i, and she could change the outcome by bidding θ'_i instead of θ_i in the case that the vector of types of the other agents is $\bar{\theta}_{-i}$ and they report their types truthfully.

We now reformulate the system in order to be able to solve it efficiently via shortest path computations in directed graphs. For every agent $i \in N$ and every fixed pair $(\theta_{-i}, \theta'_{-i}) \in \Theta_{-i}^2$ of a type vector and a bid vector of the other agents, we define a function $c_i^{(\theta_{-i}, \theta'_{-i})} : \Theta_i^2 \to \mathbb{Q}$ by

$$c_i^{(\theta_{-i}, \theta'_{-i})}(\theta_i, \theta'_i) := V_i(f(\theta_i, \theta'_{-i}), \theta) - V_i(f(\theta'_i, \theta'_{-i}), \theta) \quad \forall\, \theta_i, \theta'_i \in \Theta_i.$$

Using this notation, we can rewrite the above system of inequalities as follows:

For all $i \in N$, $(\theta_i, \theta'_i) \in \Theta_i^2$, and $\theta_{-i}, \theta'_{-i} \in \Theta_{-i}$:

$$P_i(\theta'_i, \theta'_{-i}) - P_i(\theta_i, \theta'_{-i}) \leq c_i^{(\theta_{-i}, \theta'_{-i})}(\theta_i, \theta'_i) \tag{3}$$

For all $i \in N$ and $(\theta_i, \theta'_i) \in \Theta_i^2$ with $f(\theta_i, \bar{\theta}_{-i}) \neq f(\theta'_i, \bar{\theta}_{-i})$ for *some* $\bar{\theta}_{-i} \in \Theta_{-i}$, there exists $\theta_{-i}, \theta'_{-i} \in \Theta_{-i}$ such that:

$$P_i(\theta'_i, \theta'_{-i}) - P_i(\theta_i, \theta'_{-i}) < c_i^{(\theta_{-i}, \theta'_{-i})}(\theta_i, \theta'_i) \tag{4}$$

Observe that the left-hand sides of the Inequalities (3) and (4) are independent of the type vector θ_{-i} of all agents except i. Defining

$$\underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) := \min_{\theta_{-i} \in \Theta_{-i}} c_i^{(\theta_{-i}, \theta'_{-i})}(\theta_i, \theta'_i) = \min_{\theta_{-i} \in \Theta_{-i}} \Big(V_i(f(\theta_i, \theta'_{-i}), \theta) - V_i(f(\theta'_i, \theta'_{-i}), \theta) \Big)$$

and

$$\bar{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) := \max_{\theta_{-i} \in \Theta_{-i}} c_i^{(\theta_{-i}, \theta'_{-i})}(\theta_i, \theta'_i) = \max_{\theta_{-i} \in \Theta_{-i}} \Big(V_i(f(\theta_i, \theta'_{-i}), \theta) - V_i(f(\theta'_i, \theta'_{-i}), \theta) \Big)$$

for $\theta_i, \theta'_i \in \Theta_i$, we can, thus, rewrite the system of inequalities as

For all $i \in N$, $(\theta_i, \theta'_i) \in \Theta_i^2$, and $\theta'_{-i} \in \Theta_{-i}$:

$$P_i(\theta'_i, \theta'_{-i}) - P_i(\theta_i, \theta'_{-i}) \leq \underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) \tag{5}$$

For all $i \in N$ and $(\theta_i, \theta'_i) \in \Theta_i^2$ with $f(\theta_i, \bar{\theta}_{-i}) \neq f(\theta'_i, \bar{\theta}_{-i})$ for *some* $\bar{\theta}_{-i} \in \Theta_{-i}$, there exists $\theta'_{-i} \in \Theta_{-i}$ such that:

$$P_i(\theta'_i, \theta'_{-i}) - P_i(\theta_i, \theta'_{-i}) < \bar{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) \tag{6}$$

Observe that, whenever $\underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i) < \bar{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i)$ for some $\theta'_{-i} \in \Theta_{-i}$, the second condition follows automatically from the first one for this pair (θ_i,θ'_i). Hence, the system reduces to

For all $i \in N$, $(\theta_i,\theta'_i) \in \Theta_i^2$, and $\theta'_{-i} \in \Theta_{-i}$:

$$P_i(\theta'_i,\theta'_{-i}) - P_i(\theta_i,\theta'_{-i}) \le \underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i) \tag{7}$$

For all $i \in N$ and $(\theta_i,\theta'_i) \in \Theta_i^2$ with $f(\theta_i,\bar{\theta}_{-i}) \neq f(\theta'_i,\bar{\theta}_{-i})$ for *some* $\bar{\theta}_{-i} \in \Theta_{-i}$ and $\underline{c}_i^{\bar{\theta}_{-i}}(\theta_i,\theta'_i) = \bar{c}_i^{\bar{\theta}_{-i}}(\theta_i,\theta'_i)$ *for all* $\bar{\theta}_{-i} \in \Theta_{-i}$ with $f(\theta_i,\bar{\theta}_{-i}) \neq f(\theta'_i,\bar{\theta}_{-i})$, there exists $\theta'_{-i} \in \Theta_{-i}$ such that $f(\theta_i,\theta'_{-i}) \neq f(\theta'_i,\theta'_{-i})$ and

$$P_i(\theta'_i,\theta'_{-i}) - P_i(\theta_i,\theta'_{-i}) < \underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i). \tag{8}$$

Moreover, for every fixed agent i and $\theta'_{-i} \in \Theta_{-i}$, consider a pair $(\theta_i,\theta'_i) \in \Theta_i^2$ of types of agent i such that $f(\theta_i,\theta'_{-i}) = f(\theta'_i,\theta'_{-i}) =: x \in X$. Then we have

$$\begin{aligned}
\underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i) &= \min_{\theta_{-i}\in\Theta_{-i}} \Big(V_i(f(\theta_i,\theta'_{-i}),\theta) - V_i(f(\theta'_i,\theta'_{-i}),\theta) \Big) \\
&= \min_{\theta_{-i}\in\Theta_{-i}} \Big(V_i(x,\theta) - V_i(x,\theta) \Big) = 0
\end{aligned}$$

and analogously $\underline{c}_i^{\theta'_{-i}}(\theta'_i,\theta_i) = 0$. Hence, the Inequalities (7) corresponding to (θ_i,θ'_i) imply that

$$P_i(\theta'_i,\theta'_{-i}) - P_i(\theta_i,\theta'_{-i}) \quad \le \quad 0 \quad \text{and} \quad P_i(\theta_i,\theta'_{-i}) - P_i(\theta'_i,\theta'_{-i}) \le 0,$$

which yields $P_i(\theta'_i,\theta'_{-i}) = P_i(\theta_i,\theta'_{-i})$. Thus, for fixed i and fixed $\theta'_{-i} \in \Theta_{-i}$, the payment $P_i(\theta_i,\theta'_{-i})$ does in fact only depend on the outcome $f(\theta_i,\theta'_{-i})$ chosen when agent i bids θ_i. Hence, for every $x \in X$ that results as the outcome $f(\theta_i,\theta'_{-i})$ for some $\theta_i \in \Theta_i$, we can define

$$P_i^{\theta'_{-i}}(x) := P_i(\theta_i,\theta'_{-i}) \quad \text{for some } \theta_i \in \Theta_i \text{ with } f(\theta_i,\theta'_{-i}) = x.$$

Writing

$$C(\theta_i,\theta'_i) := \{\theta'_{-i} \in \Theta_{-i} : f(\theta_i,\theta'_{-i}) \neq f(\theta'_i,\theta'_{-i})\},$$

we can, thus, write our system of inequalities as

For all $i \in N$, $(\theta_i,\theta'_i) \in \Theta_i^2$, and $\theta'_{-i} \in \Theta_{-i}$ with $f(\theta_i,\theta'_{-i}) \neq f(\theta'_i,\theta'_{-i})$:

$$P_i^{\theta'_{-i}}(f(\theta'_i,\theta'_{-i})) - P_i^{\theta'_{-i}}(f(\theta_i,\theta'_{-i})) \le \underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i) \tag{9}$$

For all $i \in N$ and $(\theta_i,\theta'_i) \in \Theta_i^2$ with $C(\theta_i,\theta'_i) \neq \emptyset$ and $\underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i) = \bar{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i)$ for all $\theta'_{-i} \in C(\theta_i,\theta'_i)$: There exists $\theta'_{-i} \in C(\theta_i,\theta'_i)$ such that

$$P_i^{\theta'_{-i}}(f(\theta'_i,\theta'_{-i})) - P_i^{\theta'_{-i}}(f(\theta_i,\theta'_{-i})) < \underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i). \tag{10}$$

As observed above, the left-hand sides of Inequalities (9) and (10) are now independent of θ_i and θ'_i as long as the respective outcomes $f(\theta_i,\theta'_{-i})$ and $f(\theta'_i,\theta'_{-i})$ do not change. Defining $K^{\theta'_{-i}}(x) := \{\theta_i \in \Theta_i : f(\theta_i,\theta'_{-i}) = x\}$ for every $x \in X$, $W(\theta'_{-i}) := \{x \in X : K^{\theta'_{-i}}(x) \neq \emptyset\}$, and

$$c_i^{\theta'_{-i}}(x,x') := \min_{\substack{\theta_i,\theta'_i\in\Theta_i: \\ f(\theta_i,\theta'_{-i})=x \\ f(\theta'_i,\theta'_{-i})=x'}} \underline{c}_i^{\theta'_{-i}}(\theta_i,\theta'_i)$$

for all $x, x' \in W(\theta'_{-i})$, the system can be written as

For all $i \in N$, $\theta'_{-i} \in \Theta_{-i}$, and $x, x' \in W(\theta'_{-i})$ with $x \neq x'$:

$$P_i^{\theta'_{-i}}(x') - P_i^{\theta'_{-i}}(x) \leq c_i^{\theta'_{-i}}(x, x') \tag{11}$$

For all $i \in N$ and $(\theta_i, \theta'_i) \in \Theta_i^2$ with $C(\theta_i, \theta'_i) \neq \emptyset$ and $\underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) = \bar{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i)$ for all $\theta'_{-i} \in C(\theta_i, \theta'_i)$: There exists $\theta'_{-i} \in C(\theta_i, \theta'_i)$ such that $\theta_i \in K^{\theta'_{-i}}(x), \theta'_i \in K^{\theta'_{-i}}(x')$ and

$$P_i^{\theta'_{-i}}(x') - P_i^{\theta'_{-i}}(x) < c_i^{\theta'_{-i}}(\theta_i, \theta'_i). \tag{12}$$

Now observe that, whenever $\theta_i \in K^{\theta'_{-i}}(x), \theta'_i \in K^{\theta'_{-i}}(x')$, and $c_i^{\theta'_{-i}}(x, x') < \underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i)$ for *some* $\theta'_{-i} \in \Theta_{-i}$, the second condition follows automatically from the first one for this pair (θ_i, θ'_i). Hence, we just have to consider the second condition for the pairs (θ_i, θ'_i) for which $c_i^{\theta'_{-i}}(x, x') = \underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i)$ for *all* $\theta'_{-i} \in \Theta_{-i}$, x, x' with $\theta_i \in K^{\theta'_{-i}}(x)$ and $\theta'_i \in K^{\theta'_{-i}}(x')$. Thus, the system can be rewritten as

For all $i \in N$, $\theta'_{-i} \in \Theta_{-i}$, and $x, x' \in W(\theta'_{-i})$ with $x \neq x'$:

$$P_i^{\theta'_{-i}}(x') - P_i^{\theta'_{-i}}(x) \leq c_i^{\theta'_{-i}}(x, x') \tag{13}$$

For all $i \in N$ and $(\theta_i, \theta'_i) \in \Theta_i^2$ with $C(\theta_i, \theta'_i) \neq \emptyset$, $\underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) = \bar{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i)$ for all $\theta'_{-i} \in C(\theta_i, \theta'_i)$, and $c_i^{\theta'_{-i}}(x, x') = \underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i)$ for all $\theta'_{-i} \in C(\theta_i, \theta'_i)$, $x, x' \in W$ with $\theta_i \in K^{\theta'_{-i}}(x)$ and $\theta'_i \in K^{\theta'_{-i}}(x')$: There exists $\theta'_{-i} \in C(\theta_i, \theta'_i)$ such that $\theta_i \in K^{\theta'_{-i}}(x), \theta'_i \in K^{\theta'_{-i}}(x')$ and

$$P_i^{\theta'_{-i}}(x') - P_i^{\theta'_{-i}}(x) < c_i^{\theta'_{-i}}(x, x'). \tag{14}$$

Finally, the second condition in the system can be reformulated as follows: All conditions on the pairs (θ_i, θ'_i) do in fact only depend on the second value θ'_i through the corresponding outcomes $f(\theta'_i, \theta'_{-i})$. In particular, the values

$$\begin{aligned} \underline{c}_i^{\theta'_{-i}}(\theta_i, x') &:= \underline{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) \text{ for some } \theta'_i \in K^{\theta'_{-i}}(x') \\ \bar{c}_i^{\theta'_{-i}}(\theta_i, x') &:= \bar{c}_i^{\theta'_{-i}}(\theta_i, \theta'_i) \text{ for some } \theta'_i \in K^{\theta'_{-i}}(x') \end{aligned}$$

are well-defined for all $x \in W(\theta'_{-i})$. Defining

$$C(\theta_i, x') := \{\theta'_{-i} \in \Theta_{-i} : x' \in W(\theta'_{-i}), f(\theta_i, \theta'_{-i}) \neq x'\},$$

we can, hence, rewrite the system as

For all $i \in N$, $\theta'_{-i} \in \Theta_{-i}$, and $x, x' \in W(\theta'_{-i})$ with $x \neq x'$:

$$P_i^{\theta'_{-i}}(x') - P_i^{\theta'_{-i}}(x) \leq c_i^{\theta'_{-i}}(x, x') \tag{15}$$

For all $i \in N$, $\theta_i \in \Theta_i$, and $x' \in X$ such that $C(\theta_i, x') \neq \emptyset$, $\underline{c}_i^{\theta'_{-i}}(\theta_i, x') = \bar{c}_i^{\theta'_{-i}}(\theta_i, x')$ for all $\theta'_{-i} \in C(\theta_i, x')$, and $c_i^{\theta'_{-i}}(f(\theta_i, \theta'_{-i}), x') = \underline{c}_i^{\theta'_{-i}}(\theta_i, x')$ for all $\theta'_{-i} \in C(\theta_i, x')$: There exists $\theta'_{-i} \in C(\theta_i, x')$ such that $f(\theta_i, \theta'_{-i}) = x$ and

$$P_i^{\theta'_{-i}}(x') - P_i^{\theta'_{-i}}(x) < c_i^{\theta'_{-i}}(x, x'). \tag{16}$$

Having reformulated the system as above, we can now solve it efficiently via shortest path computations. For every agent i and every fixed vector $\theta'_{-i} \in \Theta_{-i}$ of bids of the other agents, the Inequalities (15) corresponding to i and θ'_{-i} are exactly equivalent to the values $P_i^{\theta'_{-i}}(x)$, $x \in W(\theta'_{-i})$, defining a node potential in the complete, directed graph $G_i(\theta'_{-i})$ on the set $W(\theta'_{-i})$ with the cost of the arc from outcome x to x' given as $c_i^{\theta'_{-i}}(x, x')$. Thus, we can compute a solution $P_i^{\theta'_{-i}}(x)$, $x \in W(\theta'_{-i})$, of the Inequalities (15) by computing the shortest path distances from an arbitrary node x to all other nodes in the graph $G_i(\theta'_{-i})$ for every $i \in N$ and every $\theta'_{-i} \in \Theta_{-i}$. This can be done efficiently with the Bellman-Ford Algorithm (cf. for example [7]). In the case that one of the graphs $G_i(\theta'_{-i})$ contains a negative cycle (so we cannot compute a node potential in this graph), the Inequalities (15) do not have a solution, which implies that the given social choice function f cannot be implemented at all (not even *weakly*). Otherwise, the shortest path distances $P_i^{\theta'_{-i}}(x)$, $x \in W(\theta'_{-i})$, computed by the Bellman-Ford Algorithm yield payments $P_i(\theta')$ solving Inequalities (1) of the original system (and, thus, weakly implementing the social choice function f) by setting $P_i(\theta') := P_i^{\theta'_{-i}}(f(\theta'))$ for all $i = 1, \ldots, n$, $\theta' \in \Theta$.

For strong implementation, we now show how we can modify a solution of the Inequalities (15) obtained by shortest path computations such that it also satisfies the strict Inequalities (16) and, hence, corresponds to payments P of a mechanism $\Gamma_{(f,P)}$ that strongly implements f. This is done via the following procedure:

For every agent i and every $\theta'_{-i} \in \Theta_{-i}$, we again consider the graph $G_i(\theta'_{-i})$ and the corresponding payments $P_i^{\theta'_{-i}}(x)$ for $x \in W(\theta'_{-i})$. We delete all arcs (x, x') from $G_i(\theta'_{-i})$ for which the corresponding Inequality (15) is already fulfilled with strict inequality. After doing so, we also delete all isolated nodes, i.e., all nodes x with empty adjacency list $\mathrm{Adj}(x)$. In the remaining graph, which contains only arcs for which the corresponding Inequality (15) holds with equality, we then search for a node x with no outgoing arcs, i.e., with $\mathrm{Adj}^+(x) = \emptyset$. For such a node x, the value

$$\epsilon(x) := \min_{x' \in W(\theta'_{-i})} \left(c_i^{\theta'_{-i}}(x, x') - P_i^{\theta'_{-i}}(x') + P_i^{\theta'_{-i}}(x) \right)$$

is strictly positive, so we can lower $P_i^{\theta'_{-i}}(x)$ by $\epsilon(x)/2$ without violating any of the inequalities in our system. After doing so, all inequalities which were fulfilled with strict inequality before are still fulfilled with strict inequality, but, additionally, all the inequalities corresponding to arcs with end node x are now fulfilled with strict inequality, so we can delete these arcs and the node x from the graph $G_i(\theta'_{-i})$.

To find a node x in $G_i(\theta'_{-i})$ with $\mathrm{Adj}^+(x) = \emptyset$, we use depth-first search (DFS) starting with an arbitrary node in the graph. Doing so, we either find a node with no outgoing arcs, or we discover a directed cycle. In the first case, we lower the payment of the node as in the procedure described above and continue the DFS-procedure at the node considered before as long as there are still nodes remaining in the graph. In the case that we find a directed cycle C, all Inequalities (15) on C are fulfilled with equality and adding them up yields

$$0 = \sum_{(x_k, x_l) \in C} \left(P_i^{\theta'_{-i}}(x_l) - P_i^{\theta'_{-i}}(x_k) \right) = \sum_{(x_k, x_l) \in C} c_i^{\theta'_{-i}}(x_k, x_l), \tag{17}$$

where the first equality follows since C is a cycle. On the other hand, if the strict Inequality (16) corresponding to i and θ'_{-i} was fulfilled for any arc on C, we would obtain

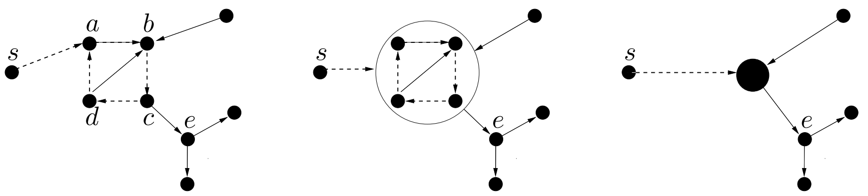
$$0 = \sum_{(x_k, x_l) \in C} \left(P_i^{\theta'_{-i}}(x_l) - P_i^{\theta'_{-i}}(x_k) \right) < \sum_{(x_k, x_l) \in C} c_i^{\theta'_{-i}}(x_k, x_l),$$

contradicting (17). Hence, the strict Inequality (16) cannot be satisfied for any arc on the cycle C. In this case, we contract all nodes on C to a single supernode and continue the DFS-procedure at this new node (see Figure 1).

When the above procedure terminates, every arc (x, x') in the complete, directed graph on $W(\theta'_{-i})$ is either contained in a cycle of arcs for which the corresponding Inequalities (15) are fulfilled with equality (so the Inequality (15) of (x, x') cannot be made strict in this graph), or the inequality corresponding to (x, x') is fulfilled with strict inequality.

Algorithm 1 summarizes the above discussion. For each agent i, the algorithm first calculates the set A of pairs $(\theta_i, x') \in \Theta_i \times X$ for which some strict Inequality (16) must hold. Then, for every possible bid vector $\theta'_{-i} \in \Theta_{-i}$ of the other agents, it calculates the set $W(\theta'_{-i})$ and a node potential in the complete, directed graph on $W(\theta'_{-i})$ via the Bellman-Ford Algorithm and perturbates this node potential such that each arc (x, x') is either contained in a cycle of arcs for which the corresponding Inequalities (15) are fulfilled with equality, or the inequality corresponding to (x, x') is fulfilled with strict inequality. The pairs $(\theta_i, x') \in \Theta_i \times X$ for which some strict Inequality (16) holds are then deleted from A, and the algorithm continues with the next bid vector $\theta'_{-i} \in \Theta_{-i}$ of the other agents. If the set A is still nonempty after processing all possible vectors $\theta'_{-i} \in \Theta_{-i}$, the remaining pairs $(\theta_i, x') \in A$ are pairs for which the second condition of the system cannot be satisfied, so the system does not have a solution. Otherwise, the algorithm continues with the next agent i. As a node in a graph can represent several outcomes due to previous contractions, the outcomes corresponding to each node u are stored in a set Outcomes(u).

The DFS-procedure used to find a node x in $G_i(\theta'_{-i})$ with $\mathrm{Adj}^+(x) = \emptyset$ is implemented in the procedure PROCESS-GRAPH (Algorithm 2). $\pi(v)$ and color(v) denote the predecessor and the current state (GRAY = already visited, WHITE = not yet visited) of a node v, respectively. A stack-like data structure S is used to store the nodes to be processed next. It supports the operations FIRST(S) (returns first element of S), PUSH_FRONT(S, v) (inserts v as first element of S), and REMOVE(S, v) (deletes v from S). More details on depth-first can be found in Cormen et al. [1].



(a) DFS finds a cycle. (b) Contraction of the cycle (c) Graph after contraction

Figure 1: Contraction of a cycle to a supernode

Algorithm 1.

for all $i \in N$ **do**
  for all $\theta'_{-i} \in \Theta_{-i}$, $\theta_i \in \Theta_i$, $x' \in X$ **do**
    $\underline{c}_i^{\theta'_{-i}}(\theta_i, x') := \min_{\theta_{-i} \in \Theta_{-i}} \big(V_i(f(\theta_i, \theta'_{-i}), \theta) - V_i(x', \theta)\big)$
    $\overline{c}_i^{\theta'_{-i}}(\theta_i, x') := \max_{\theta_{-i} \in \Theta_{-i}} \big(V_i(f(\theta_i, \theta'_{-i}), \theta) - V_i(x', \theta)\big)$
  end for
  //Calculate the sets $C(\theta_i, x')$

for all $\theta'_{-i} \in \Theta_{-i}$ **do**
$K^{\theta'_{-i}}(x) := \{\theta_i \in \Theta_i : f(\theta_i, \theta'_{-i}) = x\}$ for $x \in X$.
$W(\theta'_{-i}) := \{x \in X : K^{\theta'_{-i}}(x) \neq \emptyset\}$
for all $\theta_i \in \Theta_i, x' \in W(\theta'_{-i})$ **do**
if $f(\theta_i, \theta'_{-i}) \neq x'$ **then**
$C(\theta_i, x') := C(\theta_i, x') \cup \{\theta'_{-i}\}$
end if
end for
end for
//Find the set A of pairs $(\theta_i, x') \in \Theta_i \times X$ for which one inequality must be strict
$A := \Theta_i \times X$
for all $\theta'_{-i} \in \Theta_{-i}$ **do**
for all $(x, x') \in W(\theta'_{-i}) \times W(\theta'_{-i})$ **do**
$c_i^{\theta'_{-i}}(x, x') := \min_{\theta_i \in K^{\theta'_{-i}}(x)} \underline{c}_i^{\theta'_{-i}}(\theta_i, x')$
end for
for all $\theta_i \in \Theta_i$, $x' \in X$ **do**
if $C(\theta_i, x') = \emptyset$, $\underline{c}_i^{\theta'_{-i}}(\theta_i, x') \neq \bar{c}_i^{\theta'_{-i}}(\theta_i, x')$, or $c_i^{\theta'_{-i}}(f(\theta_i, \theta'_{-i}), x') \neq \underline{c}_i^{\theta'_{-i}}(\theta_i, x')$ **then**
$A := A \setminus \{(\theta_i, x')\}$
end if
end for
end for
for all $\theta'_{-i} \in \Theta_{-i}$ **do**
Choose $x \in W(\theta'_{-i})$ arbitrarily.
Apply the Bellman-Ford Algorithm to the complete, directed graph G with node set $W := W(\theta'_{-i})$, arc costs $c(x, x') := c_i^{\theta'_{-i}}(x, x')$, and start node x.
if G contains a negative cycle **then**
STOP: f cannot be implemented at all.
else
Denote the node potential obtained by the Bellman-Ford Algorithm by P.
$V(G) := \emptyset$, $E(G) := \emptyset$
for all $(x, x') \in W \times W$ with $x \neq x'$ **do**
if $P(x') - P(x) = c(x, x')$ **then**
$V(G) := V(G) \cup \{x, x'\}$
$E(G) := E(G) \cup \{(x, x')\}$
end if
end for
$G := (V(G), E(G))$
PROCESS-GRAPH(G, W, c, P)
end if
for all $(x, x') \in W \times W$ with $x \neq x'$ **do**
if $P(x') - P(x) < c(x, x')$ **then**
for all $\theta_i \in K^{\theta'_{-i}}(x)$ **do**
$A := A \setminus \{(\theta_i, x')\}$
end for
end if
end for
for all $\theta'_i \in \Theta_i$ **do**
$P_i(\theta'_i, \theta'_{-i}) := P(f(\theta'_i, \theta'_{-i}))$

```
      end for
   end for
   if $A \neq \emptyset$ then
      STOP: No payments $P_i(\theta')$ exist such that $\Gamma_{(f,P)}$ strongly implements $f$.
   end if
end for
STOP: The mechanism $\Gamma_{(f,P)}$ with payments $P_i(\theta')$ strongly implements $f$.
```

Algorithm 2. PROCESS-GRAPH(G, W, c, P)

```
for all $u \in V(G)$ do
   color$(u)$ := WHITE
   $\pi(u)$ := NIL
   Outcomes$(u) := \{u\}$
end for
while $w \in V(G)$ exists do
   $S := \{w\}$
   while $S$ not empty do
      $u$ =FIRST$(S)$
      color$(u)$ := GRAY
      if $\mathrm{Adj}^+(u) = \emptyset$ then
         PERTURBATE$(u, W, S, c, P)$
      else
         for all $v \in \mathrm{Adj}^+(u)$ do
            if color$(v)$ = WHITE then
               $\pi(v) := u$
               PUSH_FRONT$(S, v)$
            end if
         end for
         for all $v \in \mathrm{Adj}^+(u)$ do
            if color$(v)$ = GRAY then
               CONTRACT$(u, v, S)$
               break
            end if
         end for
      end if
   end while
end while
```

Algorithm 3. PERTURBATE(u, W, S, c, P)

```
$\epsilon := \min_{x \in \text{ Outcomes}(u)} \min_{x' \in W} c(x, x') - P(x') + P(x)$
for all $x \in$ Outcomes$(u)$ do
   $P(x) := P(x) - \epsilon/2$
end for
for all $w \in \mathrm{Adj}^-(u)$ do
   $\mathrm{Adj}^+(w) := \mathrm{Adj}^+(w) \setminus \{u\}$
end for
$V(G) := V(G) \setminus \{u\}$
REMOVE$(S, u)$
```

Algorithm 4. CONTRACT(u, v, S)

```
C := {v}
w := u
repeat
    C := C ∪ {w}
    w := π(w)
until w = v
//Introduce new supernode v_C
V(G) := V(G) ∪ {v_C}
color(v_C) := WHITE
π(v_C) := π(v)
Outcomes(v_C) := ∅
for all w ∈ C do
    Outcomes(v_C) := Outcomes(v_C) ∪ Outcomes(w)
    Adj⁺(v_C) := Adj⁺(v_C) ∪ (Adj⁺(w) \ C)
    Adj⁻(v_C) := Adj⁻(v_C) ∪ (Adj⁻(w) \ C)
    REMOVE(S, w)
end for
for all w ∈ Adj⁺(v_C) do
    Adj⁻(w) := Adj⁻(w) ∪ {v_C}
    Adj⁻(w) := Adj⁻(w) \ C
end for
for all w ∈ Adj⁻(v_C) do
    Adj⁺(w) := Adj⁺(w) ∪ {v_C}
    Adj⁺(w) := Adj⁺(w) \ C
end for
V(G) := V(G) \ C
PUSH_FRONT (S, v_C)
```

Theorem 1. *Algorithm 1 correctly computes the payments P of a mechanism $\Gamma_{(f,P)}$ that strongly implements the given social choice function f in dominant strategies or decides that no such payments exist. The algorithm runs in time $\mathcal{O}(n \cdot |\Theta| \cdot |X|^3)$.*

Proof. For every fixed $\theta'_{-i} \in \Theta_{-i}$, the arc costs $c(x, x')$ calculated in the algorithm are given by

$$\begin{aligned} c_i^{\theta'_{-i}}(x, x') &= \min_{\theta_i \in K^{\theta'_{-i}}(x)} \underline{c}_i^{\theta'_{-i}}(\theta_i, x') \\ &= \min_{\theta_i \in K^{\theta'_{-i}}(x)} \min_{\theta_{-i} \in \Theta_{-i}} \Big(V_i(f(\theta_i, \theta'_{-i}), \theta) - V_i(x', \theta) \Big) \\ &= \min_{\substack{\theta_i, \theta'_i \in \Theta_i: \\ f(\theta_i, \theta'_{-i}) = x \\ f(\theta'_i, \theta'_{-i}) = x'}} \min_{\theta_{-i} \in \Theta_{-i}} \Big(V_i(f(\theta_i, \theta'_{-i}), \theta) - V_i(f(\theta'_i, \theta'_{-i}), \theta) \Big), \end{aligned}$$

which equals the arc costs $c_i^{\theta'_{-i}}(x, x')$ used in the discussion above. Hence, correctness of the algorithm follows from the arguments preceding the algorithm.

The running time can be estimated as follows: For each agent i, the calculation of the sets $C(\theta_i, x')$ and the set A needs time at most $\mathcal{O}(|\Theta_{-i}| \cdot |\Theta_i| \cdot |X|^2) = \mathcal{O}(|\Theta| \cdot |X|^2)$. In the for-loop starting in Line 28, the application of the Bellman-Ford Algorithm to the graph G in Line 30 needs time $\mathcal{O}(|W|^3) \leq \mathcal{O}(|X|^3)$. In the procedure PROCESS-GRAPH(G, W, c, P) in Line 43, at most $|W| \leq |X|$ contraction or perturbation steps are made since each such step reduces the number of nodes in the graph by at least one. Each call of the procedure CONTRACT(u, v, S) needs time at most $|W|^2 \leq |X|^2$ since each adjacency list and each set Outcomes$(\cdot)$ contains at most $|W|$ nodes and there are at most $|W|$ nodes in the cycle C. Each call of the procedure PERTURBATE(u, W, c, P) needs time at most $|W|^2 \leq |X|^2$ as well. Thus, the overall time needed for the procedure PROCESS-GRAPH(G, W, c, P) is at most $\mathcal{O}(|X|^3)$. Since every iteration of the loop except for the Bellman-Ford Algorithm and the call of PROCESS-GRAPH(G, W, c, P) needs time at most $\mathcal{O}(|\Theta_i| \cdot |X|^2)$, this implies that the overall time required for the for-loop starting in Line 28 is at most $\mathcal{O}(|\Theta_{-i}| \cdot |\Theta_i| \cdot |X|^3) = \mathcal{O}(|\Theta| \cdot |X|^3)$. Since there are n agents, we obtain an overall running time of $n \cdot \mathcal{O}(|\Theta| \cdot |X|^2) + n \cdot \mathcal{O}(|\Theta| \cdot |X|^3) = \mathcal{O}(n \cdot |\Theta| \cdot |X|^3)$ as claimed. $\square$

As already shown, the Weak Implementability Problem can be solved in the same way by just leaving out the steps needed to make sure that the strict inequalities in our system are fulfilled. Hence, the resulting algorithm solves the Weak Implementability Problem in time $\mathcal{O}(n \cdot |\Theta| \cdot |X|^3)$.

References

[1] T. H. Cormen, C. Stein, C. E. Leiserson, and R. L. Rivest: *Introduction to Algorithms*. 3rd edition. MIT Press (2009).

[2] H. Gui, R. Müller, and R. V. Vohra: *Dominant strategy mechanisms with multidimensional types*. In: *Computing and Markets* (2005).

[3] S. O. Krumke and C. Thielen: *Strong implementation of social choice function in dominant strategies*. COMSOC 2010.

[4] A. Mas-Colell, M. D. Whinston, and J. R. Green: *Microeconomic Theory*. Oxford University Press (1995).

[5] D. Mookherjee and S. Reichelstein: *Implementation via augmented revelation mechanisms*. Review of Economic Studies 57(3): pp. 453–475 (1990).

[6] J.-C. Rochet: *A necessary and sufficient condition for rationalizability in a quasilinear context*. Journal of Mathematical Economics 16: pp. 191–200 (1987).

[7] A. Schrijver: *Combinatorial Optimization*, volume 24 of *Algorithms and Combinatorics*. Springer (2003).

Clemens Thielen and Stephan Westphal
Department of Mathematics
University of Kaiserslautern
Paul-Ehrlich-Str. 14
67663 Kaiserslautern, Germany
Email: {thielen,westphal}@mathematik.uni-kl.de

Algorithms for Pareto Stable Assignment

Ning Chen and Arpita Ghosh

Abstract

Motivated by online matching marketplaces, we study stability in a many-to-many market with ties and incomplete preference lists. When preference lists contain ties, stable matchings need not be Pareto optimal. We consider the algorithmic question of computing outcomes that are both Pareto optimal and stable in a many-to-many two-sided market with ties and incomplete lists, where agents on *both* sides can have multi-unit capacities, as well as trade multiple units with the same neighbor.
Our main result is a fast algorithm for computing Pareto-stable assignments for this very general multi-unit matching problem with arbitrary preference lists on both sides, with running time that is polynomial in the *number of agents* in the market, rather than the sum of capacities of all agents.

1 Introduction

A fundamental solution concept in the context of two-sided matching marketplaces is that of *stability*, introduced by Gale and Shapley in their seminal work on stable marriage [10]. In the marriage model, there are n men and n women, each with a strict preference ranking over all members of the other side: a matching between the men and women is *stable* if there is no unmatched man-woman pair who both prefer each other to their current partners. The concept of stability has had enormous influence both on the design of real world matching markets [25] as well as its theory [27] — a number of variants of the stable matching problem have been studied, relaxing or generalizing different assumptions in the original model.

One particularly practical generalization is to relax the requirement of strict and complete preferences over all alternatives to accommodate indifferences and intolerance — a man can have an *incomplete* preference list, *i.e.*, he need not rank all women, and can have *ties*, *i.e.*, he can be indifferent between some women in his preference list (and similarly for women). The introduction of ties and incomplete lists dramatically changes the properties and structure of the set of stable matchings relative to the Gale-Shapley marriage model, and often leads to interesting algorithmic and computational questions in the context of choosing amongst the large set of stable matchings. For instance, man or woman-optimal stable matchings[1] are no longer well-defined [27]; stable matchings need not all have the same cardinality, and the problem of finding the maximum cardinality stable matching becomes NP-hard [16].

One of the most important differences that arises due to indifferences in preference lists, however, is that *stability no longer guarantees Pareto optimality*[2], an observation that has received a great deal of attention in the economics literature (see, for example, [2, 7, 1, 29, 8, 9]). When preference lists contain ties, not all stable matchings are Pareto optimal and in fact, as [8] demonstrates, simply using a matching returned by the Gale-Shapley deferred acceptance algorithm can cause quite a severe loss in efficiency. A natural question, then, is whether one can find a matching which is both stable and Pareto optimal when preferences may contain ties. This question has recently been addressed for the *many-to-one* matching

[1] The outcome of the man-proposing deferred acceptance algorithm with strict complete preferences is a man-optimal stable matching.

[2] A simple example consists of two men and two women, where i_1 strictly prefers j_1 to j_2, but all other nodes are indifferent amongst their possible partners. The matching $(i_1, j_2), (i_2, j_1)$ is stable, but not Pareto optimal since i_1 can be reassigned to j_1 and i_2 to j_2 without making anyone worse off.

model — Erdil and Ergin [8, 9] give an algorithm that finds a Pareto-stable many-to-one matching, with runtime that is polynomial in the total capacity of all agents.

In this paper, we study the problem of efficiently computing a Pareto-stable outcome in a very general *many-to-many* setting with indifferences: agents on *both* sides can have multi-unit capacities, as well as trade multiple units with the same neighbor. (Observe that here, unlike the many-to-one setting, the total capacity of nodes in the graph is not restricted to be polynomial in the size of the graph.) The many-to-many setting has attracted growing interest in the economics literature both because of an increasing number of applications (indeed, a number of online marketplaces are many-to-many since both buyers and sellers have multi-unit demand and supply), and more importantly because of fundamental theoretical differences from the well understood many-to-one setting. We focus here on a *computational* problem that arises from allowing nodes on both sides to have multi-unit capacities— while a naive adaptation of the many-to-one algorithm would return a Pareto-stable assignment, it would do so in time that grows polynomially with the total capacity of all nodes in the graph, rather than the size of the graph itself. We therefore seek a strongly polynomial time algorithm for the problem of computing a Pareto optimal stable assignment.

The computer science literature on algorithms for stable matching in the presence of ties and incomplete lists has largely focused on the problems of deciding the existence of, and computing, strongly-stable and super-stable matchings [15], and computing stable matchings with large size or weight. While the problem of finding a strongly stable matching if it exists can be solved in polynomial time [15, 17] and the resulting outcomes are indeed Pareto optimal, such matchings need not always exist, making them an unsuitable solution concept practically. Also, the problem of finding the maximum cardinality stable matching is NP-hard [16]. The solution concept of Pareto stability offers a strict refinement of the set of stable matchings, and in addition, has the important property that it always exists, and, as we show, can be computed efficiently. Given that choosing a globally optimal stable matching is difficult, Pareto-stable matchings, which are *locally optimal*, are a natural choice amongst stable matchings — a stable matching which is not Pareto optimal unnecessarily compromises efficiency, since it is possible to make some agents strictly better off without compromising the welfare of any other agents.

1.1 Our Results

Our main result is an algorithm that finds a Pareto-optimal stable assignment, with running time that is polynomial in the *number of nodes* in the graph, for a many-to-many two-sided market where: (i) all nodes can have ties and incomplete preference lists over the other side, (ii) nodes on both sides have multi-unit capacities, (iii) there can be multiple edges between a pair (i, j) (*i.e.*, multiple units can be assigned between i and j). While ties and incomplete lists motivate Pareto stability, the actual technical challenge arises due to the multi-unit node capacities, which, unlike in the many-to-one setting of [8, 9], need not be polynomial in the size of the graph.

With unit capacity (matching), a fairly straightforward application of standard notions of augmenting paths and cycles from network flows [18] leads to an algorithm that finds Pareto-optimal stable matchings [29]. A naive approach to the many-to-many matching problem is to simply make identical copies of nodes, one for each unit of its capacity, and compute a Pareto-stable matching for this equivalent instance, using the algorithm for the relatively simple unit supply/demand setting. However, the size of this instance is proportional to the total capacity of all nodes, and therefore will not give us a strongly polynomial time algorithm. Instead, we construct a sequence of modified networks with one copy of a node for *each level in its preference list* (the number of levels in a preference list cannot exceed the number of nodes)— this allows us to correctly define the notion of "improvement edges" (§4.1) when nodes have multiunit capacities. The second challenge is to ensure that once all Pareto-improvements at a certain preference level for a particular node have been found, the reassignments made by the algorithm for a *different* node or level does not *reintroduce* Pareto-improvements for this node and preference level (Example 4.1 demonstrates that this can indeed happen for a only slightly different (and perhaps more natural) network construction). We use maximum flow computations on a series of carefully designed augmented networks such that increases in flow preserve stability and correspond to Pareto improvements in the assignment, and there are no remaining Pareto improvements after all networks have been executed once. The algorithm and its proof of correctness are given in Section 4.

Applications. The many-to-many matching problem has recently attracted growing interest because of a number of applications such as job markets where applicants seek multiple part-time positions [6], auto markets [12], as well as electronic marketplaces such as eBay, and online advertising exchanges. A specific application in the electronic marketplace setting is in the context of *social lending* [4], which is a large and rapidly expanding marketplace for matching lenders and borrowers directly without the use of traditional financial intermediaries. In the social lending marketplace, lenders have preferences over borrowers since they each represent investments with different *risk* levels— so a lender might prefer to invest in some borrowers more than others, even amongst the set of acceptable borrowers. While lenders have explicit preferences over borrowers, the interest rates offered by lenders can be used to define a preference ranking over lenders for the borrower side of the graph as well. The question of how to clear this two-sided matching market leads immediately to our problem of efficiently computing Pareto-stable assignments, since both lenders and borrowers have multiunit capacities (lending budgets and desired loan amounts respectively [4]), with preferences that are incomplete and contain ties[3]. The need for computational efficiency is particularly striking in this setting, since an algorithm that is polynomial in the total capacity of the instance, *i.e.*, the total volume of loans in the market, as opposed to the total number of agents (lenders and borrowers) is clearly not efficient.

The social lending site Zopa, with over 400,000 members and $50 million in loans, already uses a centralized matching system where lenders can specify bids for each category (arranged by credit-rating) of borrowers and a total budget, but not preferences across categories[4]. Our algorithm would permit offering a more expressive bidding language for lenders, which allows specifying preferences across categories in addition to the total budget and bids, by providing a solution for the market-clearing problem.

1.2 Related Work

Two-sided matchings have been studied extensively since the seminal work of Gale and Shapley on stable marriage [10]. There is now a vast literature studying various aspects of the original stable marriage model as well as many of its variants, such as ties in preference lists, incomplete preferences, and weighted edges, as well as non-bipartite versions such as the roommate model. For a nice review of the very large economics literature on the subject, see the book by Roth and Sotomayor [27] and the survey by Roth [25]; for an introduction to the computer science literature addressing algorithmic and computational questions, see, for instance, [11, 3, 14].

[3]Ties are ubiquitous in social lending, since lenders can often only distinguish between borrowers by credit-rating. Preference lists can be incomplete since some borrowers, for instance those with poor credit rating, may not be acceptable to a lender.

[4]A lender can specify separate budgets for each category, but clearly this is a strict subset of the expressiveness offered by allowing budgets along with preferences over categories

The papers most relevant to our work from the stable matching literature are the following. Sotomayor [29] proposes Pareto-stable matchings as a natural solution concept for a many-to-many marketplace and studies structural aspects of Pareto-stable matchings. As previously discussed, Erdil and Ergin [8, 9] study the algorithmic question of finding Pareto-optimal matchings for the *many-to-one* setting and give an algorithm whose running time is polynomial in the sum of capacities of all nodes in the graph.

The many-to-many stable matching problem is far less well-studied, with a small, but growing, body of research, motivated by practical settings such as electronic marketplaces, and job markets where some agents might seek multiple part-time positions [6]. The generalization to multi-unit node capacities on both sides is nontrivial: as Echenique and Oviedo [6] show, even a small number of agents with multi-unit capacity drastically alter the properties of matchings compared to the many-to-one setting. Much of the literature on many-to-many stable matchings focuses on settings without indifferences: Hatfield and Kominers [12] study stability in very general model with bilateral contracts and prove necessary conditions for the existence of stable matchings as well as results regarding the structure of the set of stable matchings. Echenique and Oviedo [6] show the equivalence of different solution concepts under strong substitutability for many-to-many matching, also in a setting with strict preferences. Finally, Malhotra [19] studies the algorithmic question of finding *strongly stable* matchings, if they exist, in a many-to-many matching model with ties and complete lists.

2 Model

There is an underlying bipartite graph M with nodes, or agents, (A, B) and edge set E. The existence of an edge (i, j) means agents $i \in A$ and $j \in B$ are mutually willing to be matched with, or assigned to, each other.

Each node in M has a *capacity constraint*, which is the maximum number of units that it can trade with its neighbors: we denote by this capacity by c_i. We will assume that the capacities c_i are integers, that is, the capacities are discrete rather than continuous (this assumption is easily justifiable for the natural applications of stable assignment). The presence of node capacities allows us to assume, without loss of generality, that $|A| = |B| = n$, since dummy nodes with $c_i = 0$ can be added to the market to ensure that there is an equal number of nodes on both sides.

We use the term *assignment* as a generalization of matching to our many-to-many setting to mean a multi-unit pairing between the nodes in A and B. A *feasible assignment* $X = (x_{ij})_{(i,j)\in E}$, where $x_{ij} \geq 0$ is the number of units assigned between $i \in A$ and $j \in B$, satisfies capacity constraints on both sides, that is, $\sum_j x_{ij} \leq c_i$ and $\sum_i x_{ij} \leq c_j$. Note that both inequalities can be strict in a feasible assignment, that is, a node's capacity need not be exhausted completely. When all nodes have unit capacity, a feasible assignment is identical to a bipartite *matching*.

Preference Model. Each node $i \in A$ (respectively $j \in B$) has a preference list P_i ranking its neighbors $\{j \in B : (i, j) \in E\}$ (respectively $\{i \in A : (i, j) \in E\}$). The preference lists are allowed to have *ties*, *i.e.*, a node can be indifferent amongst any subset of its neighbors. Since a node's preference list is restricted to the set of its neighbors, the preference list is naturally incomplete. For example, a possible preference list for node i is $P_i = ([j_1, j_2], [j_3, j_5])$: that is, i is indifferent between j_1 and j_2, and prefers either of them to j_3, j_5, which i is indifferent amongst, and finds all other partners unacceptable.

Definition 2.1 (Level function). *We use the function $L_i(\cdot)$ to encode the preference list of a node $i \in A$ over individual nodes in B: for each $j \in P_i$, $L_i(j) \in \{1, \ldots, n\}$ gives the ranking of j in i's preference list. That is, for any $j, j' \in P_i$, if $L_i(j) < L_i(j')$, then i strictly* *prefers j to j'; if $L_i(j) = L_i(j')$, then i is indifferent between j and j'. (In the example above, $L_i(j_1) = L_i(j_2) = 1$ and $L_i(j_3) = L_i(j_4) = L_i(j_5) = 2$.)*

The definition of the level function $L_j(\cdot)$ for each $j \in B$ is symmetric.

The preferences of nodes over individual neighbors define a natural ranking over sets of neighbors, which we use to define the preference of a node over sets of neighbors: Given sets of neighbors S and S', arrange the nodes in S and S' in decreasing order of rank in i's preference list. i prefers S to S' if and only if $j_l \succeq j'_l$ for each l (using $\emptyset$ to make the sets equal-sized if one set has fewer neighbors than the other). Note that this is only a partial order, and specifically, some sets may not be comparable— for example, i cannot compare (or equivalently, is indifferent between) the sets $\{j_1, j_4\}$ and $\{j_2, j_3\}$, where j_l is at level l in i's preference list. This model of preferences for nodes with multi-unit capacity is both natural and has the advantage that nodes continue to only express preferences over individuals, and is exactly that used by Erdil and Ergin [8, 9] in their work on Pareto-stability for many-to-one matchings. We note that the choice of preference model over *sets* is relevant only to the Pareto optimality component of our discussion, and does *not* matter for stability, which is a pairwise solution concept and is not affected by preferences over sets.

3 Pareto-Stability

We first state the definition of stability for assignment; again, we use the term *stable assignment* to make the distinction with the unit-capacity setting, where an assignment reduces to a matching.

Definition 3.1 (Stable assignment)**.** *We say that an assignment $X = (x_{ij})$ is* stable *if there is no blocking pair (i, j), $i \in A$ and $j \in B$, $(i, j) \in E$, satisfying the one of the following conditions:*

- *Both i and j have leftover capacity;*
- *i has leftover capacity and there is i', $x_{i'j} > 0$, such that j strictly prefers i to i'; or j has capacity remaining and there is j', $x_{ij'} > 0$, such that i prefers j to j';*
- *There are i' and j', $x_{ij'} > 0$ and $x_{i'j} > 0$, such that i strictly prefers j to j' and j strictly prefers i to i'.*

Note that both members of a blocking pair must strictly prefer each other to their current partners. A stable assignment always exists, and can be found using a variant of Gale-Shapley algorithm [10] for computing stable matchings. We next define Pareto optimal assignments.

Definition 3.2 (Pareto-optimal assignment)**.** *Given assignment $X = (x_{ij})$, let $x_i(\alpha) = \sum_{j:\ L_i(j)\leq\alpha} x_{ij}$ be the total number of units of i's capacity that is assigned at levels better than or equal to level α, and $x_j(\beta) = \sum_{i:\ L_j(i)\leq\beta} x_{ij}$ be the total number of assigned units of j's capacity that are better than or equal to level β. We say that $X = (x_{ij})$ is* Pareto-optimal *if there is no other feasible assignment $Y = (y_{ij})$ such that $y_i(\alpha) \geq x_i(\alpha)$ and $y_j(\beta) \geq x_j(\beta)$, for all α, β, and at least one of the inequalities is strict.*

Recall from §1 that when preference lists contain ties, a stable matching need not be Pareto optimal. This leads naturally to the concept of Pareto stable matchings [29], which combines both Pareto-optimality and stability to provide a stronger solution concept to choose from amongst the set of stable matchings. (Note that the presence of ties in preference lists cannot be addressed by the standard trick of breaking ties using small perturbations:

if ties are broken arbitrarily, the set of stable matchings with respect to the new strict preferences can be strictly smaller than the set of stable matchings with respect to the original preferences with ties— that is, artificial tiebreaking does not preserve the set of stable matchings in the original problem.)

Definition 3.3. *A* Pareto-stable *assignment is a feasible assignment that is both stable and Pareto optimal.*

Augmenting Paths and Cycles. Given the connection between assignment and network flow, it is not surprising that the existence of augmenting paths and cycles in an assignment is closely related to whether it can be improved, *i.e.*, its Pareto optimality. The main difference in the context of stable matching is that nodes have preferences in addition to capacities: thus, augmenting paths and cycles must improve not just the size of an assignment, but also its quality, as determined by node preferences. We first define augmenting paths and cycles in the context of stable assignment.

Definition 3.4 (Augmenting Path)**.** *Given an assignment $X = (x_{ij})$, we say that $[i_0, j_1, i_1, \ldots, j_\ell, i_\ell, j_{\ell+1}]$ is an* augmenting path *if (i) $x_{i_0} < c_{i_0}$ and $x_{j_{\ell+1}} < c_{j_{\ell+1}}$, (ii) $x_{i_k j_k} > 0$ and $x_{i_{k-1} j_k} < c_{i_{k-1} j_k}$ for $k = 1, \ldots, \ell$, and (iii) $L_{i_k}(j_k) \geq L_{i_k}(j_{k+1})$ and $L_{j_k}(i_{k-1}) \leq L_{j_k}(i_k)$ for $k = 1, \ldots, \ell$.*

Definition 3.5 (Augmenting Cycle)**.** *Given an assignment $X = (x_{ij})$, we say that $[i_1, j_2, i_2, \ldots, j_\ell, i_\ell, j_1, i_1]$ is an* augmenting cycle *if (i) $x_{i_k j_k} > 0$ and $x_{i_{k-1} j_k} < c_{i_{k-1} j_k}$ for $k = 1, \ldots, \ell$, (ii) $L_{i_k}(j_k) \geq L_{i_k}(j_{k+1})$ and $L_{j_k}(i_{k-1}) \leq L_{j_k}(i_k)$ for $k = 1, \ldots, \ell$, where $i_0 = i_\ell$ and $j_{\ell+1} = j_1$, and (iii) at least one of the above inequalities is strict. If i_k is such a node (resp. j_k), we say it is an augmenting cycle associated with i_k (resp. j_k) at level $L_{i_k}(j_k)$ (resp. $L_{j_k}(i_k)$).*

The following easy lemma implies that if a stable assignment has no augmenting paths or cycles, then it must be Pareto stable (a similar result for Pareto-stable *matching* was shown in [29].)

Lemma 3.1. *Any assignment X that has no augmenting path or cycle is Pareto-optimal.*

4 Computing a Pareto-Stable Assignment

We now give a strongly polynomial time algorithm to compute a Pareto-stable assignment. Note that if X is a stable assignment, reassigning according to any augmenting path or cycle of X preserves stability, *i.e.*, any assignment Y that Pareto dominates a stable assignment X is stable as well [9]. This, together with Lemma 3.1, suggests that starting with a stable assignment, and then making improvements to it using augmenting paths and cycles until no more improvements are possible, will result in a Pareto stable assignment.

How do we find such augmenting paths and cycles? First consider the simplest case with unit capacity, *i.e.*, $c_i = c_j = 1$ for all i, j; here, an assignment degenerates to a matching. Given an existing matching, define a new directed bipartite graph with the same nodes, where all forward edges are *"weak improvement"* edges with respect to the existing matching, and backward edges correspond to the pairings in current matching. Then we are able to find augmenting paths by introducing a source and sink that link to unmatched nodes on each side. For cycles, since we need strict improvement for at least one node, we consider subgraphs, one for each node, which only consists of strict improvement edges for that node; then any cycle in the subgraph containing that node gives an augmenting cycle.

For our general case where $c_i \geq 1$, however, note that *even the concept of improvement edges for a node is not well defined* — since a node can have multiple partners in an assignment, a particular edge can present an improvement for some part of that node's capacity and not for some others. For instance, suppose that node i (with $c_i = 2$) is matched to nodes j_1 and j_3, and suppose that i strictly prefers j_1 to j_2 to j_3. Then, (i, j_2) would only represent an improvement relative to (i, j_3), but not with respect to (i, j_1), both of which exist in the current assignment.

An obvious way to fix this problem is to make copies of each node, one for each unit of its capacity, in which case improvement edges are well-defined– each unit of flow is associated with a unique neighbor in any assignment. However, note that this new graph has size $\sum_i c_i + \sum_j c_j$, consequently computing a Pareto-stable assignment in time polynomial in $\sum_i c_i + \sum_j c_j$, which, alas, is exponential in the size of the input.

4.1 Construction of Networks

In order to define improvement edges in this setting with multiple units of supply and demand, we will create a new augmented bipartite graph from the original bipartite graph and preference lists of nodes. The vertex set consists of copies of each node, where each copy represents a level on that node's preference list. We then define forward and backward edges between the vertices: forward edges are the (weak) improvement edges, while there is one backward edge for every feasible pair (i, j), $i \in A, j \in B$, corresponding to their respective levels in the others' preference list. This augmented graph, which is assignment-independent and depends only on the preference lists of the nodes, is then used to define a sequence of networks with assignment-*dependent* capacities, which allow us to find augmenting paths and cycles. The constructions are described formally below.

Definition 4.1. *Given the preference lists of nodes, we construct a directed graph G as follows.*

- *Vertices: For each node $i \in M$ (either in A or B), we introduce n new vertices $T(i) = \{i(1), \ldots, i(n)\}$, where $i(\alpha)$ corresponds the α-th level of the preference list of i. (If i has $k < n$ levels in his preference list, it suffices to introduce only k vertices $i(1), \ldots, i(k)$; here, we use n levels for uniformity.)*
- *Edges: For each pair $(i, j) \in E(M)$, let $\alpha = L_i(j)$ and $\beta = L_j(i)$. We add a* backward *edge between $i(\alpha)$ and $j(\beta)$, i.e. $j(\beta) \to i(\alpha)$. Further, we add a* forward *edge $i(\alpha') \to j(\beta')$ for every pair of vertices $i(\alpha')$ and $j(\beta')$ satisfying $\alpha' \geq \alpha$ and $\beta' \geq \beta$.*

The following figure shows an example of the construction, where the first figure gives the input instance (the number on each node is its supply/demand).

Definition 4.2 (Network H)**.** *Given graph G and an assignment X, we define network $H(X)$ as follows. We assign capacity infinity to all forward edges in G, and capacity x_{ij} to the backward edge between $T(i)$ and $T(j)$. We include a source s and a sink t; further, for each $i \in A$ and $j \in B$, we add an extra vertex h_i and h_j, respectively. We connect $s \to h_i$ with capacity $c_i - x_i$, and $h_j \to t$ with capacity $c_j - x_j$, where $x_i = \sum_j x_{ij}$ and $x_j = \sum_i x_{ij}$. Further, we connect $h_i \to i(\alpha)$ with capacity infinity for $\alpha = 1, \ldots, n$, and connect $j(\beta) \to h_j$ with capacity infinity for $\beta = 1, \ldots, n$.*

We will use the network H to find augmenting paths with respect to an existing stable assignment X. Observe that the only edges from the source with nonzero capacity are those that connect to a node $i \in A$ with leftover capacity; similarly, the only edges to the sink with nonzero capacity are from a node $j \in B$ with leftover capacity. Sending flow from s to t in

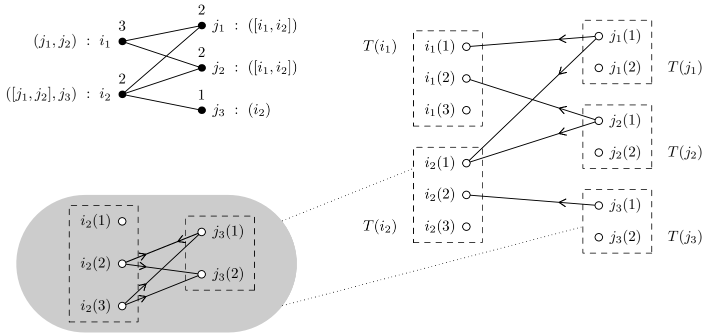


Figure 1: Construction of graph G.

H therefore involves increasing the total size of the assignment, exactly as in an augmenting path for X. In fact, as we will show in Proposition 4.1, after finding the maximum flow in H and updating the assignment accordingly, there are no remaining augmenting paths in the new assignment.

Definition 4.3 (Networks $H_{i,\alpha}$ and $H_{j,\beta}$)**.** *Given the graph G and X, let $G(X)$ be the network where all forward edges in G are assigned capacity infinity, and all backward edges are assigned capacity x_{ij}. We use $G(X)$ define the networks $H_{i,\alpha}(X)$ and $H_{j,\beta}(X)$ for each $i \in A$ and $j \in B$, and $\alpha, \beta = 1, \ldots, n$, as follows.*

To get network $H_{i,\alpha}(X)$ from $G(X)$, we add a source s and a sink t, and connect $s \to j(\beta)$ with capacity infinity for each vertex $j(\beta)$ satisfying $\alpha > L_i(j)$ and $\beta \geq L_j(i)$ (an equivalent definition is that we connect $s \to j(\beta)$ if there is an edge $i(\alpha) \to j(\beta)$ and $\alpha > L_i(j)$). Further, we connect $j(\beta) \to t$ with capacity x_{ij} for each $j(\beta)$ satisfying $\alpha \leq L_i(j)$ and $\beta = L_j(i)$.

The network $H_{j,\beta}(X)$ is defined symmetrically. That is, we include a source s and a sink t, and connect $s \to i(\alpha)$ with capacity x_{ij} for each vertex $i(\alpha)$ satisfying $\alpha = L_i(j)$ and $\beta \leq L_j(i)$. Further, we connect $i(\alpha) \to t$ with capacity infinity for each $i(\alpha)$ satisfying $\alpha \geq L_i(j)$ and $\beta > L_j(i)$.

We will use the networks $H_{i,\alpha}$ and $H_{j,\beta}$ to find augmenting cycles associated with i and j at level α and β, respectively. Consider any flow from s to t in $H_{i,\alpha}$, say $[s, j_1(\beta_1), i_1(\alpha_1), \ldots, i_2(\alpha_2), j_2(\beta_2), t]$, we know that $\alpha > L_i(j_1)$ (i.e. i strictly prefers j_1 to all its neighbors at level α) and $L_{j_1}(i_1) = \beta_1 \geq L_{j_1}(i)$ (i.e. j_1 weakly prefers i to i_1). Further, we have $\alpha \leq L_i(j_2)$ (this implies that i strictly prefers j_1 to j_2) and $L_{j_2}(i_2) \leq \beta_2 = L_{j_2}(i)$ (i.e. j_2 weakly prefers i_2 to i). That is, flows from s to t in $H_{i,\alpha}$ correspond to augmenting cycles for node i at levels less than or equal to α in X (a symmetric argument holds for graph $H_{j,\beta}$). We will show in Proposition 4.2 that once we compute the maximum flow in $H_{i,\alpha}$, there are no remaining augmenting cycles for node i at level α (note, not level α or below).

4.2 Algorithm

PARETO STABLE ASSIGNMENT (PARETO-ASSIGNMENT)

1. Let X be an arbitrary stable assignment
2. Construct networks H, $H_{i,\alpha}$ and $H_{j,\beta}$, for each $i \in A$, $j \in B$, and $\alpha, \beta = 1, \dots, n$ given X
3. For H, $H_{i,\alpha}$ and $H_{j,\beta}$ constructed above (H to be executed first)
   (a) Compute a maximum flow $F = (f_{uv})$ from s to t (if there is no flow from vertex u to v, set $f_{uv} = 0$)
   (b) For each forward edge $i(\alpha) \to j(\beta)$, let $x_{ij} = x_{ij} + f_{i(\alpha)j(\beta)}$
   (c) For each backward edge $j(\beta) \to i(\alpha)$, let $x_{ij} = x_{ij} - f_{j(\beta)i(\alpha)}$
   (d) If the graph is $H_{i,\alpha}$
      - Let $x_{ij} = x_{ij} + f_{sj(\beta)}$ for each edge $s \to j(\beta)$
      - Let $x_{ij} = x_{ij} - f_{j(\beta)t}$ for each edge $j(\beta) \to t$
   (e) If the graph is $H_{j,\beta}$
      - Let $x_{ij} = x_{ij} - f_{sj(\beta)}$ for each edge $s \to i(\alpha)$
      - Let $x_{ij} = x_{ij} + f_{j(\beta)t}$ for each edge $i(\alpha) \to t$
   (f) Update the capacities for the next graph to be executed according to the new assignment X
4. Output X (denoted by X^*)

To prove that PARETO-ASSIGNMENT indeed computes a Pareto-stable assignment, we need to show two main things — first, that the resulting assignment is feasible, stable, and all nodes' assignments are weakly enhanced through the course of the algorithm.

Second, we need to show that no further Pareto improvements are possible when the algorithm terminates, *i.e.*, X^* is Pareto optimal. Note that the assignment X changes through the course of the algorithm, and therefore we need to show that, for instance, no other augmenting paths can be found after the network H has been executed, *even though* the assignment X that was used to define the network H has been changed (and similarly for all augmenting cycles). That is, while we compute maximum flows in $H(X)$ to find all augmenting paths for a given assignment X, we need to show that no new augmenting paths have showed up in the updated assignments X' computed by the algorithm. Similarly, finding (i, α) augmenting cycles via $H_{i,\alpha}(X)$ for *some* assignment X does not automatically imply that no further (i, α) augmenting cycles will ever be found in *any* of the (different) assignments X' computed through the course of the algorithm, since the assignments of all nodes can change each time when a maximum flow is computed, leading to the possibility of new valid s-t paths, and therefore possibly new augmenting cycles. Note that this is hardly obvious, and in fact, as Example 4.1 demonstrates, that this does not happen is due to a careful choice of the construction of the networks $H_{i,\alpha}, H_{j,\beta}$.

Example 4.1. *Suppose there are four nodes i_1, i_2, i_3, k in A and five nodes j_1, j_2, j_3, j_4, j_5 in B. All nodes except k have unit capacity and are indifferent between all possible partners (i.e., have only one level in their preference list). Node k has capacity 2, and preference list $([j_1, j_5], [j_3, j_4], j_2)$. Suppose we start with the (stable) assignment X_0 where k is matched to j_2, j_3, and the remaining assignments are $(i_1, j_1), (i_2, j_4), (i_3, j_5)$ (note there are no augmenting paths in X_0). Consider finding the maximum flow in network $H_{i,\alpha}$* without *the link $j_2 \to t$ for $\alpha = 2$. In this network, the total capacity of edges incident to the sink is 1, thus we can send at most one unit flow, for example $k \to j_1 \to i_1 \to j_2 \to k \to j_4 \to i_2 \to j_3 \to t$*

(note that the two k's here correspond to different vertices in $T(k)$ in the network). After reassigning assignment according to this flow, we obtain the new assignment X' $(i_1, j_2), (i_2, j_3), (i_3, j_5), (k, j_1), (k, j_4)$. But observe that X' still has an augmenting cycle at level 2 for node k: $k \to j_5 \to i_3 \to j_4 \to k$. However, with the original definition of $H_{i,\alpha}$, which links $j_2 \to t$, the maximum flow consists of pushing flow along the paths $k \to j_1 \to i_1 \to j_2 \to t$ and $k \to j_5 \to i_3 \to j_4 \to i_2 \to j_3 \to t$, leading to the new assignment $X'' = (i_1, j_2), (i_2, j_3), (i_3, j_4), (k, j_1), (k, j_5)$ which has no remaining augmenting cycles for k.

The Pareto-optimality of the assignment X^* returned by the algorithm follows from the following two claims.

Proposition 4.1. *There is no augmenting path after graph H is executed in Step 3 of* PARETO-ASSIGNMENT*.*

Proposition 4.2. *There is no augmenting cycle associated with i (resp. j) at level α (resp. β) after graph $H_{i,\alpha}$ (resp. $H_{j,\beta}$) is executed in step 3 of* PARETO-ASSIGNMENT*.*

Together, these two propositions imply that the outcome returned by PARETO-ASSIGNMENT is indeed a Pareto-optimal assignment. Note that the construction of each graph $H, H_{i,\alpha}$ and $H_{j,\beta}$ is in polynomial time. In total there are $O(n^2)$ such graphs with $O(n^2)$ vertices each. For each graph $H, H_{i,\alpha}$ and $H_{j,\beta}$, its maximum flow can be computed in strongly polynomial time $O(n^6)$ with respect to its number of vertices $O(n^2)$ [18]. Therefore, the running time of the algorithm is in $O(n^8)$. This gives us our main result:

Theorem 4.1. *Algorithm* PARETO-ASSIGNMENT *computes a Pareto-stable assignment in strongly polynomial time $O(n^8)$, where n is the total number of nodes in the bipartite graph M.*

5 Remarks

In one-to-one matching, the solution concepts of pairwise stability, core, and setwise stability all coincide, but this is not the case with many-to-many matching [27]. For our preference model for many-to-many matching, the core is not a suitable solution concept, since matchings in the core need not be pairwise stable ([28], Fig.1a), and the strong core need not exist ([27], §5.7) (it is easy to adapt the examples in these references to our model of preferences over sets). We also note that Pareto stability is incomparable with set-wise stability (both of which are strictly stronger than pairwise stability) in the sense that neither solution concept is stronger than the other— an easy example shows set-wise stable matchings need not be Pareto-optimal, and vice versa. The problem of computing set-wise, rather than pairwise, stable matchings, appears to be a challenging algorithmic question, and we leave it as an open problem for future work.

Acknowledgments. We are very grateful to Fuhito Kojima for helpful discussions and pointers to relevant literature. We also thank Mohammad Mahdian, David Pennock and Michael Schwarz for useful comments.

References

[1] A. Abdulkadiroglu, P. Pathak, A. E. Roth, *Strategy-proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match*, American Economic Review, V.99(5), 1954-1978, 2009.

[2] A. Abdulkadiroglu, T. Sonmez. *School Choice: A Mechanism Design Approach*, American Economic Review, V.93(3), 729-747, 2003.

[3] D. Abraham, *Algorithmics of Two-Sided Matching Problems*, M.S. Thesis, University of Glasgow, 2003.

[4] N. Chen, A. Ghosh, N. Lambert, *Social Lending*, EC 2009, 335-344.

[5] J. Eeckhout, *On the Uniqueness of Stable Marriage Matchings*, Economics Letters, V.69, 1-8, 2000.

[6] F. Echenique, J. Oviedo, *A Theory of Stability in Many-to-many Matching Markets*, Theoretical Economics. Volume 1, Issue 2, June 2006, Pages 233-273.

[7] A. Erdil, *Two-sided Matching with Ties*, Ph.D. Dissertation, University of Chicago, 2006.

[8] A. Erdil, H. Ergin, *What's the Matter with Tie-breaking? Improving Efficiency in School Choice*, American Economic Review, V.98(3), 669-689, 2008.

[9] A. Erdil, H. Ergin, *Two-Sided Matching with Indifferences*, working paper.

[10] D. Gale, L. S. Shapley, *College Admissions and the Stability of Marriage*, American Mathematical Monthly, V.69, 9-15, 1962.

[11] D. Gusfield, R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, 1989.

[12] J. Hatfield, S. Kominers, *Matching in Networks with Bilateral Contracts*, Proceedings of the 11th ACM Conference on Electronic Commerce, 2010.

[13] S. Herrero-Lopez, *Social Interactions in P2P Lending*, SNA-KDD Workshop on Social Network Mining and Analysis, 2009.

[14] K. Iwama, S. Miyazaki, *Stable Marriage with Ties and Incomplete Lists*, Encyclopedia of Algorithms, 2008.

[15] R. W. Irving, *Stable Marriage and Indifference*, Discrete Applied Mathematics, V.48, 261-272, 1994.

[16] K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, *Stable Marriage with Incomplete Lists and Ties*, ICALP 1999, 443-452.

[17] T. Kavitha, K. Mehlhorn, D. Michail, K. E. Paluch, *Strongly Stable Matchings in Time O(nm) and Extension to the Hospitals-Residents Problem*, ACM Transactions on Algorithms, V.3(2), 2007.

[18] J. Kleinberg, E. Tardos, *Algorithm Design*, Addison Wesley, 2005.

[19] V. S. Malhotra, *On the Stability of Multiple Partner Stable Marriages with Ties*, ESA 2004, 508-519.

[20] D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, Y. Morita, *Hard Variants of Stable Marriage*, Theoretical Computer Science, V.276(1-2), 261-279, 2002.

[21] M. Niederle, L. Yariv, *Decentralized Matching with Aligned Preferences*, working paper.

[22] A. Ockenfels, A. E. Roth, *Late and Multiple Bidding in Second-Price Internet Auctions: Theory and Evidence Concerning Different Rules for Ending an Auction*, Games and Economic Behavior, V.55, 297-320, 2006.

[23] A. E. Roth, *The Evolution of the Labor Market for Medical Interns and Residents: a Case Study in Game Theory*, Journal of Political Economy, V.92(6), 991-1016, 1984.

[24] A. E. Roth, *The Economist as Engineer: Game Theory, Experimentation, and Computation as Tools for Design Economics*, Econometrica, V.70(4), 1341-1378, 2002.

[25] A. E. Roth, *Deferred Acceptance Algorithms: History, Theory, Practice, and Open Questions*, International Journal of Game Theory, 537-569, 2008.

[26] A. E. Roth, A. Ockenfels, *Last-Minute Bidding and the Rules for Ending Second-Price Auctions: Evidence from eBay and Amazon Auctions on the Internet*, American Economic Review, V.92(4), 1093-1103, 2002.

[27] A. E. Roth, M. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, Cambridge University Press, 1992.

[28] M. Sotomayor, *Three Remarks on the Many-to-Many Stable Matching Problem*, Mathematical Social Sciences, V.38(1), 55-70, 1999.

[29] M. Sotomayor, *The Pareto-Stability Concept is a Natural Solution Concept*, working paper.

[30] H. Yanagisawa, *Approximation Algorithms for Stable Marriage Problems*, Ph.D. Thesis, Kyoto University, 2007.

Ning Chen
Division of Mathematical Sciences,
Nanyang Technological University,
Singapore.
Email: `ningc@ntu.edu.sg`

Arpita Ghosh
Yahoo! Research,
Santa Clara, CA, USA.
Email: `arpita@yahoo-inc.com`

Stable Marriage Problems with Quantitative Preferences

Maria Silvia Pini, Francesca Rossi, K. Brent Venable, and Toby Walsh

Abstract

The stable marriage problem is a well-known problem of matching men to women so that no man and woman, who are not married to each other, both prefer each other. Such a problem has a wide variety of practical applications, ranging from matching resident doctors to hospitals, to matching students to schools or more generally to any two-sided market. In the classical stable marriage problem, both men and women express a strict preference order over the members of the other sex, in a qualitative way. Here we consider stable marriage problems with quantitative preferences: each man (resp., woman) provides a score for each woman (resp., man). Such problems are more expressive than the classical stable marriage problems. Moreover, in some real-life situations it is more natural to express scores (to model, for example, profits or costs) rather than a qualitative preference ordering. In this context, we define new notions of stability and optimality, and we provide algorithms to find marriages which are stable and/or optimal according to these notions. While expressivity greatly increases by adopting quantitative preferences, we show that in most cases the desired solutions can be found by adapting existing algorithms for the classical stable marriage problem.

1 Introduction

The stable marriage problem (SM) [5] is a well-known problem of matching the elements of two sets. It is called the *stable marriage* problem since the standard formulation is in terms of men and women, and the matching is interpreted in terms of a set of marriages. Given n men and n women, where each person expresses a strict ordering over the members of the opposite sex, the problem is to match the men to the women so that there are no two people of opposite sex who would both rather be matched with each other than their current partners. If there are no such people, all the marriages are said to be *stable*. In [4] Gale and Shapley proved that it is always possible to find a matching that makes all marriages stable, and provided a polynomial time algorithm which can be used to find one of two extreme stable marriages, the so-called *male-optimal* or *female-optimal* solutions. The Gale-Shapley algorithm has been used in many real-life scenarios, such as in matching hospitals to resident doctors [12], medical students to hospitals, sailors to ships [8], primary school students to secondary schools [13], as well as in market trading [14].

In the classical stable marriage problem, both men and women express a strict preference order over the members of the other sex in a qualitative way. Here we consider stable marriage problems with quantitative preferences. In such problems each man (resp., woman) provides a score for each woman (resp., man). Stable marriage problems with quantitative preferences are interesting since they are more expressive than the classical stable marriage problems, since in classical stable marriage problem a man (resp., a woman) cannot express how much he (resp., she) prefers a certain woman (resp., man). Moreover, they are useful in some real-life situations where it is more natural to express scores, that can model notions such as profit or cost, rather than a qualitative preference ordering. In this context, we define new notions of stability and optimality, we compare such notions with the classical ones, and we show algorithms to find marriages which are stable and/or optimal according to these notions. While expressivity increases by adopting quantitative preferences, we show that in most cases the desired solutions can be found by adapting existing algorithms for the classical stable marriage problem.

Stable marriage problems with quantitative preferences have been studied also in [6, 7]. However, they solve these problems by looking at the stable marriages that maximize the sum of the weights of the married pairs, where the weights depend on the specific criteria used to find an optimal solution, that can be minimum regret criterion [6], the egalitarian criterion [7] or the Lex criteria [7]. Therefore, they consider as stable the same marriages that are stable when we don't consider the weights. We instead use the weights to define new notions of stability that may lead to stable marriages that are different from the classical case. They may rely on the difference of weights that a person gives to two different people of the other sex, or by the strength of the link of the pairs (man,woman), i.e., how much a person of the pair wants to be married with the other person of the pair. The classical definition of stability for stable marriage problems with quantitative preferences has been considered also in [2] that has used a semiring-based soft constraint approach [3] to model and solve these problems.

The paper is organized as follows. In Section 2 we give the basic notions of classical stable marriage problems, stable marriage problems with partially ordered preferences and stable marriage problems with quantitative preferences (SMQs). In Section 3 we introduce a new notion of stability, called α-stability for SMQs, which depends on the difference of scores that every person gives to two different people of the other sex, and we compare it with the classical notion of stability. Moreover, we give a new notion of optimality, called lex-optimality, to discriminate among the new stable marriages, which depends on a voting rule. We show that there is a unique optimal stable marriage and we give an algorithm to find it. In Section 4 we introduce other notions of stability for SMQs that are based on the strength of the link of the pairs (man,woman), we compare them with the classical stability notion, and we show how to find marriages that are stable according to these notions with the highest global link. In Section 5 we summarize the results contained in this paper, and we give some hints for future work.

2 Background

We now give some basic notions on classical stable marriage problems, stable marriage problems with partial orders, and stable marriage problems with quantitative preferences.

2.1 Stable marriage problems

A *stable marriage problem* (SM) [5] of size n is the problem of finding a stable marriage between n men and n women. Such men and women each have a preference ordering over the members of the other sex. A marriage is a one-to-one correspondence between men and women. Given a marriage M, a man m, and a woman w, the pair (m, w) is a *blocking pair* for M if m prefers w to his partner in M and w prefers m to her partner in M. A marriage is said to be *stable* if it does not contain blocking pairs.

The sequence of all preference orderings of men and women is usually called a *profile*. In the case of classical stable marriage problem (SM), a profile is a sequence of strict total orders.

Given a SM P, there may be many stable marriages for P. However, it is interesting to know that there is always at least one stable marriage.

Given an SM P, a *feasible partner* for a man m (resp., a woman w) is a woman w (resp., a man m) such that there is a stable marriage for P where m and w are married.

The set of all stable marriages for an SM forms a lattice, where a stable marriage M_1 dominates another stable marriage M_2 if men are happier (that is, are married to more or equally preferred women) in M_1 w.r.t. M_2. The top of this lattice is the stable marriage where men are most satisfied, and it is usually called the *male-optimal* stable marriage. Conversely, the bottom is the stable marriage where men's preferences are least satisfied (and women are happiest, so it is usually called the *female-optimal* stable marriage). Thus, a stable marriage is male-optimal iff every man is paired with his highest ranked feasible partner.

The *Gale-Shapley* (GS) *algorithm* [4] is a well-known algorithm to solve the SM problem. At the start of the algorithm, each person is free and becomes engaged during the execution of the algorithm. Once a woman is engaged, she never becomes free again (although to whom she is engaged may change), but men can alternate between being free and being engaged. The following step is iterated until all men are engaged: choose a free man m, and let m propose to the most preferred woman w on his preference list, such that w has not already rejected m. If w is free, then w and m become engaged. If w is engaged to man m', then she rejects the man (m or m') that she least prefers, and becomes, or remains, engaged to the other man. The rejected man becomes, or remains, free. When all men are engaged, the engaged pairs form the male optimal stable matching. It is female optimal, of course, if the roles of male and female participants in the algorithm were interchanged.

This algorithm needs a number of steps that, in the worst case, is quadratic in n (that is, the number of men), and it guarantees that, if the number of men and women coincide, and all participants express a strict order over all the members of the other group, everyone gets married, and the returned matching is stable.

Example 1 *Assume $n = 2$. Let $\{w_1, w_2\}$ and $\{m_1, m_2\}$ be respectively the set of women and men. The following sequence of strict total orders defines a profile:*

- $m_1 : w_1 > w_2$ *(i.e., man m_1 prefers woman w_1 to woman w_2),*
- $m_2 : w_1 > w_2$*,*
- $w_1 : m_2 > m_1$*,*
- $w_2 : m_1 > m_2$*.*

For this profile, the male-optimal solution is $\{(m_1, w_2), (m_2, w_1)\}$. For this specific profile the female-optimal stable marriage coincides with the male-optimal one. □

2.2 Stable marriage problems with partially ordered preferences

In SMs, each preference ordering is a strict total order over the members of the other sex. More general notions of SMs allow preference orderings to be partial [9]. This allows for the modelling of both indifference (via ties) and incomparability (via absence of ordering) between members of the other sex. In this context, a stable marriage problem is defined by a sequence of $2n$ partial orders, n over the men and n over the women. We will denote with SMP a stable marriage problem with such partially ordered preferences.

Given an SMP, we will sometimes use the notion of a *linearization* of such a problem, which is obtained by linearizing the preference orderings of the profile in a way that is compatible with the given partial orders.

A marriage M for an SMP is said to be *weakly-stable* if it does not contain blocking pairs. Given a man m and a woman w, the pair (m, w) is a blocking pair if m and w are not married to each other in M and each one *strictly prefers* the other to his/her current partner.

A weakly stable marriage M dominates a weakly stable marriage M' iff for every man m, $M(m) \geq M'(m)$ and there is a man m' s.t. $M(m') > M'(m')$. Notice that there may be more than one undominated weakly stable marriage for an SMP.

2.3 Stable marriage problems with quantitative preferences

In classical stable marriage problems, men and women express only qualitative preferences over the members of the other sex. For every pair of women (resp., men), every man (resp., woman) states only that he (resp., she) prefers a woman (resp., a man) more than another one. However, he (resp., she) cannot express how much he (resp., she) prefers such a woman (resp., a man). This is nonetheless possible in stable marriage problems with quantitative preferences.

A *stable marriage problem with quantitative preferences* (SMQ) [7] is a classical SM where every man/woman gives also a numerical preference value for every member of the other sex, that represents how much he/she prefers such a person. Such preference values are natural numbers and higher preference values denote a more preferred item. Given a man m and a woman w, the *preference value* for man m (resp., woman w) of woman w (resp., man m) will be denoted by $p(m, w)$ (resp., $p(w, m)$).

Example 2 *Let $\{w_1, w_2\}$ and $\{m_1, m_2\}$ be respectively the set of women and men. An instance of an SMQ is the following:*

- $m_1 : w_1^{[9]} > w_2^{[1]}$ *(i.e., man m_1 prefers woman w_1 to woman w_2, and he prefers w_1 with value 9 and w_2 with value 1),*
- $m_2 : w_1^{[3]} > w_2^{[2]}$*,*
- $w_1 : m_2^{[2]} > m_1^{[1]}$*,*
- $w_2 : m_1^{[3]} > m_2^{[1]}$*.*

The numbers written into the round brackets identify the preference values. □

In [7] they consider stable marriage problems with quantitative preferences by looking at the stable marriage that maximizes the sum of the preference values. Therefore, they use the classical definition of stability and they use preference values only when they have to look for the optimal solution. We want, instead, to use preference values also to define new notions of stability and optimality.

We will introduce new notions of stability and optimality that are based on the quantitative preferences expressed by the agents and we will show how to find them by adapting the classical Gale-Shapley algorithm [4] for SMs described in Section 2.

3 α-stability

A simple generalization of the classical notion of stability requires that there are not two people that prefer *with at least degree* α (where α is a natural number) to be married to each other rather than to their current partners.

Definition 1 (α-stability) *Let us consider a natural number α with $\alpha \geq 1$. Given a marriage M, a man m, and a woman w, the pair (m, w) is an α-blocking pair for M if the following conditions hold:*

- *m prefers w to his partner in M, say w', by at least α (i.e., $p(m, w) - p(m, w') \geq \alpha$),*
- *w prefers m to her partner in M, say m', by at least α (i.e., $p(w, m) - p(w, m') \geq \alpha$).*

A marriage is α-stable if it does not contain α-blocking pairs. A man m (resp., woman w) is α-feasible for woman w (resp., man m) if m is married with w in some α-stable marriage.

3.1 Relations with classical stability notions

Given an SMQ P, let us denote with $c(P)$, the classical SM problem obtained from P by considering only the preference orderings induced by the preference values of P.

Example 3 *Let us consider the SMQ, P, shown in Example 2. The stable marriage problem $c(P)$ is shown in Example 1.* $\Box$.

If α is equal to 1, then the α-stable marriages of P coincide with the stable marriages of $c(P)$. However, in general, α-stability allows us to have more marriages that are stable according to this definition, since we have a more relaxed notion of blocking pair. In fact, a pair (m, w) is an α-blocking if both m and w prefer each other to their current partner *by at least* α and thus pairs (m', w') where m' and w' prefer each other to their current partner of less than α are not considered α-blocking pairs.

The fact that α-stability leads to a larger number of stable marriages w.r.t. the classical case is important to allow new stable marriages where some men, for example the most popular ones, may be married with partners better than all the feasible ones according to the classical notion of stability.

Given an SMQ P, let us denote with $I_\alpha(P)$ the set of the α-stable marriages of P and with $I(c(P))$ the set of the stable marriages of $c(P)$. We have the following results.

Proposition 1 *Given an SMQ P, and a natural number α with $\alpha \geq 1$,*

- *if $\alpha = 1$, $I_\alpha(P) = I(c(P))$;*
- *if $\alpha > 1$, $I_\alpha(P) \supseteq I(c(P))$.*

Given an SMP P, the set of α-stable marriages of P contains the set of stable marriages of $c(P)$, since the α-blocking pairs of P are a subset of the blocking pairs of $c(P)$.

Let us denote with $\alpha(P)$ the stable marriage with incomparable pairs obtained from an SMQ P by setting as incomparable every pair of people that don't differ for at least α, and with $I_w(\alpha(P))$ the set of the weakly stable marriages of $\alpha(P)$. It is possible to show that the set of the weakly stable marriages of $\alpha(P)$ coincides with the set of the α-stable marriages of P.

Theorem 1 *Given an SMQ P, $I_\alpha(P) = I_w(\alpha(P))$.*

Proof: We first show that $I_\alpha(P) \subseteq I_w(\alpha(P))$. Assume that a marriage $M \notin I_w(\alpha(P))$, we now show that $M \notin I_\alpha(P)$. If $M \notin I_w(\alpha(P))$, then there is a pair (man,woman), say (m, w), in $\alpha(P)$ such that m prefers w to his partner in M, say w', and w prefers m to her partner in M, say m'. By definition of $\alpha(P)$, this means that m prefers w to w' by at least degree α and w prefers m to m' by at least degree α in P, and so $M \notin I_\alpha(P)$. Similarly, we can show that $I_\alpha(P) \supseteq I_w(\alpha(P))$. In fact, if $M \notin I_\alpha(P)$, then there is a pair (man,woman), say (m, w), in P such that m prefers w to w' by at least degree α and w prefers m to m' by at least degree α. By definition of $\alpha(P)$, this means that m prefers w to w' and w prefers m to m' in $\alpha(P)$ and so $M \notin I_w(\alpha(P))$, i.e., M is not a weakly stable marriage for $\alpha(P)$. $\Box$

This means that, given an SMQ P, every algorithm that is able to find a weakly stable marriage for $\alpha(P)$ provides an α-stable marriage for P.

Example 4 *Assume that α is 2. Let us consider the following instance of an SMQ, say P.*

- $m_1 : w_1^{[3]} > w_2^{[2]}$
- $m_2 : w_1^{[4]} > w_2^{[2]}$,

- $w_1 : m_1^{[8]} > m_2^{[5]}$,

- $w_2 : m_1^{[3]} > m_2^{[1]}$.

The SMP $\alpha(P)$ is the following:

- $m_1 : w_1 \bowtie w_2$ *(where $\bowtie$ means incomparable),*

- $m_2 : w_1 > w_2$,

- $w_1 : m_1 > m_2$,

- $w_2 : m_1 > m_2$.

The set of the α-stable marriages of P, that coincides with the set of the weakly stable marriages of $\alpha(P)$, by Theorem 1, contains the following marriages: $M_1 = \{(m_1, w_1), (m_2, w_2)\}$ and $M_2 = \{(m_1, w_2), (m_2, w_1)\}$. □

On the other hand, not all stable marriage problems with partially ordered preferences can be expressed as stable marriage problems with quantitative preferences such that the stable marriages in the two problems coincide. More precisely, given any SMP problem P, we would like to be able to generate a corresponding SMQ problem P' and a value α such that, in P', the weights of elements ordered in P differ more than α, while those of elements that are incomparable in P differ less than α. Consider for example the case of a partial order over six elements, defined as follows: $x_1 > x_2 > x_3 > x_4 > x_5$ and $x_1 > y > x_5$. Then there is no way to choose a value α and a linearization of the partial order such that the weights of x_i and x_j differ for at least α, for any i,j between 1 and 5, while at the same time the weight of y and each of the x_i's differ for less than α.

3.2 Dominance and lex-male-optimality

We recall that in SMPs a weakly-stable marriage dominates another weakly-stable marriage if men are happier (or equally happy) and there is at least a man that is strictly happier. The same holds for α-stable marriages. As in SMPs there may be more than one undominated weakly-stable marriage, in SMQs there may be more than one undominated α-stable marriage.

Definition 2 (dominance) *Given two α-stable marriages, say M and M', M dominates M' if every man is married in M to more or equally preferred woman than in M' and there is at least one man in M married to a more preferred woman than in M'.*

Example 5 *Let us consider the SMQ shown in Example 4. We recall that α is 2 and that the α-stable marriages of this problem are $M_1 = \{(m_1, w_1), (m_2, w_2)\}$ and $M_2 = \{(m_1, w_2), (m_2, w_1)\}$. M_2 does not dominate M_1 since, for m_1, $M_1(m_1) > M_2(m_1)$ and M_1 does not dominate M_2 since, for m_2, $M_2(m_2) > M_1(m_2)$.* □

We now discriminate among the α-stable marriages of an SMQ, by considering the preference values given by women and men to order pairs that differ for less than α.

We will consider a marriage optimal when the most popular men are as happy as possible and they are married with the most popular α-feasible women.

To compute a strict ordering on the men where the most popular men (resp., the most popular women) are ranked first, we follow a reasoning similar to the one considered in [11, 10], that is, we apply a voting rule [1] to the preferences given by the women (resp., by the men) . More precisely, such a voting rule takes in input the preference values given by the women over the men (resp., given by the men over the women) and returns a strict total order over the men (resp., women).

Definition 3 (lex-male-optimal) *Consider an SMQ P, a natural number α, and a voting rule r. Let us denote with o_m (resp., o_w) the strict total order over the men (resp., over the women) computed by applying r to the preference values that the women give to the men (resp., the men give to the women). An α-stable marriage M is* lex-male-optimal *w.r.t. o_m and o_w, if, for every other α-stable marriage M', the following conditions hold:*

- *there is a man m_i such that $M(m_i) \succ_{o_w} M'(m_i)$,*
- *for every man $m_j \prec_{o_m} m_i$, $M(m_j) = M'(m_j)$.*

Proposition 2 *Given an SMQ P, a strict total ordering o_m (resp., o_w) over the men (resp., women),*

- *there is a unique lex-male-optimal α-stable marriage w.r.t. o_m and o_w, say L.*
- *L may be different from the male-optimal stable marriage of $c(P)$;*
- *if $\alpha(P)$ has a unique undominated weakly stable marriage, say L', then L coincides with L', otherwise L is one of the undominated weakly stable marriages of $\alpha(P)$.*

Example 6 *Let us consider the SMQ, P, shown in Example 4. We have shown previously that this problem has two α-weakly stable marriages that are undominated. We now want to discriminate among them by considering the lex-male-optimality notion. Let us consider as voting rule the rule that takes in input the preference values given by the women over the men (resp., by the men over the women) and returns a strict preference ordering over the men (resp., women). This preference ordering is induced by the overall score that each man (resp., woman) receives: men (women) that receive higher overall scores are more preferred. The overall score of a man m (resp., woman w), say $s(m)$ (resp., $s(w)$), is computed by summing all the preference values that the women give to him (the men give to her). If two candidates receive the same overall score, we use a tie-breaking rule to order them. If we apply this voting rule to the preference values given by the women in P, then we obtain $s(m_1) = 8 + 3 = 11$, $s(m_2) = 5 + 1 = 6$, and thus the ordering o_m is such that $m_1 \succ_{o_m} m_2$. If we apply the same voting rule to the preference values given by the men in P, $s(w_1) = 3 + 4 = 7$, $s(w_2) = 2 + 2 = 4$, and thus the ordering o_w is such that $w_1 \succ_{o_w} w_2$. The lex-male-optimal α-stable marriage w.r.t. o_m and o_w is the marriage $M_1 = \{(m_1, w_1), (m_2, w_2)\}$.* □

3.3 Finding the lex-male-optimal α-stable marriage

It is possible to find optimal α-stable marriages by adapting the GS-algorithm for classical stable marriage problems [4].

Given an SMQ P and a natural number α, by Theorem 1, to find an α-stable marriage it is sufficient to find a weakly stable marriage of $\alpha(P)$. This can be done by applying the GS algorithm to any linearization of $\alpha(P)$.

Given an SMQ P, a natural number α, and two orderings o_m and o_w over men and women computed by applying a voting rule to P as described in Definition 3, it is possible to find the α-stable marriage that is lex-male-optimal w.r.t o_m and o_w by applying the GS algorithm to the linearization of $\alpha(P)$ where we order incomparable pairs, i.e., the pairs that differ for less than α in P, in accordance with the orderings o_m and o_w.

Proposition 3 *Given an SMQ P, a natural number α, o_m (resp., o_w) an ordering over the men (resp., women), algorithm* Lex-male-α-stable-GS *returns the lex-male-optimal α-stable marriage of P w.r.t. o_m and o_w.*

Algorithm 1: *Lex-male-α-stable-GS*

Input: P: an SMQ, α: a natural number, r: a voting rule
Output: μ: a marriage
$o_m \leftarrow$ the strict total order over the men obtained by applying r to the preference values given by the women over the men
$o_w \leftarrow$: the strict total order over the women obtained by applying r to the preference values given by the men over the women
$P' \leftarrow$ the linearization of $\alpha(P)$ obtained by ordering incomparable pairs of $\alpha(P)$ in accordance with o_m and o_w;
$\mu \leftarrow$ the marriage obtained by applying the GS algorithm to P';
return μ

4 Stability notions relying on links

Until now we have generalized the classical notion of stability by considering separately the preferences of the men and the preferences of the women. We now intend to define new notions of stability that take into account simultaneously the preferences of the men and the women. Such a new notion will depend on the strength of the link of the married people, i.e., how much a man and a woman want to be married with each other. This is useful to obtain a new notion of stable marriage, that looks at the happiness of the pairs (man,woman) rather than at the happiness of the members of a single sex.

A way to define the strength of the link of two people is the following.

Definition 4 (link additive-strength) *Given a man m and a woman w, the* link additive-strength *of the pair (m, w), denoted by $la(m, w)$, is the value obtained by summing the preference value that m gives to w and the preference value that w gives to m, i.e., $la(m, w) = p(m, w) + p(w, m)$. Given a marriage M, the* additive-link *of M, denoted by $la(M)$, is the sum of the links of all its pairs, i.e., $\sum_{\{(m,w)\in M\}} la(m, w)$.*

Notice that we can use other operators beside the sum to define the link strength, such as, for example, the maximum or the product.

We now give a notion of stability that exploit the definition of the link additive-strength given above.

Definition 5 (link-additive-stability) *Given a marriage M, a man m, and a woman w, the pair (m, w) is a link-additive-blocking pair for M if the following conditions hold:*

- $la(m, w) > la(m', w)$,
- $la(m, w) > la(m, w')$,

where m' is the partner of w in M and w' is the partner of m in M. A marriage is link-additive-stable *if it does not contain link-additive-blocking pairs.*

Example 7 *Let $\{w_1, w_2\}$ and $\{m_1, m_2\}$ be, respectively, the set of women and men. Consider the following instance of an SMQ, P:*

- $m_1 : w_1^{[30]} > w_2^{[3]}$,
- $m_2 : w_1^{[4]} > w_2^{[3]}$,
- $w_1 : m_2^{[6]} > m_1^{[5]}$,

- $w_2 : m_1^{[10]} > m_2^{[2]}$.

In this example there is a unique link-additive-stable marriage, that is $M_1 = \{(m_1, w_1), (m_2, w_2)\}$*, which has additive-link* $la(M_1) = 35 + 5 = 40$*. Notice that such a marriage has an additive-link higher than the male-optimal stable marriage of* $c(P)$ *that is* $M_2 = \{(m_1, w_2), (m_2, w_1)\}$ *which has additive-link* $la(M_2) = 13 + 10 = 23$. □

The strength of the link of a pair (man,woman), and thus the notion of link stability, can be also defined by considering the maximum operator instead of the sum operator.

Definition 6 (link maximal-strength) *Given a man m and a woman w, the* link maximal-strength *of the pair (m, w), denoted by $lm(m, w)$, is the value obtained by taking the maximum between the preference value that m gives to w and the preference value that w gives to m, i.e., $lm(m, w) = max(p(m, w), p(w, m))$. Given a marriage M, the* maximal-link *of M, denoted by $lm(M)$, is the maximum of the links of all its pairs, i.e., $max_{\{(m,w)\in M\}} lm(m, w)$.*

Definition 7 (link-maximal-stability) *Given a marriage M, a man m, and a woman w, the pair (m, w) is a link-maximal-blocking pair for M if the following conditions hold:*

- $lm(m, w) > lm(m', w)$,
- $lm(m, w) > lm(m, w')$,

where m' is the partner of w in M and w' is the partner of m in M. A marriage is link-maximal-stable *if it does not contain link-maximal-blocking pairs.*

4.1 Relations with other stability notions

Given an SMQ P, let us denote with $Linka(P)$ (resp., $Linkm(P)$) the stable marriage problem with ties obtained from P by changing every preference value that a person x gives to a person y with the value $la(x, y)$ (resp., $lm(x, y)$), by changing the preference rankings accordingly, and by considering only these new preference rankings.

Let us denote with $I_{la}(P)$ (resp., $I_{lm}(P)$) the set of the link-additive-stable marriages (resp., link-maximal-stable marriages) of P and with $I_w(Linka(P))$ (resp., $I_w(Linkm(P))$) the set of the weakly stable marriages of $Linka(P)$ (resp., $Linkm(P)$). It is possible to show that these two sets coincide.

Theorem 2 *Given an SMQ P, $I_{la}(P) = I_w(Linka(P))$ and $I_{lm}(P) = I_w(Linkm(P))$.*

Proof: Let us consider a marriage M. We first show that if $M \in I_w(Linka(P))$ then $M \in I_{la}(P)$. If $M \notin I_{la}(P)$, there is a pair (m, w) that is a link-additive-blocking pair, i.e., $la(m, w) > la(m, w')$ and $la(m, w) > la(m', w)$, where w' (resp., m') is the partner of m (resp., w) in M. Since $la(m, w) > la(m, w')$, m prefers w to w' in the problem $Linka(P)$, and, since $la(m, w) > la(m', w)$, w prefers m to m' in the problem $Linka(P)$. Hence (m, w) is a blocking pair for the problem $Linka(P)$. Therefore, $M \notin I_w(Linka(P))$.

We now show that if $M \in I_{la}(P)$ then $M \in I_w(Linka(P))$. If $M \notin I_w(Linka(P))$, there is a pair (m, w) that is a blocking pair for $I_w(Linka(P))$, i.e., m prefers w to w' in the problem $Linka(P)$, and w prefers m to m' in the problem $Linka(P)$. By definition of the problem $Linka(P)$, $la(m, w) > la(m, w')$ and $la(m, w) > la(m', w)$. Therefore, (m, w) is a link-additive-blocking pair for the problem P. Hence, $M \notin I_{la}(P)$.

It is possible to show similarly that $I_{lm}(P) = I_w(Linkm(P))$. □

When no preference ordering changes in $Linka(P)$ (resp., $Linkm(P)$) w.r.t. P, then the link-additive-stable (resp., link-maximal-stable) marriages of P coincide with the stable marriages of $c(P)$.

Proposition 4 *Given an SMQ P, if $Linka(P) = c(P)$ ($Linkm(P) = c(P)$) , then $I_{la}(P) = I(c(P))$ (resp., $I_{lm}(P) = I(c(P))$).*

If there are no ties in $Linka(P)$ (resp., $Linkm(P)$), then there is a unique link-additive-stable marriage (resp., link-maximal-stable marriage) with the highest link.

Proposition 5 *Given an SMQ P, if $Linka(P)$ (resp., $Linkm(P)$) has no ties, then there is a unique link-additive-stable (resp., link-maximal-stable) marriage with the highest link.*

If we consider the definition of link-maximal-stability, it is possible to define a class of SMQs where there is a unique link-maximal-stable marriage with the highest link.

Proposition 6 *In an SMQ P where the preference values are all different, there is a unique link-maximal-stable marriage with the highest link.*

4.2 Finding link-additive-stable and link-maximal-stable marriages with the highest link

We now show that for some classes of preferences it is possible to find optimal link-additive-stable marriages and link-maximal-stable marriages of an SMQ by adapting algorithm GS, which is usually used to find the male-optimal stable marriage in classical stable marriage problems.

By Proposition 2, we know that the set of the link-additive-stable (resp., link-maximal-stable) marriages of an SMQ P coincides with the set of the weakly stable marriages of the SMP $Linka(P)$ (resp., $Linkm(P)$). Therefore, to find a link-additive-stable (resp., link-maximal-stable) marriage, we can simply apply algorithm GS to a linearization of $Linka(P)$ (resp., $Linkm(P)$).

Algorithm 2: *link-additive-stable-GS (resp., link-maximal-stable-GS)*

Input: P: an SMQ
Output: μ: a marriage
$P' \leftarrow Linka(P)$ (resp., $Linkm(P)$);
$P'' \leftarrow$ a linearization of P';
$\mu \leftarrow$ the marriage obtained by applying GS algorithm to P'';
return μ

Proposition 7 *Given an SMQ P, the marriage returned by algorithm* link-additive-stable-GS (link-maximal-stable-GS) *over P, say M, is link-additive-stable (resp., link-maximal-stable). Moreover, if there are not ties in $Linka(P)$ (resp., $Linkm(P)$), M is link-additive-stable (resp., link-maximal-stable) and it has the highest link.*

When there are no ties in $Linka(P)$ (resp., $Linkm(P)$), the marriage returned by algorithm *link-additive-stable-GS* (resp., *link-maximal-stable-GS*) is male-optimal w.r.t. the profile with links. Such a marriage may be different from the classical male-optimal stable marriage of $c(P)$, since it considers the happiness of the men reordered according to their links with the women, rather than according their single preferences.

This holds, for example, when we assume to have an SMQ with preference values that are all different and we consider the notion of link2-stability.

Proposition 8 *Given an SMQ P where the preference values are all different, the marriage returned by algorithm* link-maximal-stable-GS *algorithm over P is link-maximal-stable and it has the highest link.*

5 Conclusions and future work

In this paper we have considered stable marriage problems with quantitative preferences, where both men and women can express a score over the members of the other sex. In particular, we have introduced new stability and optimality notions for such problems and we have compared them with the classical ones for stable marriage problems with totally or partially ordered preferences. Also, we have provided algorithms to find marriages that are optimal and stable according to these new notions by adapting the Gale-Shapley algorithm.

We have also considered an optimality notion (that is, lex-male-optimality) that exploits a voting rule to linearize the partial orders. We intend to study if this use of voting rules within stable marriage problems may have other benefits. In particular, we want to investigate if the procedure defined to find such an optimality notion inherits the properties of the voting rule with respect to manipulation: we intend to check whether, if the voting rule is NP-hard to manipulate, then also the procedure on SMQ that exploits such a rule is NP-hard to manipulate. This would allow us to transfer several existing results on manipulation complexity, which have been obtained for voting rules, to the context of procedures to solve stable marriage problems with quantitative preferences.

Acnowledgements

This work has been partially supported by the MIUR PRIN 20089M932N project "Innovative and multi-disciplinary approaches for constraint and preference reasoning".

References

[1] K.J. Arrow, A.K. Sen, and K. Suzumura. *Handbook of Social Choice and Welfare*. North Holland, Elsevier, 2002.

[2] S. Bistarelli, S. N. Foley, B. O'Sullivan, and F. Santini. From marriages to coalitions: A soft csp approach. In *CSCLP*, pages 1–15, 2008.

[3] S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint solving and optimization. *Journal of the ACM*, 44(2):201–236, 1997.

[4] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *Amer. Math. Monthly*, 69:9–14, 1962.

[5] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Boston, Mass., 1989.

[6] Dan Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM J. Comput.*, 16(1):111–128, 1987.

[7] Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3):532–543, 1987.

[8] J. Liebowitz and J. Simien. Computational efficiencies for multi-agents: a look at a multi-agent system for sailor assignment. *Electonic government: an International Journal*, 2(4):384–402, 2005.

[9] D. Manlove. The structure of stable marriage with indifference. *Discrete Applied Mathematics*, 122(1-3):167–181, 2002.

[10] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Manipulation complexity and gender neutrality in stable marriage procedures. *Jornal of Autonomous Agents and Multi-Agent Systems*. To appear.

[11] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Manipulation and gender neutrality in stable marriage procedures. In *Proc. AAMAS'09*, volume 1, pages 665–672, 2009.

[12] A. E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92:991–1016, 1984.

[13] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. Gale-shapley stable marriage problem revisited: Strategic issues and applications. *Manage. Sci.*, 47(9):1252–1267, 2001.

[14] L. Tesfatsion. Gale-shapley matching in an evolutionary trade network game. *Economic Report*, 43, 1998.

Maria Silvia Pini
Department of Pure and Applied Mathematics
University of Padova, Italy
Email: `mpini@math.unipd.it`

Francesca Rossi
Department of Pure and Applied Mathematics
University of Padova, Italy
Email: `frossi@math.unipd.it`

K. Brent Venable
Department of Pure and Applied Mathematics
University of Padova, Italy
Email: `kvenable@math.unipd.it`

Toby Walsh
NICTA and UNSW, Sydney, Australia
Email: `Toby.Walsh@nicta.com.au`

Local Search for Stable Marriage Problems[1]

M. Gelain, M. S. Pini , F. Rossi, K. B. Venable, and T. Walsh

Abstract

The stable marriage (SM) problem has a wide variety of practical applications, ranging from matching resident doctors to hospitals, to matching students to schools, or more generally to any two-sided market. In the classical formulation, n men and n women express their preferences (via a strict total order) over the members of the other sex. Solving a SM problem means finding a stable marriage where stability is an envy-free notion: no man and woman who are not married to each other would both prefer each other to their partners or to being single. We consider both the classical stable marriage problem and one of its useful variations (denoted SMTI) where the men and women express their preferences in the form of an incomplete preference list with ties over a subset of the members of the other sex. Matchings are permitted only with people who appear in these preference lists, an we try to find a stable matching that marries as many people as possible. Whilst the SM problem is polynomial to solve, the SMTI problem is NP-hard. We propose to tackle both problems via a local search approach, which exploits properties of the problems to reduce the size of the neighborhood and to make local moves efficiently. We evaluate empirically our algorithm for SM problems by measuring its runtime behaviour and its ability to sample the lattice of all possible stable marriages. We evaluate our algorithm for SMTI problems in terms of both its runtime behaviour and its ability to find a maximum cardinality stable marriage. Experimental results suggest that for SM problems, the number of steps of our algorithm grows only as $O(n \log(n))$, and that it samples very well the set of all stable marriages. It is thus a fair and efficient approach to generate stable marriages. Furthermore, our approach for SMTI problems is able to solve large problems, quickly returning stable matchings of large and often optimal size despite the NP-hardness of this problem.

1 Introduction

The stable marriage problem (SM) [6] is a well-known problem of matching men to women to achieve a certain type of "stability". Each person expresses a strict preference ordering over the members of the opposite sex. The goal is to match men to women so that there are no two people of opposite sex who would both rather be matched with each other than with their current partners. The stable marriage problem has a wide variety of practical applications, ranging from matching resident doctors to hospitals, sailors to ships, primary school students to secondary schools, as well as in market trading. Surprisingly, such a stable marriage always exists and one can be found in polynomial time. Gale and Shapley give a quadratic time algorithm to solve this problem based on a series of proposals of the men to the women (or vice versa) [2].

There are many variants of the traditional formulation of the stable marriage problem. Some of the most useful in practice include incomplete preference lists (SMI), that allow one to model unacceptability for certain members of the other sex, and preference lists with ties (SMT), that model indifference in the preference ordering. With a SMI problem, the goal is to find a stable marriage in which the married people accept each other. It is known that all solutions of a SMI problem have the same size (that is, number of married people). In SMT problems, instead, solutions are stable marriages where everybody is married. Both

[1]Research partially supported by the Italian MIUR PRIN project 20089M932N: "Innovative and multidisciplinary approaches for constraint and preference reasoning".

of these variants are polynomial to solve. In real world situations, both ties and incomplete preference lists may be needed. Unfortunately, when we allow both, the problem becomes NP-hard [12]. In a SMTI (Stable Marriage with Ties and Incomplete lists) problem, there may be several stable marriages of different sizes, and solving the problem means finding a stable marriage of maximum size.

In this paper we investigate the use of a local search approach to tackle both the classical and the NP-hard variant of the problem. In particular, when we consider the classical problem, we investigate the fairness of stable marriage procedures based on local search. On the other hand, for SMTI problems, we focus on efficiency. Our algorithms are based on the same schema: they start from a randomly chosen marriage and, at each step, we move to a neighbor marriage by minimizing the distance to stability, which is measured by the number of unstable pairs. To avoid redundant computation due to the possibly large number of unstable pairs, we consider only those that are undominated, since their elimination maximises the distance to stability. Random moves are also used, to avoid stagnation in local minima. The algorithms stop when they find a solution or when a given limit on the number of steps is reached. A solution for an SMTI is a perfect matching (that is, a stable marriage with no singles), whereas, for an SM, a solution is just a stable marriage.

For the SM problem, we performed experiments on randomly generated problems with up to 500 men and women. It is interesting to notice that our algorithm always finds a stable marriage. Also, its runtime behaviour shows that the number of steps grows as little as $O(n\log(n))$. We also tested the fairness of our algorithm at generating stable marriages, measuring how well the algorithm samples the set of all stable marriages. As it is non-deterministic, it should ideally return any of the possible stable marriages with equal probability. We measure this capability in the form of an entropy that should be as close to that of an uniform sampe as possible. The computed entropy is about 70% of that of an uniform sample, and even higher on problems with small size.

For the SMTI problem, we performed experiments on randomly generated problems of size 100. We observe that our algorithm is able to find stable marriages with at most two singles on average in tens of seconds at worst. The SMTI problem has been tackled also in [4], where the problem is modeled in terms of a constraint optimization problem and solved employing a constraint solver. This systematic approach is guaranteed to find always an optimal solution. However, our experimental results show that our local search algorithm in practice always appears to find optimal solutions. Moreover, it scales well to sizes much larger than those considered in [4]. An alternative approach to local search is to use approximation methods. An overview of some results on SM problems is presented in [3].

2 Background

In this section we give some basic notions about the stable marriage problem. In addition, we present some basic notions about local search.

2.1 Stable marriage problem

A stable marriage (SM) problem [6] consists of matching members of two different sets, usually called men and women. When there are n men and n women, the SM problem is said to have size n. Each person strictly ranks all members of the opposite sex. The goal is to match the men with the women so that there are no two people of opposite sex who would both rather marry each other than their current partners. If there are no such pairs (called blocking pairs) the marriage is *"stable"*.

Definition 1 (Marriage) *Given an SM P of size n, a marriage M is a one-to-one matching of the men and the women. If a man m and a woman w are matched in M, we write $M(m) = w$ and $M(w) = m$.*

Definition 2 (Blocking pair) *Given a marriage M, a pair (m, w), where m is a man and w is a woman, is a blocking pair iff m and w are not partners in M, but m prefers w to $M(m)$ and w prefers m to $M(w)$.*

Definition 3 (Stable Marriage) *A marriage M is stable iff it has no blocking pairs.*

A convenient and widely used SM representation is showed in Table 1, where each person is followed by his/her preference list in decreasing order.

men's preference lists	women's preference lists
1: 5 7 1 2 6 8 4 3	1: 5 3 7 6 1 2 8 4
2: 2 3 7 5 4 1 8 6	2: 8 6 3 5 7 2 1 4
3: 8 5 1 4 6 2 3 7	3: 1 5 6 2 4 8 7 3
4: 3 2 7 4 1 6 8 5	4: 8 7 3 2 4 1 5 6
5: 7 2 5 1 3 6 8 4	5: 6 4 7 3 8 1 2 5
6: 1 6 7 5 8 4 2 3	6: 2 8 5 4 6 3 7 1
7: 2 5 7 6 3 4 8 1	7: 7 5 2 1 8 6 4 3
8: 3 8 4 5 7 2 6 1	8: 7 4 1 5 2 3 6 8

Table 1: An example of an SM of size 8.

For example, Table 1 shows that man 1 prefers woman 5 to woman 7 to woman 1 and so on. It is known that, at least one stable marriage exists for every SM problem. For a given SM instance, we can define a partial order relation on the set of stable marriages.

Definition 4 (Dominance) *Let M and M' be two stable marriages. M dominates M' iff every man has a partner in M who is at least as good as the one he has in M'.*

Under the partial order given by the dominance relation, the set of stable marriages forms a distributive lattice [11]. Gale and Shapley give a polynomial time algorithm (GS) to find the stable marriage at the top (or bottom) of this lattice [2]. The top of such lattice is the male optimal stable marriage M_m, that is optimal from the men's point of view. This means that there are no other stable marriages in which each man is married with the same woman or with a woman he prefers to the one in M_m. The GS algorithm can also be used to find the female optimal stable marriage M_w (that is the bottom of the stable marriage lattice), which is optimal from the women's perspective, by just replacing men with women (and vice versa) before applying the algorithm. A clear way to represent this lattice is a Hasse diagram representing the transitive reduction of the partial order relation. Figure 1 shows the Hasse diagram of the SM in Table 1.

A common concern with the standard Gale-Shapley algorithm is that it unfairly favors one sex at the expense of the other. This gives rise to the problem of finding "fairer" stable marriages. Previous work on finding fair marriages has focused on algorithms for optimizing an objective function that captures the happiness of both genders [7, 9]. A different approach is to investigate non-deterministic procedures that can generate a random stable marriage from the lattice with a distribution which is as uniform as possible.

In [1] the authors use a Markov chain approach to sample the stable marriage lattice. More precisely, the edges of the lattice dictate exactly how to formalize the moves to walk from one stable marriage to another one, so that there are at most a linear number of moves

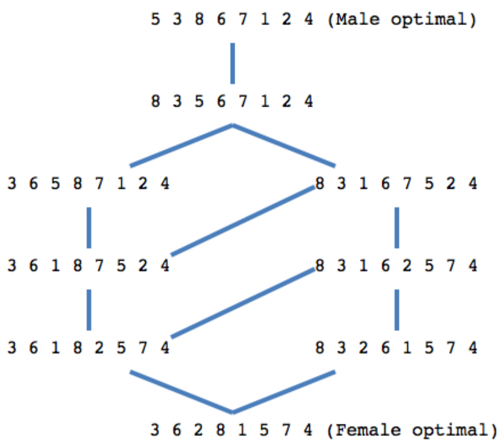


Figure 1: The Hasse diagram of the set of all stable marriages for the SM in Table 1.

at each step, these are easily identifiable, and they form reversible moves that connect the state space and converge to the uniform distribution. Unfortunately, Bhatnagar et al. show that this random walk has an exponential convergence time, which would appear to suggest that the approach may not be feasible in practice.

In this paper we also consider a variant of the SM problem where preference lists may include ties and may be incomplete. This variant is denoted by SMTI [10]. Ties express indifference in the preference ordering, while incompleteness models unacceptability only for certain partners. Finally, our empirical data support the theoretical results in [14] about parameterized complexity of the stable marriage problem.

Definition 5 (SMTI marriage) *Given a SMTI problem with n men and n women, a marriage M is a one-to-one matching between men and women such that partners accept each other. If a man m and a woman w are matched in M, we write $M(m) = w$ and $M(w) = m$. If a person p is not matched in M we say that he/she is single.*

Definition 6 (Marriage size) *Given a SMTI problem of size n and a marriage M, its size is the number of men (or women) that are married.*

Definition 7 (Blocking pairs in SMTI problems) *Consider a SMTI problem P, a marriage M for P, a man m and a woman w. A pair (m, w) is a blocking pair in M iff m and w accept each other and m is either single in M or he strictly prefers w to $M(m)$, and w is either single in M or she strictly prefers m to $M(w)$.*

Definition 8 (Weakly Stable Marriages) *Given a SMTI problem P, a marriage M for P is weakly stable iff it has no blocking pairs.*

As we will consider only weakly stable marriages, we will simply call them stable marriages. Given a SMTI problem, there may be several stable marriages of different size. If the size of a marriage coincides with the size of the problem, it is said to be a perfect matching. Solving a SMTI problem means finding a stable marriage with maximal size. This problem is NP-hard [12].

2.2 Local search

Local search [8] is one of the fundamental paradigms for solving computationally hard combinatorial problems. Local search methods in many cases represent the only feasible way for solving large and complex instances. Moreover, they can naturally be used to solve optimization problems.

Given a problem instance, the basic idea underlying local search is to start from an initial search position in the space of all solutions (typically a randomly or heuristically generated candidate solution, which may be infeasible, sub-optimal or incomplete), and to improve iteratively this candidate solution by means of typically minor modifications. At each *search step* we move to a position selected from a *local neighborhood*, chosen via a heuristic evaluation function. The evaluation function typically maps the current candidate solution to a number such that the global minima correspond to solutions of the given problem instance. The algorithm moves to the neighbor with the smallest value of the evaluation function. This process is iterated until a *termination criterion* is satisfied. The termination criterion is usually the fact that a solution is found or that a predetermined number of steps is reached, although other variants may stop the search after a predefined amount of time.

Different local search methods vary in the definition of the neighborhood and of the evaluation function, as well as in the way in which situations are handled when no improvement is possible. To ensure that the search process does not stagnate in unsatisfactory candidate solutions, most local search methods use randomization: at every step, with a certain probability a random move is performed rather than the usual move to the best neighbor.

3 Local search on Stable Marriages

We now present an adaptation of the local search schema to deal with the classical stable marriage problem. Then, we will point out the aspects that have to be changed to deal with SMTI problems.

Given an SM problem P, we start from a randomly generated marriage M. Then, at each search step, we compute the set BP of blocking pairs in M and compute the neighborhood, which is the set of all marriages obtained obtained by removing one of the blocking pairs in BP from M. Consider a blocking pair $bp = (m, w)$ in M, $m' = M(w)$, and $w' = M(m)$. Then, removing bp from M means obtaining a marriage M' in which m is married with w and m' is married with w', leaving the other pairs unchanged. To select the neighbor M' of M to move to, we use an evaluation function $f : \mathcal{M}_n \to Z$, where $\mathcal{M}_n$ is the set of all possible marriages of size n, and $f(M) = nbp(M)$. For each marriage M, $nbp(M)$ is the number of blocking pairs in M, and we move to one with the smallest value of f.

To avoid stagnation in a local minimum of the evaluation function, at each search step we perform a random walk with probability p (where p is a parameter of the algorithm), which removes a randomly chosen blocking pair in BP from the current marriage M. In this way we move to a randomly selected marriage in the neighborhood. The algorithm terminates if a stable marriage is found or when a maximal number of search steps or a timeout is reached.

This basic algorithm, called SML, has been improved in the computation of the neighborhood, obtaining SML1. When SML moves from one marriage to another one, it takes as input the current marriage M and the list $PAIRS$ of its blocking pairs and returns the marriage in the neighborhood of M with the best value of the evaluation function, i.e. the one with fewer blocking pairs. However, the number of such blocking pairs may be very large. Also, some of them may be useless, since their removal would surely lead to new marriages that will not be chosen by the evaluation function. This is the case for the so-called *dominated* blocking pairs. Algorithm SML1 considers only undominated blocking pairs.

Definition 9 (Dominance in blocking pairs) *Let (m, w) and (m, w') be two blocking pairs. Then (m, w) dominates (from the men's point of view) (m, w') iff m prefers w to w'.*

There is an equivalent concept from the women's point of view.

Definition 10 (Undominated blocking pair) *A men- (resp., women-) undominated blocking pair is a blocking pair such that there is no other blocking pair that dominates it from the men's (resp., women's) point of view.*

It is easy to see that, if M is an unstable marriage, (m, w) an men- (resp., women-) undominated blocking pair in M, $m' = M(w)$, $w' = M(m)$, and M' is obtained from M by removing (m, w), there are no blocking pairs in M' in which m (resp., w) is involved. This property would not be true if we removed a dominated blocking pair. This is why we focus on the removal of undominated blocking pairs when we pass from one marriage to another in our local search algorithm.

Considering again the SM in Table 1 and the marriage 2 7 4 8 6 3 5 1. The blocking pair (m_8, w_4) dominates (from the men's point of view) (m_8, w_2). If we remove (m_8, w_2) from the marriage, (m_8, w_4) will remain. On the other hand, removing (m_8, w_4) also eliminates (m_8, w_2). Thus, removing (m_8, w_4) is more useful than removing (m_8, w_2).

By using the undominated blocking pairs instead of all the blocking pairs, we also limit the size of the neighborhood, since each man or woman is involved in at most one undominated blocking pair. Hence we have at most $2n$ neighbor marriages to evaluate.

Let us now analyse more carefully the set of blocking pairs considered by SML1. Consider the case in which a man m_i is in two blocking pairs, say (m_i, w_j) and (m_i, w_k), and assume that (m_i, w_j) dominates (m_i, w_k) from the men's point of view. Then, let w_j be in another blocking pair, say (m_z, w_j), that dominates (m_i, w_j) from the women's point of view. In this situation, SML1 returns (m_z, w_j) because it computes the undominated blocking pairs from men's point of view (which are (m_i, w_j) and (m_z, w_j)) and, among those, maintain the undominated ones from the women's point of view ((m_z, w_j) in this case). The removal of (m_z, w_j) automatically eliminates (m_i, w_j) from the set of blocking pairs of the marriage, since it is dominated by (m_z, w_j). However, the blocking pair (m_i, w_k) is still present because the blocking pair that dominated it (i.e. (m_i, w_j)) is not a blocking pair any longer. We also consider a procedure that will return in addition the blocking pair (m_i, w_k), so to avoid having to consider it again in the subsequent step of the local search algorithm. We call SML2 the algorithm obtained from SML1 by using this new way to compute the blocking pairs.

Since dominance between blocking pairs is defined from one gender's point of view, at the beginning of our algorithms we randomly choose a gender and, at each search step we change the role of the two genders. For example, in SML1, if we start by finding the undominated blocking pairs from the men's point of view and, among those, we keep only the undominated blocking pairs from the women's point of view, in the following second step we do the opposite, and so on. In this way we ensure that SML1 and SML2 are gender neutral.

Summarizing, we have defined three algorithms, called SML, SML1, and SML2, to find a stable marriage for a given SM instance. Such algorithms differ only for the set of blocking pairs considered to define the neighborhood.

4 Local search for SMTI problems

To adapt the SML algorithm to solve problems with ties and incomplete lists it is important to recall that an SMTI may have several stable marriages of different size. Thus, solving an SMTI problem means finding a stable marriage with maximal size. If the size of the marriage coincide with the size of the problem, it is said to be perfect and the algorithm can stop before the step limit. Otherwise the algorithm returns the best marriage found during search, defined as follows: if no stable marriage has been found, then the best marriage is the one with the smallest value of the evaluation function; otherwise, it is the stable marriage with fewest singles.

The SML algorithm is therefore modified in the following ways:

- the evaluation function has to take into account that some person may be not married, so we use: $f(M) = nbp(M) + ns(M)$, where, for each marriage M, $ns(M)$ is the number of singles in M which are not in any blocking pair.
- When we remove a blocking pair (m, w) from a marriage M, their partners $M(m)$ and $M(w)$ become single.
- The algorithm performs a random restart when a stable marriage is reached, since its neighborhood is empty (because it has no blocking pairs).

We call LTIU the modified algorithm for SMTI problems, obtained from SML by the above modifications and by using undominated blocking pairs.

5 Experiments

We tested our algorithms on randomly generated sets of SM and SMTI instances. For SM problems, we generated stable marriage problems of size n using the impartial culture model (IC) [5] which assigns to each man and to each woman a preference list uniformly chosen from the $n!$ possible total orders of n persons. This means that the probability of any particular ordering is $1/n!$.

For SMTI problems, we generated problems using the same method as in [4]. More precisely, the generator takes three parameters: the problem's size n, the probability of incompleteness p_1, and the probability of ties p_2. Given a triple (n, p_1, p_2), a SMTI problem with n men and n women is generated, as follows:

1. For each man and woman, we generate a random preference list of size n, i.e., a permutation of n persons;
2. We iterate over each man's preference list: for a man m_i and for each women w_j in his preference list, with probability p_1 we delete w_j from m_i's preference list and m_i from w_j's preference list. In this way we get a possibly incomplete preference list.
3. If any man or woman has an empty preference list, we discard the problem and go to step 1.
4. We iterate over each person's (men and women's) preference list as follows: for a man m_i and for each woman in his preference list, in position $j \geq 2$, with probability p_2 we set the preference for that woman as the preference for the woman in position $j - 1$ (thus putting the two women in a tie).

Note that this method generates SMTI problems in which the acceptance is symmetric. If a man m does not accept a woman w, m is removed from w's preference list as well. This does not introduce any loss of generality because m and w cannot be matched together in any stable marriage.

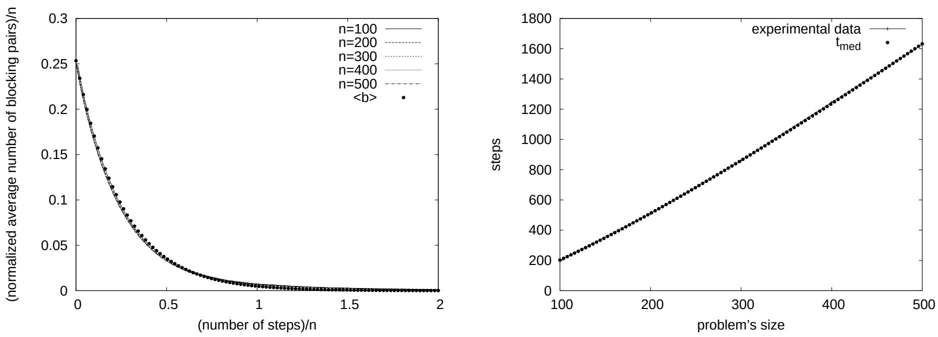
Notice also that this generator will not construct a SMTI problem in which a man (resp., woman) accepts only women (resp., men) who do not find him (resp, her) acceptable. Such a man (resp., woman) will remain single in every stable matching. A simple preprocessing step can remove such men and women from any problem, giving a smaller instance of the form constructed by our generator.

6 Results on SM problems

We measured the performance of our algorithms in terms of number of search steps. For these tests, we generated 100 SM problems for each of the following sizes: 100, 200, 300, 400 and 500. In the following we show only the results of our best algorithm, which is SML2. We studied how fast SML2 converges to a stable marriage, by measuring the ratio between the number of blocking pairs and the size of the problem during the execution. Figure 2(a) shows that SML2 has a very simple scaling behavior. Let us denote by $\langle b \rangle$ the average number of blocking pairs of the marriage found by SML2 for SM problems of size n after t steps. Then the experimental results shown in Figure 2(a) have a very good fit with the function $\langle b \rangle = an^2 2^{-bt/n}$, where a and b are constants computed empirically ($a \approx 0.25$ and $b \approx 5.7$). Figure 2(a) shows that the analytical function $\langle b \rangle$ has practically the same curve as the experimental data. The figure shows also that the average number of blocking pairs, normalized by dividing it by n, decreases during the search process in a way that is independent from the size of the problem.

We can use function $\langle b \rangle$ to conjecture the runtime behavior of our local search method. Consider the median number of steps, t_{med}, taken by SML2. Assume this occurs when half the problems have one blocking pair left and the other half have zero blocking pairs. Thus, $\langle b \rangle = \frac{1}{2}$. Substituting this value in the equation for $\langle b \rangle$, taking logs, solving for t_{med}, and grouping constant terms, we get $t_{med} = cn(d + 2\log_2(n))$ where c and d are constants. Hence, we can conclude that t_{med} grows as $O(n \log(n))$.

We then fitted this equation for t_{med} to the experimental data (using $c \approx 0.26$ and $d \approx -5.7$). The result is shown in Figure 2(b), where we see that the experimental data have the same curve as function t_{med}. This means that we can use such an equation to predict the number of steps our algorithms needs to solve a given SM instance.



(a) Blocking pair ratio during the execution. (b) Number of steps necessary to find a stable marriage.

Figure 2: Results using SML2.

6.1 Sampling the stable marriage lattice

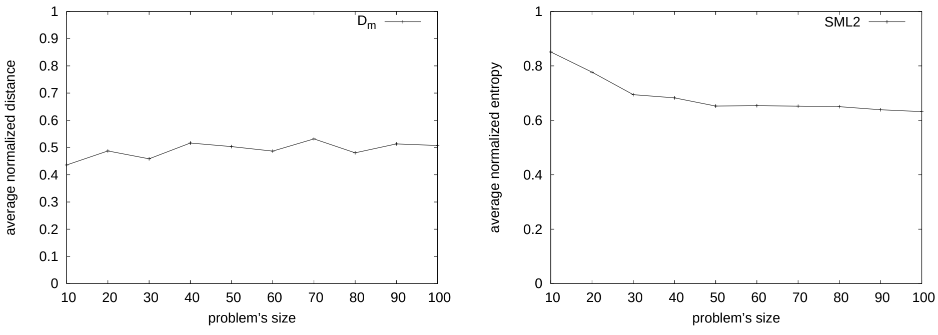
We also evaluated the ability of SML2 to sample the lattice of stable marriages of a given SM problem. To do this, we randomly generated 100 SM problems for each size between 10 and 100, with step 10. Then, we run the SML2 algorithm 500 times on each instance. To evaluate the sampling capabilities of SML2, we first measured the distance of the found stable marriages (on average) from the male-optimal marriage (the one that would be returned by the GS algorithm).

Given a SM problem P, consider a stable marriage M for P. The distance of M from M_m is the number of arcs from M to M_m in the Hasse diagram of the stable marriage lattice for P. This diagram can be computed in $O(n^2 + n|S|)$ time [7], where S is the set of all possible stable marriages of a given SM instance. For each SM problem, we compute the average normalized distance from the male-optimal marriage considering 500 runs. Notice that normalizations is needed since different SM instances with the same size may have a different number of stable lattices. Then, we compute the average D_m of these distances over all the 100 problems with the same size, which is therefore formally defined as $D_m = \frac{1}{100}\sum_{j=1}^{100}\frac{1}{500}\sum_{i=1}^{500}\frac{d_m(M_i,P_j)}{d_m(M_i,P_j)+d_w(M_i,P_j)}$, where $d_m(M_i, P_j)$ (resp., $d_w(M_i, P_j)$) is the distance of M_i from the male (resp., female)-optimal marriage in the lattice of an SM P_j. If $D_m = 0$, it means that all the stable marriages returned coincides with the male-optimal marriage. On the other extreme, if $D_m = 1$, it means that all stable marriages returned coincide with the female-optimal one. Figure 3(a) shows that, for the stable marriages returned by algorithm SML2, the average distance from the male-optimal is around 0.5.

This is encouraging but not completely informative, since an algorithm which returns the same stable marriage all the times, with distance 0.5 from the male-optimal would also have $D_m = 0.5$. To have more informative results, we consider the entropy of the stable marriages returned by SML2. This measures the randomness in the solutions. Let $f(M_i)$ be the frequency that SML2 finds a marriage M_i (for i in $[1, |S|]$) that is: $f(M_i) = \frac{1}{500}\sum_{j=1}^{500}\mathbb{1}_{M_i}(j)$, where $\mathbb{1}_{M_i}(j)$ is the indicator function that returns 1 if in the j-th execution the algorithm finds M_i, and 0 otherwise. The entropy $E(P)$ for each SM instance P (i.e., for each lattice) of size m is then: $E(P) = -\sum_{i=1\in\{1..|S|\}} f(M_i)\log_2(f(M_i))$. In an ideal case, when each stable marriage in the lattice has a uniform probability of $1/m!$ to be reached, the entropy is $\log_2(|S|)$ bits. On the other hand, the worst case is when the same stable marriage is always returned, and the entropy is thus 0 bits. As we want a measure that is independent from the problem's size, we consider a normalized entropy, that is $E(P)/\log_2(|S|)$, which is in [0,1].

As we have 100 different problems for each size, we compute the average of the normalized entropies for each class of problems with the same size: $E_n = \frac{1}{100}\sum_{i=1}^{100} E(P_i)/\log_2(|S_i|)$, where S_i is the set of stable marriages of P_i.

Figure 3(b) shows that SML2 is not far from the ideal behavior. The normalized entropy starts from a value of 0.85 per bit at size 10, decreasing to just above 0.6 per bit as the problem's size grows.



(a) Average normalized distance D_m varying n.

(b) Entropy.

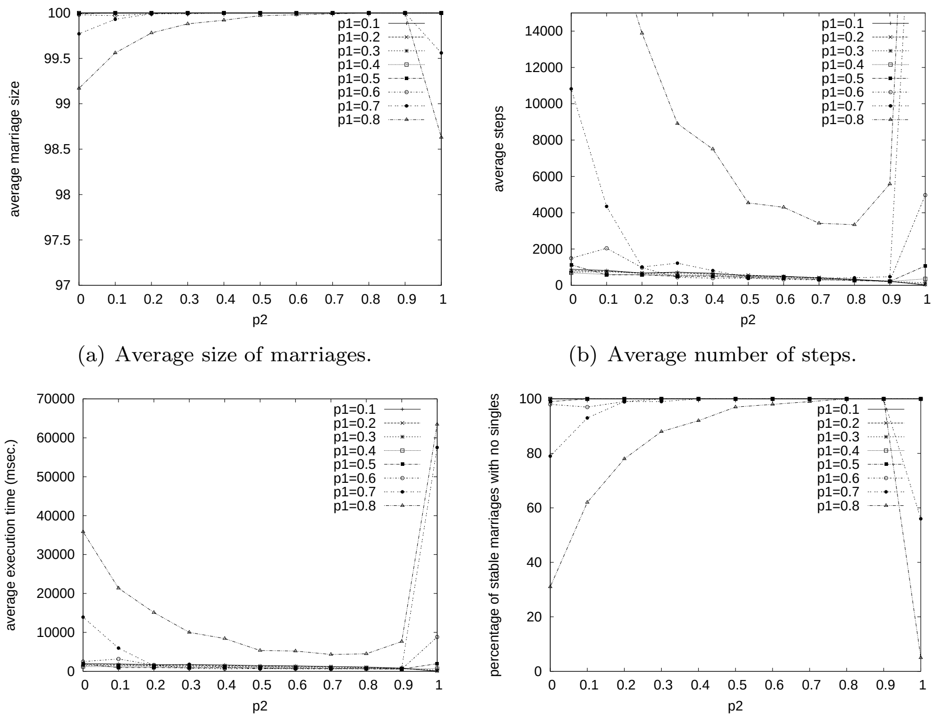
Figure 3: Sampling with SML2.

Considering both Figures 3(b) and 3(a), it appears that SML2 samples the stable marriage lattice very well. Considering also the distance D_m (Figure 3(a)), the possible outcomes appear to be equally distributed along the paths from the top to the bottom of the lattice.

7 Results on SMTI problems

We generated random SMTI problems of size 100, by letting p_2 vary in [0, 1.0] with step 0.1, and p_1 vary in [0.1, 0.8] with step 0.1 (above 0.8 the preference lists start to be empty). For each parameter combination, we generated 100 problem instances. Moreover, the probability of the random walk is set to p=20% and the search step limit is s=50000.

We start by showing the average size of the marriages returned by LTIU. In Figure 4(a) we see that LTIU almost always finds a perfect marriage (that is, a stable marriage with no singles). Even in settings with a large amount of incompleteness (that is, $p1 = 0.7$ - 0.8) the algorithm finds very large marriages, with only 2 singles on average.



(a) Average size of marriages.

(b) Average number of steps.

(c) Average execution time.

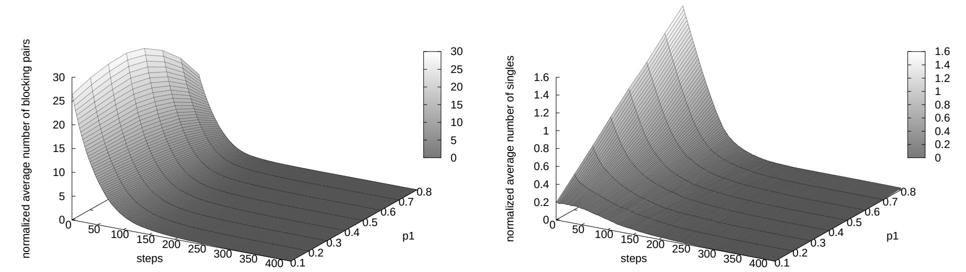
(d) Percentage of perfect matchings.

Figure 4: LTIU varying p_2 for different values of p_1.

We also consider the number of steps needed by our algorithm. From Figure 4(b), we can see that the number of steps is less than 2000 most of the time, except for problems with a large amount of incompleteness (i.e. $p_1 = 0.8$). As expected, with $p_1 > 0.6$ the algorithm requires more steps. In some cases, it reaches the step limit of 50000. Moreover, as the percentage of ties rises, stability becomes easier to achieve and thus the number of steps tends to decrease slightly. From the results we see that complete indifference (p_2=1) is a special case. In this situation, the number of steps increases for almost every value of p_1. This is because the algorithm makes most of its progress via random restarts. In these problems every person (if accepted) is equally preferred to all others accepted. The only blocking pairs are those involving singles who both accept each other. Hence, after a few steps all singles that can be married are matched, stability is reached, and the neighborhood becomes empty. The algorithm therefore randomly restarts. In this situation it is very difficult to find a perfect matching and the algorithm therefore often reached the step limit.

The algorithm is fast. It takes, on average, less than 40 seconds to give a result even for very difficult problems (see Figure 4(c)). As expected, with $p_2 = 1$ the time increases for the same reason discussed above concerning the number of steps.

Re-considering Figure 4(a) and the fact that all the marriages the algorithm finds are stable, we notice that most of the marriages are perfect. From Figure 4(d) we see that the average percentage of matchings that are perfect is almost always 100% and this percentage only decreases when the incompleteness is large. We compared our local search approach to the one in [4]. In their experiments, they measured the maximum size of the stable marriages in problems of size 10, fixing p_1 to 0.5 and varying p_2 in [0,1]. We did similar experiments, and obtained stable marriages of a very similar size to those reported in [4]. This means that although our algorithm is incomplete in principle, it always appears to find an optimal solution in practice, and for small sizes it behaves like a complete algorithm in terms of size of the returned marriage. However, it can also tackle problems of much larger sizes, still obtaining optimal solutions most of the times. We also considered the runtime behavior of our algorithm. In Figure 5(a) we show the average normalized number of blocking pairs and, in Figure 5(b), the average normalized number of singles of the best marriage as the execution proceeds. Although the step limit is 50000, we only plot results for the first steps because the rest is a long plateau that is not very interesting. We show the results only for $p_2 = 0.5$. However, for greater (resp., lower) number of ties the curves are shifted slightly down (resp., up). From Figure 5(a) we see that the average number of blocking pairs decreases very rapidly, reaching 5 blocking pairs after only 100 steps. Then, after 300-400 steps, we almost always reach a stable marriage, irrespective of the value of p_1. Considering Figure 5(b), we see that the algorithm starts with more singles for greater values of p_1. This happens because, with more incompleteness, it is more difficult for a person to be accepted. However, after 200 steps, the average number of singles becomes very small no matter the incompleteness in the problem.



(a) Average normalized number of blocking pairs.

(b) Average normalized number of singles.

Figure 5: LTIU runtime behaviour (p_2=0.5).

Looking at both Figures 5(a) and 5(b), we observe that, although we set a step limit $s = 50000$, the algorithm reaches a very good solution after just 300-400 steps. After this number of steps, the best marriage found by the algorithm usually has no blocking pairs nor singles. This appears largely independent of the amount of incompleteness and the number of ties in the problems. Hence, for SMTI problems of size 100 we could set the step limit to just 400 steps and still be reasonably sure that the algorithm will return a stable marriage of a large size, no matter the amount of incompleteness and ties.

8 Conclusions and future works

We have presented a local search approach for solving the classical stable marriage (SM) problem and its variant with ties and incomplete lists (SMTI). Our algorithm for SM problems has a simple scaling and size independent behavior and it is able to find a solution in a number of steps which grows as little as $O(n \log(n))$. Moreover it samples the stable marriage lattice reasonably well and it is a fair method to generate random stable marriages. We also provided an algorithm for SMTI problems which is both fast and effective at finding large stable marriages for problems of sizes not considered before in the literature. The algorithm was usually able to obtain a very good solution after a small amount of time.

We plan to apply a local search approach also to the hospital-resident problem and to compare our algorithms to the ones in [13], where residents express their preferences in strict order and hospitals allow ties in their preferences and have a finite number of posts each. We also aim to compare our algorithm with the Markov-chain-based model in [1] on the basis of execution time and sampling capabilities.

References

[1] Nayantara Bhatnagar, Sam Greenberg, and Dana Randall. Sampling stable marriages: why spouse-swapping won't work. In *SODA*, pages 1223–1232, 2008.

[2] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[3] M. Gelain, M. S. Pini, F. Rossi, K. V. Venable, and T. Walsh. Local search algorithms on the stable marriage problem: Experimental studies. In *ECAI*, 2010.

[4] Ian P. Gent and Patrick Prosser. An empirical study of the stable marriage problem with ties and incomplete lists. In *ECAI*, pages 141–145, 2002.

[5] G.T. Guilbauld. Les théories de l'intérêt général et le problème logique de l'agrégation. *Économie Appliquée*, 5(4):501–584, 1952.

[6] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, Boston, MA, 1989.

[7] Dan Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM J. Comput.*, 16(1):111–128, 1987.

[8] H. H. Hoos and T. Stutzle. *Stochastic Local Search: Foundations and Applications.* Elsevier - Morgan Kaufmann, 2004.

[9] Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3):532–543, 1987.

[10] Kazuo Iwama, David Manlove, Shuichi Miyazaki, and Yasufumi Morita. Stable marriage with incomplete lists and ties. In *ICALP*, pages 443–452, 1999.

[11] D.E. Knuth. *Marriages Stables.* Les Presses du l'Universit de Montral, 1976.

[12] D. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theor. Comput. Sci.*, 276(1-2):261–279, 2002.

[13] R. W. Irving and D. Manlove. Finding Large Stable Matchings. *ACM Journal of Experimental Algorithmics*, 14:1.2–1.30, 2009.

[14] D. Marx and I. Schlotter. Parameterized Complexity and Local Search Approaches for the Stable Marriage Problem with Ties. *Algorithmica*, 58(1):170-187, 2009.

Mirco Gelain, Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable
Department of Pure and Applied Mathematics, University of Padova, Italy
Email: `{mgelain,mpini,frossi,kvenable}@math.unipd.it`

Toby Walsh
NICTA and UNSW, Sidney, Australia
Email: `toby.walsh@nicta.com.au`

Distance Rationalization of Voting Rules[1]

Edith Elkind, Piotr Faliszewski, and Arkadii Slinko

Abstract

The concept of *distance rationalizability* allows one to define new voting rules or "rationalize" existing ones via a consensus class of elections and a distance. A consensus class consists of elections in which there is a consensus in the society who should win. A distance measures the deviation of the actual election from consensus elections. Together, a consensus class and a distance define a voting rule: a candidate is declared an election winner if she is the consensus candidate in one of the nearest consensus elections. It is known that many classic voting rules are defined in this way or can be represented via a consensus class and a distance, i.e., *distance-rationalized*. In this paper, we focus on the power and the limits of the distance rationalizability approach. We first show that if we do not place any restrictions on the class of possible distances then essentially all voting rules are distance-rationalizable. Thus, to make the concept of distance ratioanalizability meaningful, we have to restrict the class of distances involved. To this end, we present a very natural class of distances, which we call *votewise* distances. We investigate which voting rules can be rationalized via votewise distances and study the properties of such rules.

1 Introduction

Preference aggregation is an important task both for human societies and for multi-agent systems. Indeed, it is often the case that a group of agents has to make a joint decision, e.g., to select a unique alternative from a space of options available to them, even though the agents may have different opinions about the relative merits of these alternatives. A standard method of preference aggregation is voting. The agents submit ballots, which are usually rankings (total orders) of the alternatives (candidates), and a *voting rule* is used to select the "best" alternative. While in such settings the goal is usually to select the alternative that reflects the individual preferences of voters as well as possible, there is no universal agreement on how to reach this goal. As a consequence, there is a multitude of voting rules, and these rules are remarkably diverse (see, e.g., [4]).

Why cannot we settle on a single voting rule, which will aggregate the preferences optimally? One answer to this question is provided by the long list of impossibility theorems—starting with the famous Arrow's impossibility theorem [1]—which state that there is no voting rule (or a social welfare function) that simultaneously satisfies several natural desiderata. Thus in each real-life scenario we have to decide which of desired conditions we are willing to sacrifice.

An earlier view, initiated by Marquis de Condorcet, is that a voting rule must be a method for aggregating information. Voters have different opinions because they make errors of judgment; absent these errors, they would all agree on the best choice. The goal is to design a voting rule that identifies the best choice with highest probability. This approach is called *maximum likelihood estimation* and it has been actively pursued by Young who showed [22] that consistent application of Condorcet's ideas leads to the Kemeny rule [14]. It has been shown since then that several other voting rules can be obtained as maximum likelihood estimators for different models of errors (see Conitzer, Rognlie, and Xia [6] and

[1]This paper combines three earlier papers by the same authors: "On Distance Rationalizability of Some Voting Rules" (presented at TARK-2009), "On the Role of Distances in Defining Voting Rules" (presented at AAMAS-2010), and "Good Rationalizations of Voting Rules" (presented at AAAI-2010).

Conitzer and Sandholm [7]).

The third approach that has emerged recently in a number of papers (see, e.g., Baigent [2] and Meskanen and Nurmi [19]) can be called consensus-based. The result of each election is viewed as an imperfect approximation to some kind of electoral consensus. Under this view, the winner of a given election, or a *preference profile*, is the most preferred candidate in the "closest" consensus preference profile. The differences among voting rules can then be explained by the fact that there are several ways of defining consensus, as well as several ways of defining closeness. The heart of this approach is the decision which situations should be viewed as "electoral consensuses", be it the existence of Condorcet winner, universal agreement on which candidate is best, or something else. The concept of closeness should also be agreed upon. This approach is ideologically close to bargaining.

In this paper we concentrate on the third approach. To date, the most complete list of distance-rationalizable rules is provided by Meskanen and Nurmi [19] (but see also [2, 16, 15]). There, the authors show how to distance-rationalize many voting rules, including, among others, Plurality, Borda, Veto, Copeland, Dodgson, Kemeny, Slater, and STV. However, in Section 3 we show that the usefulness of these results is limited, as essentially every reasonable voting rule can be distance rationalized with respect to some distance and some notion of consensus. This indicates that the notion of distance rationalizability used in the early work is too broad to be meaningful. Hence, we have to determine what are the "reasonable" consensus classes and the "reasonable" distances and to reexamine all existing results.

In Section 4 we suggest a family of "good" distances (which we call *votewise* distances) and study voting rules that are distance rationalizable with respect to such distances. In particular, in Section 4.2 we show that many of the rules considered in [19], as well as all scoring rules and a variant of the Bucklin rule, can be rationalized via distances from this family. In contrast, we demonstrate that STV, which was shown to be distance-rationalizable in [19], is not distance-rationalizable via votewise distances, i.e., the restricted notion of distance rationalizability is indeed meaningful.

Now, the distance rationalizability framework can be viewed as a general method for specifying and analyzing voting rules. As such, it may be useful for proving results for entire families of voting rules, rather than isolated rules. For instance, a lot of recent research in computational social choice has focused on the complexity of determining (possible) election winners (see, e.g., [11, 17]), and the complexity of various types of attacks on elections (e.g., manipulation [8], bribery [9], and control [18, 10]).[2] However, most of the results in this line of work are specific to particular voting rules. We believe that the ability to describe multiple voting rules in a unified way (e.g., via the distance rationalizability framework) will lead to more general results. To provide an argument in favor of this belief, in Sections 4.1 and Section 4.3 we present initial results of this type, relating the type of distance and consensus used to rationalize a voting rule with the complexity of winner determination under this rule as well as the rule's axiomatic properties (such as anonymity, neutrality and consistency).

Due to space restrictions, all proofs are omitted. However, the reader may find many of them in the conference papers on which this paper is based (see the title footnote).

2 Preliminaries

2.1 Elections. An *election* is a pair $E = (C, V)$ where $C = \{c_1, \dots, c_m\}$ is the set of *candidates* and $V = (v_1, \dots, v_n)$ is an ordered list of *voters*. Each voter is represented by her *vote*, i.e., a strict, linear order over the set of candidates (also called a *preference order*).

[2] These references are only examples; an overview of literature is far beyond the scope of this paper.

We will refer to the list V as a *preference profile*, and we denote the number of voters in V by $|V|$. The number of alternatives will be denoted by $|C|$.

A *voting rule* $\mathcal{R}$ is a function that given an election $E = (C, V)$ returns a set of *election winners* $\mathcal{R}(E) \subseteq C$. Note that it is legal for the set of winners to contain more that one candidate. To simplify notation, we will sometimes write $\mathcal{R}(V)$ instead of $\mathcal{R}(E)$. We sometimes consider voting rules defined for a particular number of candidates (or even a particular set of candidates) only.

Below we define several prominent voting rules.

Scoring rules. For any sequence of non-negative real numbers $(\alpha_1, \dots, \alpha_m)$, we can define a *scoring rule* $\boldsymbol{\mathcal{R}_{(\alpha_1,\dots,\alpha_m)}}$ for elections with m candidates as follows: each candidate receives α_j points for each vote that ranks her in the jth position. The winner(s) are the candidate(s) with the highest score. Note that a scoring rule is defined for a fixed number of candidates. However, many standard voting rules can be defined via families of scoring rules. For example, *Plurality* is defined via the family of vectors $(1, 0, \dots, 0)$, *veto* is defined via the family of vectors $(1, \dots, 1, 0)$, and *Borda* is defined via the family of vectors $(m-1, m-2, \dots, 0)$; k-*approval* is the scoring rule with $\alpha_i = 1$ for $i \leq k$, $\alpha_i = 0$ for $i > k$.

Bucklin and Simplified Bucklin. Given a positive integer k, $1 \leq k \leq |C|$, we say that a candidate c is a k-*majority winner* if more than $\frac{|V|}{2}$ voters rank c among the top k candidates. Let k' be the smallest positive integer such that there is at least one k'-majority winner for E. The *Bucklin score* of a candidate c is the number of voters that rank her in top k' positions. The *Bucklin winners* are the candidates with the highest Bucklin score; clearly, all of them are k'-majority winners. The *simplified Bucklin winners* are all k'-majority winners.

Single Transferable Vote (STV). In STV the winner is chosen as follows. We find a candidate with the lowest Plurality score (i.e., one that is ranked first the least number of times) and remove him from the votes. We repeat the process until a single candidate remains; this candidate is declared to be the winner. For STV the issue of handling ties—that is, the issue of the order in which candidates with lowest Plurality scores are deleted—is quite important, and is discussed in detail by Conitzer, Rognlie and Xia [6]. However, the results in our paper are independent of the tie-breaking rule.

Dodgson. Dodgson voting is based on measuring closeness to becoming a Condorcet winner. A *Condorcet winner* is a candidate that is preferred to any other candidate by a majority of voters. The Dodgson score of a candidate c is the smallest number of swaps of adjacent candidates that have to be performed on the votes to make c a Condorcet winner. The winner(s) are the candidate(s) with the lowest score.

Kemeny. Let $\succ$ and $\succ'$ be two preference orders over C. The number of *disagreements* between $\succ$ and $\succ'$, denoted $t(\succ, \succ')$, is the number of pairs of candidates c_i, c_j such that either $c_i \succ c_j$ and $c_j \succ' c_i$ or $c_j \succ c_i$ and $c_i \succ' c_j$. A candidate c_i is a Kemeny winner if there exists a preference order $\succ$ such that c_i is ranked first in $\succ$ and $\succ$ minimizes the sum $\sum_{i=1}^{n} t(\succ, \succ_i)$. We note that usually the Kemeny rule is defined to return the ranking $\succ$ that minimizes $\sum_{i=1}^{n} t(\succ, \succ_i)$, or a set of such rankings in case of a tie; however, here we focus on rules that return sets of winners and not rankings.

2.2 Distances. Let X be a set. A function $d\colon X \to \mathbb{R} \cup \{\infty\}$ is a *distance* (or, a *metric*) if for each $x, y, z \in X$ it satisfies the following four conditions: (a) $d(x, y) \geq 0$ (nonnegativity), (b) $d(x, y) = 0$ if and only if $x = y$ (identity of indiscernibles), (c) $d(x, y) = d(y, x)$ (symmetry), and (d) $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality). If d satisfies all of the above conditions except the second one (identity of indiscernibles) then d is called a *pseudodistance*.

In the context of elections, it is useful to consider both distances over votes and over entire elections (that is, distances where the set X is the set of all linear orders over some given candidate set, and distances where X is the set of all possible elections); we remark that the former can be extended to the latter in a natural way (see the paragraph below and Section 4).

Two particularly useful distances over votes are the *discrete distance* and the *swap distance*.[3] Let C be a set of candidates and let u and v be two votes over C. The *discrete distance* $d_{\mathrm{discr}}(u, v)$ is defined to be 0 if $u = v$ and to be 1 otherwise. The *swap distance* $d_{\mathrm{swap}}(u, v)$ is the least number of swaps of adjacent candidates that transform vote u into vote v. Any distance d over votes can be extended in several ways to the distance over the profiles. For example, for any two elections, $E' = (C', V')$ and $E'' = (C'', V'')$, where $C' = C''$ and $V' = (v'_1, \dots, v'_n)$, $V'' = (v''_1, \dots, v''_n)$, we may define $\widehat{d}(E', E'') = \sum_{i=1}^{n} d(v'_i, v''_i)$ (and we set $\widehat{d}(E', E'') = \infty$ if the candidate sets are different or the profiles have different number of voters).

2.3 Consensus classes. Intuitively, we say that an election $E = (C, V)$ is a consensus if it has an undisputed winner. Formally, a *consensus class* is a pair $(\mathcal{E}, \mathcal{W})$ where $\mathcal{E}$ is a set of elections and $\mathcal{W}\colon \mathcal{E} \to C$ is a mapping which for each election $E \in \mathcal{E}$ assigns a unique alternative, which is called the *consensus alternative (winner).* We consider the following four natural classes that can be accepted by societies as consensus:

Strong unanimity. Denoted $\mathcal{S}$, this class contains elections $E = (C, V)$ where all voters report the same preference order. The consensus alternative is the candidate ranked first by all the voters.

Unanimity. Denoted $\mathcal{U}$, this class contains all elections $E = (C, V)$ where all voters rank some candidate c first. The consensus alternative is c.

Majority. Denoted $\mathcal{M}$, this class contains all elections $E = (C, V)$ where more than half of the voters rank some candidate c first. The consensus alternative is c.

Condorcet. Denoted $\mathcal{C}$, this class contains all elections $E = (C, V)$ with a Condorcet winner (defined above). The Condorcet winner is the consensus alternative.

2.4 Distance rationalizability. We now define the concept of distance rationalizability of a voting rule which has been used in the previous work.

Definition 2.1. *Let d be a distance over elections and let $\mathcal{K} = (\mathcal{E}, \mathcal{W})$ be a consensus class. We define the $(\mathcal{K}, d)$-*score *of a candidate c_i in an election E to be the distance (according to d) between E and a closest election $E' \in \mathcal{E}$ such that $c_i = \mathcal{W}(E')$. The set of $(\mathcal{K}, d)$-winners of an election $E = (C, V)$ consists of those candidates in C whose $(\mathcal{K}, d)$-score is the smallest.*

Definition 2.2. *A voting rule $\mathcal{R}$ is* distance-rationalizable *via a consensus class $\mathcal{K} = (\mathcal{E}, \mathcal{W})$ and a distance d over elections ($(\mathcal{K}, d)$-rationalizable), if for each election E, a candidate c is an $\mathcal{R}$-winner of E if and only if she is a $(\mathcal{K}, d)$-winner of E.*

Meskanen and Nurmi [19] show that many of the common voting rules are distance-rationalizable in a very natural way. For example, Kemeny is $(\mathcal{S}, \widehat{d}_{\mathrm{swap}})$-rationalizable, Borda is $(\mathcal{U}, \widehat{d}_{\mathrm{swap}})$-rationalizable, and Dodgson is $(\mathcal{C}, \widehat{d}_{\mathrm{swap}})$-rationalizable. It is quite remarkable that these three major voting rules are rationalized by the same distance. It is also easy to see that Plurality is $(\mathcal{U}, \widehat{d}_{\mathrm{discr}})$-rationalizable.

We remark that the notion of distance rationalizability introduced in Definition 2.2 allows for arbitrary consensus classes and distances; as we will see in the next section, this lack of constraints results in a definition that is too broad to be practically applicable.

[3]Swap distance is also called Kendall tau distance, Dodgson distance and bubble-sort distance.

3 Unrestricted Distance-Rationalizability: an Impasse

We say that a voting rule $\mathcal{R}$ over a set of candidates C satisfies *nonimposition* if for every $c \in C$ there exists an election with the set of candidates C in which c is the unique winner under $\mathcal{R}$. Clearly, nonimposition is a very weak condition that is satisfied by all common voting rules. Nevertheless, it turns out to be sufficient for unrestricted distance-rationalizability.

Theorem 3.1. *For any voting rule $\mathcal{R}$ over a set of candidates C that satisfies nonimposition, there is a consensus class $(\mathcal{K}, \mathcal{W})$ and a distance d such that $\mathcal{R}$ is $(\mathcal{K}, d)$-rationalizable.*

The consensus class used in the proof of Theorem 3.1 is somewhat artificial. However, the following theorem shows that a similar result holds for our natural consensus notions, too.

Definition 3.2. *Let $\mathcal{R}$ be a voting rule and let $(\mathcal{E}, \mathcal{W})$ be a consensus class. We say that $\mathcal{R}$ is* compatible *with $(\mathcal{E}, \mathcal{W})$, or $(\mathcal{E}, \mathcal{W})$-*compatible *if for each election $E = (C, V)$ in $\mathcal{E}$ it holds that $\mathcal{R}(E) = \{\mathcal{W}(E)\}$.*

Theorem 3.3. *For any consensus class $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}, \mathcal{C}\}$, a voting rule $\mathcal{R}$ is $(\mathcal{K}, d^{\mathcal{K}})$-rationalizable for some distance $d^{\mathcal{K}}$ if and only if $\mathcal{R}$ is $\mathcal{K}$-compatible.*

The proof of Theorem 3.3 is fairly simple: we construct the distance so that any given election is at distance 1 from all consensus elections with appropriate winners and at distance 2 from any other election.

Effectively, Theorem 3.3 shows that any interesting voting rule is distance-rationalizable with respect to the strong unanimity consensus. Thus, knowing that a rule is distance rationalizable—even with respect to a standard notion of consensus—provides no further insight into the properties of the rule. Moreover, the dichotomy between distance-rationalizable and non-distance-rationalizable rules becomes essentially meaningless.

However, the distances employed in the proof of Theorem 3.3 are very unnatural. In particular, the following proposition holds.

Proposition 3.4. *Let $\mathcal{R}$ be a voting rule that is $(\mathcal{K}, d^{\mathcal{K}})$-rationalizable via a consensus class $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}, \mathcal{C}\}$ and the distance $d^{\mathcal{K}}$ constructed in the proof of Theorem 3.3. If $d^{\mathcal{K}}$ is polynomial-time computable then the winner determination problem for $\mathcal{R}$ is in* P.

For example, this implies that, if P $\neq$ NP, the distance produced in the proof of Theorem 3.3 for the rationalization of Kemeny rule with respect to $\mathcal{S}$ is not polynomial-time computable. On the other hand, we know that Kemeny does have a very natural rationalization with respect to $\mathcal{S}$ via distance $\widehat{d}_{\mathrm{swap}}$. The requirement that the distance should be polynomial-time computable is essential for the distance rationalizability framework to be interesting, in addition to further, structural, restrictions on the distances that we will introduce in the next section.

4 Rationalizability via Votewise Distances

The results of the previous section make it clear that we need to restrict the set of distances that we consider. To identify an appropriate restriction, consider rationalizations of Borda and Plurality via distances $\widehat{d}_{\mathrm{swap}}$ and $\widehat{d}_{\mathrm{discr}}$, respectively (see the end of Section 2). To build either of these distances, we first defined a distance over votes and then extended it to a distance over elections (with the same candidate sets and equal-cardinality voter lists) via summing the distances between respective votes. This technique can be interpreted as taking the direct product of the metric spaces that correspond to individual votes, and defining the distance on the resulting space via the ℓ_1-norm. It turns out that distances obtained in this manner (possibly using norms other than ℓ_1), which we will call *votewise* distances, are very versatile and expressive. They are also attractive from the social choice point of view, as they exhibit continuous and monotone dependence on the voters' opinions.

In this section we will define votewise distances and attempt to answer the following three questions regarding voting rules that can be rationalized via them:

(a) What properties do such rules have?
(b) Which rules can be rationalized with respect to votewise distances?
(c) What is the complexity of winner determination for such rules?

Definition 4.1. *Given a vector space S over $\mathbb{R}$, a* norm *on S is a mapping N from S to $\mathbb{R}$ that satisfies the following properties:*

(i) positive scalability: $N(\alpha u) = |\alpha| N(u)$ for all $u \in S$ and all $\alpha \in \mathbb{R}$;
(ii) positive semidefiniteness: $N(u) \geq 0$ for all $u \in S$, and $N(u) = 0$ if and only if $u = 0$;
(iii) triangle inequality: $N(u + v) \leq N(u) + N(v)$ for all $u, v \in S$.

A well-known class of norms on $\mathbb{R}^n$ are the *p-norms* ℓ_p given by $\ell_p(x_1, \dots, x_n) = (\sum_{i=1}^{n}(|x_i|^p))^{\frac{1}{p}}$, with the convention that $\ell_\infty(x_1, \dots, x_n) = \max\{x_1, \dots, x_n\}$. A norm N on $\mathbb{R}^n$ is said to be *symmetric* if it satisfies $N(x_1, \dots, x_n) = N(x_{\sigma(1)}, \dots, x_{\sigma(n)})$ for any permutation $\sigma : [1, n] \to [1, n]$; clearly, all p-norms are symmetric. We can now define our family of votewise distances.

Definition 4.2. *We say that a function d on pairs of preference profiles is* votewise *if the following conditions hold:*

1. *$d(E, E') = +\infty$ if E and E' have a different set of candidates or a different number of voters.*
2. *For any set of candidates C, there exists a distance $d_C(\cdot, \cdot)$ defined on votes over C;*
3. *For any $n \in \mathbb{N}$, there exists a norm N_n on $\mathbb{R}^n$ such that for any two preference profiles $E = (C, U)$, $E' = (C, V)$ with $U = (u_1, \dots, u_n)$ and $V = (v_1, \dots, v_n)$ we have $d(E, E') = N_n(d_C(u_1, v_1), \dots, d_C(u_n, v_n))$.*

It is well known that any function defined in this manner is a metric. Thus, in what follows, we refer to votewise functions as *votewise distances*; we will also use the term "N-votewise distance" to refer to a votewise distance defined via a norm N, and denote a votewise distance that is based on a distance d over votes by $\widehat{d}$. Similarly, we will use the term *N-votewise rules* to refer to voting rules that can be distance-rationalized via one of our four consensus classes and an N-votewise distance.

An important special case of our framework is when N_n is the ℓ_1-norm, i.e., $N_n(x_1, \dots, x_n) = x_1 + \cdots + x_n$; we will call any such distance an *additively votewise* distance, or, in line with the notation introduced above, an *ℓ_1-votewise* distance. So far, ℓ_1-votewise distances were the only votewise distances used in distance rationalizability constructions:[4] Meskanen and Nurmi [19] use them to distance-rationalize the Kemeny rule, Dodgson, Plurality and Borda, and we will show that the construction for Borda can be generalized to all scoring rules (also using an ℓ_1-votewise distance). However, N-votewise distances with $N \neq \ell_1$ are almost as easy to work with as ℓ_1-votewise distances and may be useful for rationalizing natural voting rules. In fact, later on we will see that simplified Bucklin is an ℓ_∞-votewise rule.

[4]However, see [23, Footnote 7].

4.1 Properties of Votewise Rules

In this section we consider three basic properties of voting rules. Specifically, given a consensus class $\mathcal{K}$ and a votewise distance $\widehat{d}$, we ask under which circumstances the voting rule that is distance-rationalizable via $(\mathcal{K}, \widehat{d})$ is anonymous, neutral, or consistent. To start, we recall the formal definitions of these properties.

Let $E = (C, V)$ be an election with $V = (v_1, \ldots, v_n)$, and let σ and π be permutations of V and C, respectively. For any $C' \subseteq C$, set $\pi(C') = \{\pi(c) \mid c \in C'\}$. Let $\tilde{\pi}(v)$ be the vote obtained from v by replacing each occurrence of a candidate $c \in C$ by an occurrence of $\pi(c)$; we can extend this definition to preference profiles by setting $\tilde{\pi}(v_1, \ldots, v_n) = (\tilde{\pi}(v_1), \ldots, \tilde{\pi}(v_n))$.

Anonymity. A voting rule is *anonymous* if its result depends only on the number of voters reporting each preference order. Formally, a voting rule $\mathcal{R}$ is anonymous if for each election $E = (C, V)$ with $V = (v_1, \ldots, v_n)$ and each permutation σ of V, the election $E' = (C, \sigma(V))$ satisfies $\mathcal{R}(E) = \mathcal{R}(E')$.

Neutrality. A voting rule is *neutral* if its result does not depend on the candidates' names. Formally, a voting rule $\mathcal{R}$ is neutral if for each election $E = (C, V)$, where $C = \{c_1, \ldots, c_m\}$ and each permutation π of C, the election $E' = (C, \tilde{\pi}(V))$ satisfies $\mathcal{R}(E) = \pi^{-1}(\mathcal{R}(E'))$.

Consistency. A voting rule $\mathcal{R}$ is *consistent* if for any two elections $E_1 = (C, V_1)$ and $E_2 = (C, V_2)$ such that $\mathcal{R}(E_1) \cap \mathcal{R}(E_2) \neq \emptyset$, the election $E = (C, V_1 + V_2)$ (i.e., the election where the collections of voters from E_1 and E_2 are concatenated) satisfies $\mathcal{R}(E) = \mathcal{R}(E_1) \cap \mathcal{R}(E_2)$. This property was introduced by Young [21] and is also known as *reinforcement* [5].

For votewise distance-rationalizable rules, a symmetric norm produces an anonymous rule.

Proposition 4.3. *Suppose that a voting rule $\mathcal{R}$ is $(\mathcal{K}, \widehat{d})$-rationalizable, where $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}, \mathcal{C}\}$ and $\widehat{d}$ is an N-votewise distance, where N is a symmetric norm. Then $\mathcal{R}$ is anonymous.*

In contrast, neutrality is inherited from the underlying distance over votes.

Definition 4.4. *Let C be a set of candidates and let d be a distance on votes over C. We say that d is* neutral *if for each permutations π over C and any two votes u and v over C it holds that $d(u, v) = d(\tilde{\pi}(u), \tilde{\pi}(v))$. Further, we say that a votewise distance $\widehat{d}$ that corresponds to a distance d on votes is* neutral *if d is.*

Proposition 4.5. *Suppose that a voting rule $\mathcal{R}$ is $(\mathcal{K}, \widehat{d})$-rationalizable, where $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}, \mathcal{C}\}$ and $\widehat{d}$ is a neutral votewise distance. Then $\mathcal{R}$ is neutral.*

It is natural to ask if the converse of Proposition 4.5 is also true, i.e., if every neutral votewise rule can be rationalized via a neutral distance. Indeed, paper [6] provides a positive answer to a similar question in the context of representing voting rules as maximum likelihood estimators. However, the natural extension of the approach of [6] is not necessarily applicable in our setting. Nevertheless, all votewise distances that have so far arisen in the study of distance rationalizability of natural voting rules are neutral.

Our results for anonymity and neutrality are applicable to all consensus classes considered in this paper. In contrast, when discussing consistency, we need to limit ourselves to the unanimity consensus, and to ℓ_p-votewise rules.

Theorem 4.6. *Suppose that a voting rule $\mathcal{R}$ is $(\mathcal{U}, \widehat{d})$-rationalizable, where $\widehat{d}$ is an ℓ_p-votewise distance. Then $\mathcal{R}$ is consistent.*

While Theorem 4.6 may hold for some norms other than ℓ_p, we cannot hope to prove it for all votewise distances: fundamentally, consistency is a constraint on the relationship among N_s, N_t and N_{s+t} (i.e., the norms used for s voters, t voters, and $s+t$ voters), and our definition of votewise distances allows us to select norms N_n for different values of n independently of each other. Further, for our proof to work, the consensus class should be closed with respect to "splitting" and "merging" of the consensus profiles, and neither of the classes $\mathcal{S}$, $\mathcal{C}$, and $\mathcal{M}$ satisfies both of these conditions. Indeed, for $\mathcal{S}$ and $\mathcal{C}$ the conclusion of the theorem itself is not true: the counterexamples are provided by the Kemeny rule and the Dodgson rule, respectively (both are not consistent, yet rationalizable via $\widehat{d}_{\rm swap}$).

4.2 ℓ_p-Votewise Rules

Now that we know that ℓ_p-votewise rules have some desirable properties, let us see which voting rules are in fact ℓ_p-votewise distance rationalizable. We will generally focus on additively votewise rules, but we will look at ℓ_∞ as well. Naturally, we expect the answer to this question to strongly depend on the consensus notion used. Thus, let us consider unanimity, strong unanimity, majority, and Condorcet consensuses one by one.

We start with the unanimity consensus. By combining Propositions 4.3, 4.5 and Theorem 4.6, we conclude that any rule that is $(\mathcal{U}, \widehat{d})$-rationalizable, where $\widehat{d}$ is a neutral ℓ_1-votewise distance, is neutral, anonymous and consistent; it is not hard to check that the conclusion still holds if $\widehat{d}$ is a pseudodistance rather than a distance. In contrast, Young's famous characterization result [21] says that every voting rule that has all three of these properties is either a scoring rule or a composition of scoring rules (see [21] for an exact definition of composition of voting rules). It turns out that our framework allows us to refine Young's result by characterizing exactly the scoring rules themselves rather than their compositions. Moreover, we can actually "extract" the scoring rule from the corresponding distance, albeit not efficiently (see Section 4.3 for a discussion of the related complexity issues).

Theorem 4.7. *Let $\mathcal{R}$ be a voting rule. There exists a neutral ℓ_1-votewise pseudodistance $\widehat{d}$ such that $\mathcal{R}$ is $(\mathcal{U}, \widehat{d})$-rationalizable if and only if $\mathcal{R}$ can be defined via a family of scoring rules.*[5]

That is, the above theorem gives a complete characterization of voting rules rationalizable via neutral ℓ_1-votewise distances with respect to the unanimity consensus. However, the situation with respect to other consensus notions is more difficult.

Let us consider strong unanimity next. Intuitively, strong unanimity is quite challenging to work with as it provides very little flexibility. Meskanen and Nurmi [19] have shown that Kemeny is ℓ_1-votewise with respect to $\mathcal{S}$, but, at least at first, it seems that no other natural rule is. Interestingly, and very counterintuitively, Plurality is also ℓ_1-votewise with respect to strong unanimity.

Theorem 4.8. *There exists an ℓ_1-votewise distance $\widehat{d}$ such that Plurality rule is $(\mathcal{S}, \widehat{d})$-rationalizable.*

Naturally, this result suggests that, perhaps, all scoring rules are votewise distance-rationalizable with respect to $\mathcal{S}$. However, this turns out to be false.

Theorem 4.9. *There is no ℓ_1-votewise distance $\widehat{d}$ such that Borda rule is $(\mathcal{S}, \widehat{d})$-rationalizable.*

[5]Note that in this paper, following Young [21], we do not require $(\alpha_1, \ldots, \alpha_m)$ to be nondecreasing or integer. Indeed, the distance rationalizability framework does not impose any ordering over different positions in a vote, so it works equally well for a scoring rule with, e.g., $\alpha_1 < \alpha_2$.

Thus, the class of rules ℓ_1-votewise rationalizable with respect to $\mathcal{S}$ is rather enigmatic. On the one hand, it does contain Kemeny, a very complex rule, and Plurality, a very simple rule, yet it does not contain other natural scoring rules such as Borda. We believe that characterizing this class exactly is a very interesting research problem, particularly so since the rules in this class can be shown to be related to MLERIV rules of [7] and and [6] (we omit a description of this connection here due to space constraints).

Our understanding of rules that are votewise rationalizable with respect to $\mathcal{C}$ and $\mathcal{M}$ is even more limited. For example, Meskanen and Nurmi [19] have shown that Dodgson is ℓ_1-votewise rationalizable with respect to $\mathcal{C}$, and it is easy to see that no scoring rule is distance-rationalizable with respect to $\mathcal{C}$ because scoring rules are not Condorcet-consistent [20]. It is very interesting if, e.g., Young's rule is votewise with respect to $\mathcal{C}$ (however, see Section 5 for some comments). For the case of $\mathcal{M}$, we can show that simplified Bucklin is ℓ_∞-votewise with respect to $\mathcal{M}$; note that this result provides an argument for considering votewise distances that use a norm other that ℓ_1.

Theorem 4.10. *Simplified Bucklin is ℓ_∞-votewise with respect to consensus $\mathcal{M}$.*

The regular Bucklin rule is also rationalizable via a distance very similar to the one for simplified Bucklin but, nonetheless, not votewise. Finding further natural voting rules that are votewise rationalizable with respect to either $\mathcal{C}$ or $\mathcal{M}$ is an open question.

We conclude this section with a quick look at the STV rule. Conitzer, Rognlie, and Xia [6] have shown that STV is not MLERIV. It can be shown that this implies that STV is not distance-rationalizable via an ℓ_1-votewise distance with respect to $\mathcal{S}$. It turns out that this result can be extended to (almost) any votewise distance as well as two other consensus classes, namely, $\mathcal{U}$ and $\mathcal{C}$.

Definition 4.11 ([3])**.** *A norm N in $\mathbb{R}^n$ is* monotonic in the positive orthant*, or* $\mathbb{R}^n_+$-monotonic*, if for any two vectors $(x_1,\ldots,x_n),(y_1,\ldots,y_n) \in \mathbb{R}^n_+$ such that $x_i \le y_i$ for all $i = 1,\ldots,n$ we have $N(x_1,\ldots,x_n) \le N(y_1,\ldots,y_n)$.*

We say that a votewise distance is *monotonic* if the respective norm is monotonic in the positive orthant. We remark that monotonicity is a very weak constraint that is satisfied by any reasonable norm.

Theorem 4.12. *STV (together with any intermediate tie-breaking rule) is not distance-rationalizable with respect to either of $\mathcal{S}$, $\mathcal{U}$, or $\mathcal{C}$ and any neutral anonymous monotonic votewise distance.*

Note that Meskanen and Nurmi [19] show that STV can be distance-rationalized with respect to $\mathcal{U}$, but their distance is not votewise, and it is not immediately clear whether it is polynomial-time computable.

4.3 Winner Determination for Votewise Rules

Now that we have some understanding of the nature of votewise rules, we are ready to study the complexity of determining winners under them.[6] Clearly, to prove upper bounds on the complexity of this problem, we need to impose restrictions on the complexity of the distance itself. Thus, in what follows, we focus on distances that take values in $\mathbb{Z} \cup \{\infty\}$ and are polynomial-time computable; we will call a distance *normal* if it has both of these properties. We remark that restricting ourselves to distances with values in $\mathbb{Z} \cup \{\infty\}$ may prevent us from using ℓ_p-distances for values of p other than 1 and ∞. For example, taking

[6] We assume the reader is familiar with standard notions of complexity theory and fixed-parameter complexity. Due to space limits we cannot provide appropriate background in the paper.

the p-th root of an integer may yield a non-integer value. However, it is easy to see that for winner-determination, instead of using an ℓ_p-distance d, we can use function d^p, despite the fact that it is not a distance. This is so, because for winner-determination we only need to compare distances between elections.

The winner determination problem can be formally stated as follows.

Definition 4.13. *Let $\mathcal{R}$ be a voting rule. In the* $\mathcal{R}$-winner problem *we are given an election $E = (C, V)$ and a candidate $c \in C$ and we ask whether $c \in \mathcal{R}(E)$.*

This problem can be hard even for ℓ_1-votewise rules: for Dodgson and Kemeny it is known to be Θ_2^p-complete [11, 12]. On the positive side, for both of these rules the winner determination problem can be solved in polynomial time if the number of candidates is fixed. In fact, a stronger statement is true: the winner determination problem for both Dodgson and Kemeny is fixed parameter tractable with respect to the number of candidates.

We will now show that from the complexity perspective, Dodgson and Kemeny exhibit some of the worst possible behavior.

Theorem 4.14. *Suppose that a voting rule $\mathcal{R}$ is $(\mathcal{K}, d)$-rationalizable, where $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}, \mathcal{C}\}$, and d is a normal distance that satisfies $d((C_1, V_1), (C_2, V_2)) = +\infty$ whenever $C_1 \neq C_2$ or $|V_1| \neq |V_2|$. Then the $\mathcal{R}$-winner problem is in* P^{NP}. *Moreover, if, in addition, for any two elections $E_1 = (C, V_1)$, $E_2 = (C, V_2)$, the distance $d(E_1, E_2)$ is either $+\infty$ or at most polynomial in $|C| + |V_1| + |V_2|$, then the $\mathcal{R}$-winner problem is in Θ_2^p.*

Note that the distance used to rationalize Dodgson and Kemeny is polynomially bounded. On the other hand, there are natural distances that are not polynomially bounded; this includes distances that appear in our distance rationalizabiliy constructions for scoring rules with "large" coefficients.

If, in addition to being normal, the distance in question is an ℓ_1-votewise distance, the winner determination problem is fixed-parameter tractable with respect to the number of candidates.

Theorem 4.15. *Suppose that a voting rule $\mathcal{R}$ is $(\mathcal{K}, d)$-rationalizable, where $\mathcal{K} \in \{\mathcal{S}, \mathcal{U}, \mathcal{M}, \mathcal{C}\}$. and d is a normal ℓ_1-votewise distance. Then the $\mathcal{R}$-winner problem is FPT with respect to the number of candidates.*

In the previous section we have seen that neutral ℓ_1-votewise rules that use unanimity consensus correspond to families of scoring rules. Thus, one would expect their winner problems to be in P. Note, however, that in our setting we are given the distance, but not the scoring vector and computing the latter from the former might be hard. Nevertheless, it turns out that in this setting we can easily determine the winner if we are allowed to use polynomial-size *advice.*

Theorem 4.16. *Suppose that a voting rule $\mathcal{R}$ is distance-rationalizable via a normal neutral ℓ_1-votewise distance and unanimity consensus. Then $\mathcal{R}$-winner is in* P/poly.

P/poly is a complexity class that captures the power of polynomial computation "with advice." Karp–Lipton theorem [13] says that if there is an NP-hard problem in P/poly then the Polynomial Hierarchy collapses. Thus, for voting rules that are distance-rationalizable via a normal neutral ℓ_1-votewise distance and the consensus class $\mathcal{U}$ the winner determination problem is unlikely to be NP-hard. In contrast, this problem is hard for both Dodgson and Kemeny, even though they are both rationalizable via a normal neutral ℓ_1-votewise distance (and consensus classes $\mathcal{C}$ and $\mathcal{S}$, respectively). Thus, from computational perspective, the unanimity consensus appears to be easier to work with than the strong consensus and the Condorcet consensus. Indeed, both $\mathcal{S}$ and $\mathcal{C}$ impose "global" constraints on the closest consensus and $\mathcal{U}$ only imposes "local" ones.

5 Conclusions and Open Problems

In this paper we have presented general results regarding the recently introduced distance rationalizability framework. Our paper has two main contributions. First, we have shown that without any restrictions, essentially every reasonable voting rule is distance-rationalizable and further refinement of this framework is needed. Second, we have put forward a natural class of distances to consider—votewise distances—and proved that the rules which can be distance-rationalized using such distances have several desirable properties. We have identified a number of votewise rules, as well as showed that some rules are not votewise rationalizable with respect to standard consensus classes, and established complexity results for winner determination under votewise rules.

Are votewise distances the only natural distances that one should consider? Such distances are based on the assumption that, given an election $E = (C, V)$, if a voter changes her opinion in a minor way, then the resulting election $E' = (C, V')$ must not deviate from E too far. However some rules have discontinuous nature by definition, especially Young's rule which picks the winner of a largest Condorcet-consistent subelection. It is unlikely that such rules can be distance-rationalized via a votewise distance. Indeed, it can be shown that Young's rule and Maximin can be rationalized with respect to $\mathcal{C}$ via fairly intuitive distances that operate on profiles with different numbers of voters: in the case of Maximin we are, essentially, adding voters, and in the case of Young, we are deleting voters. (We omit the definitions of these rules and the construction due to space constraints). However, neither of these rules is known to be votewise rationalizable. Thus, it would be desirable to extend the class of "acceptable" distances to include some non-votewise distances; how to do this is an interesting research direction.

We mention that our work is closely related to a sequence of papers of Conitzer, Rognlie, Sandholm, and Xia [7, 6] on interpreting voting rules as maximum likelihood estimators. There are some very interesting connections (and differences) between the two approaches, but, unfortunately, due to space constraints, we cannot elaborate on them here.
Acknowledgements: This research was partially supported by NRF (Singapore NRF Fellowship RF-2009-08) and AGH University of Technology (Grant no. 11.11.120.865).

References

[1] K. Arrow. *Social Choice and Individual Values.* John Wiley and Sons, 1951 (revised editon, 1963).

[2] N. Baigent. Metric rationalisation of social choice functions according to principles of social choice. *Mathematical Social Sciences*, 13(1):59–65, 1987.

[3] F. Bauer, J. Stoer, and C. Witzgall. Absolute and monotonic norms. *Numerische Matematic*, 3:257–264, 1961.

[4] S. Brams and P. Fishburn. Voting procedures. In K. Arrow, A. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare, Volume 1*, pages 173–236. Elsevier, 2002.

[5] P. Yu. Chebotarev and E. Shamis. Characterizations of scoring methods for preference aggregation. *Annals of Operations Research*, 80:299 332, 1998.

[6] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. AAAI Press, July 2009.

[7] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence*, pages 145–152. AUAI Press, July 2005.

[8] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[9] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[10] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[11] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[12] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[13] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309. ACM Press, April 1980. An extended version has also appeared as: Turing machines that take advice, *L'Enseignement Mathématique*, 2nd series, 28:191–209, 1982.

[14] J. Kemeny. Mathematics without numbers. *Daedalus*, 88:575–591, 1959.

[15] C. Klamler. Borda and Condorcet: Some distance results. *Theory and Decision*, 59(2):97–109, 2005.

[16] C. Klamler. The Copeland rule and Condorcet's principle. *Economic Theory*, 25(3):745–749, 2005.

[17] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedins of the Multidisciplinary IJCAI-05 Worshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[18] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.

[19] T. Meskanen and H. Nurmi. Closeness counts in social choice. In M. Braham and F. Steffen, editors, *Power, Freedom, and Voting*. Springer-Verlag, 2008.

[20] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1991.

[21] H. Young. Social choice scoring functions. *SIAM Journal on Applied Mathematics*, 28(4):824–838, 1975.

[22] H. Young. Optimal voting rules. *Journal of Economic Perspectives*, 9(1):51–64, 1995.

[23] W. Zwicker. Consistency without neutrality in voting rules: When is a vote an average? *Mathematical and Computer Modelling*, 48(9–10):1357–1373, 2008.

Edith Elkind
Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
Email: eelkind@gmail.com

Piotr Faliszewski
Department of Computer Science
AGH University of Science and Technology
Kraków, Poland
Email: faliszew@agh.edu.pl

Arkadii Slinko
Department of Mathematics
University of Auckland
Auckland, New Zealand
Email: slinko@math.auckland.ac.nz

Social Choice without the Pareto Principle under Weak Independence

Ceyhun Coban and M. Remzi Sanver

Abstract

We show that the class of social welfare functions that satisfy a weak independence condition identified by Campbell (1976) and Baigent (1987) is fairly rich and freed of a power concentration on a single individual. This positive result prevails when a weak Pareto condition is imposed. Hence, we can overcome the impossibility of Arrow (1951) by simultaneously weakening the independence and Pareto conditions. Moreover, under weak independence, an impossibility of the Wilson (1972) type vanishes.

1 Introduction

We consider the preference aggregation problem in a society which confronts at least three alternatives. A *Social Welfare Function* (SWF) is a mapping which assigns a social ranking to any logically possible profile of individual rankings. A SWF is *independent of irrelevant alternatives* (IIA) if the social ranking of any pair of alternatives depends only on individuals' preferences over that pair. We know, since the seminal work of Arrow (1951), that IIA and Pareto optimality are incompatible, unless one is ready to admit dictatorial SWFs.

The Arrovian impossibility is remarkably robust against weakenings of IIA.[1] For example, letting k stand for the number of alternatives that the society confronts, Blau (1971) proposes the concept of *m-ary independence* for any integer between 2 and k. A SWF is *m-ary independent* if the social ranking of any set of alternatives with cardinality m depends only on individuals' preferences over that set. Clearly, when $m = 2$, m-ary independence coincides with IIA. Moreover, every SWF trivially satisfies m-ary independence when $m = k$. It is also straighforward to see that m-ary independence implies n-ary independence when $m < n$. Nevertheless, Blau (1971) shows that m-ary independence implies n-ary independence when $n < m < k$ as well. Thus, weakening IIA by imposing independence over sets with cardinality more than two does not allow to escape the Arrovian impossibility, unless independence is imposed over the whole set of alternatives - a condition which is satisfied by the definition of a SWF.

Campbell and Kelly (2000a, 2007) further weaken m-ary independence by requiring that the social preference over a pair of alternatives depends only on individuals' preferences over some proper subset of the set of available alternatives. This condition, which they call *independence of some alternatives* (ISA) is considerably weak. As a result, non-dictatorial SWF that satisfy Pareto optimality and ISA -such as the "gateau rules" identified by Campbell and Kelly (2000a)- do exist. On the other hand, "gateau rules" fail neutrality and as Campbell and Kelly (2007) later show, within the Arrovian framework, an extremely weaker version of ISA disallows both anonymity and neutrality.

Denicolo (1998) identifies a condition called *relational independent deciseveness* (RID). He shows that although IIA implies RID, the Arrovian impossibility prevails when IIA is replaced by RID.

[1] In fact, it is robust against weakenings of other conditions as well: Wilson (1972) shows that the Arrovian impossibility essentially prevails when the Pareto condition is not used. Ozdemir and Sanver (2007) identify severely restricted domains which exhibit the Arrovian impossibility.

Campbell (1976) proposes a weakening of IIA which requires that the social decision between a pair of alternatives cannot be reversed at two distinct preference profiles that admit the same individual preferences over that pair. We refer to this condition as *quasi IIA*.[2] Baigent (1987) shows that every Pareto optimal and quasi IIA SWF must be dictatorial in a sense which is close to the Arrovian meaning of the concept - hence a version of the Arrovian impossibility.[3]

In brief, the literature which explores the effects of weakening IIA on the Arrovian impossibility presents results of a negative nature. We revisit this literature in order to contribute by a positive result. We show that under the weakening proposed by Baigent (1987), the Arrovian impossibility can be surpassed by dropping the Pareto condition: We characterize the class of quasi IIA SWFs and show that this is a fairly large class which is not restricted to SWFs where the decision power is concentrated on one given individual. In fact, this class contains SWFs that are both anonymous and neutral. This positive result prevails when a weak version of the Pareto condition is imposed.

Our findings pave the way to surpass the impossibility of Arrow (1951). Moreover, we establish that there is no tension between quasi IIA and the transitivity of the social outcome. Thus, we also contrast the results of Wilson (1972) and Barberà (2003) who show that the Pareto condition has little impact on the Arrovian impossibility which is essentially a tension between IIA and the range restriction imposed over SWFs.

Section 2 presents the basic notions. Section 3 states our results. Section 4 makes some concluding remarks.

2 Basic Notions

We consider a finite set of individuals N with $\#N \geq 2$, confronting a finite set of alternatives A with $\#A \geq 3$. An *aggregation rule* is a mapping $f : \Pi^N \to \Theta$ where Π is the set of complete, transitive and antisymmetric binary relations over A while Θ is the set of complete binary relations over A. We conceive $P_i \in \Pi$ as the *preference* of $i \in N$ over A.[4] We write $P = (P_1, ..., P_{\#N}) \in \Pi^N$ for a *preference profile* and $f(P) \in \Theta$ reflects the social preference obtained by the aggregation of P through f. Note that $f(P)$ need not be transitive. Moreover, as $f(P)$ need not be antisymmetric, we write $f^*(P)$ for its strict counterpart.[5]

An aggregation rule f is *independent of irrelevant alternatives* (IIA) iff given any distinct $x, y \in A$ and any $P, P' \in \Pi^N$ with $x\ P_i\ y \iff x\ P'_i\ y\ \forall i \in N$, we have $x\ f(P)\ y \iff x\ f(P')\ y$. We write Φ for the set of aggregation rules which satisfy IIA. For any distinct $x, y \in A$, let $\{\substack{x \\ y}, \substack{y \\ x}, xy\}$ be the set of complete and transitive preferences over $\{x, y\}$.[6] An *elementary aggregation rule* is a mapping $f_{\{x,y\}} : \{\substack{x \\ y}, \substack{y \\ x}\}^N \to \{\substack{x \\ y}, \substack{y \\ x}, xy\}$. Any family $f = \{f_{\{x,y\}}\}$ of elementary aggregation rules indexed over all possible distinct pairs $x, y \in A$ induces an aggregation rule as follows: For each $P \in \Pi^N$ and each $x, y \in A$, let $x\ f(P)$

[2] See Campbell (1976) for a discussion of the computational advantages of quasi IIA. Note that when social indifference is not allowed, IIA and quasi IIA are equivalent.

[3] Baigent (1987) claims this impossibility in an environment with at least three alternatives. Nevertheless, Campbell and Kelly (2000b) show the existence of Pareto optimal and quasi IIA SWF when there are precisely three alternatives. They also show that the impossibility announced by Baigent (1987) prevails when there are at least four alternatives and even under restricted domains.

[4] As usual, for any distinct $x, y \in A$, we intepret $x\ P_i\ y$ as x being preferred to y in view of i.

[5] So for any distinct $x, y \in A$, we have $x\ f^*(P)\ y$ whenever $x\ f(P)\ y$ and not $y\ f(P)\ x$.

[6] We interpret $\substack{x \\ y}$ as x being preferred to y; $\substack{y \\ x}$ as y being preferred to x; and xy as indifference between x and y.

$y \iff f_{\{x,y\}}(P^{\{x,y\}}) \in \{\substack{x \\ y}, xy\}$ where $P^{\{x,y\}} \in \{\substack{x \\ y}, \substack{y \\ x}\}^N$ is the restriction of $P \in \Pi^N$ over $\{x, y\}$.[7] Note that $f = \{f_{\{x,y\}}\} \in \Phi$. Moreover, any $f \in \Phi$ can be expressed in terms of a family $\{f_{\{x,y\}}\} = f$ of elementary aggregation rules.

Let $\Re$ be the set of complete and transitive binary relations over A. A *Social Welfare Function* (SWF) is an aggregation rule whose range is restricted to $\Re$. A SWF $\alpha : \Pi^N \to \Re$ is *Pareto optimal* iff given any distinct $x, y \in A$ and any $P \in \Pi^N$ with $x\ P_i\ y\ \forall i \in N$, we have $x\ \alpha^*(P)\ y$. A SWF $\alpha : \Pi^N \to \Re$ is *dictatorial* iff $\exists i \in N$ such that $x\ P_i\ y$ implies $x\ \alpha^*(P)\ y\ \forall P \in \Pi^N, \forall x, y \in A$. The Arrovian impossibility, as we consider, announces that a SWF $\alpha : \Pi^N \to \Re$ is Pareto optimal and IIA if and only if is α dictatorial.

3 Results

Baigent (1987) proves a version of the Arrovian impossibility where IIA and dictatoriality are replaced by their following weaker versions: A SWF α is *quasi IIA* iff given any distinct $x, y \in A$ and any $P, P' \in \Pi^N$ with $x\ P_i\ y \iff x\ P'_i\ y\ \forall i \in N$, we have $x\ \alpha^*(P)\ y \Rightarrow x\ \alpha(P')\ y$. Note that quasi IIA and IIA coincide when indifferences are ruled out from the social preference. A SWF α is *weakly dictatorial* iff $\exists i \in N$ such that $x\ P_i\ y$ implies $x\ \alpha(P)\ y\ \forall P \in \Pi^N, \forall x, y \in A$. Baigent (1987) establishes that every Pareto optimal and quasi IIA SWF is a weak dictatorship. Nevertheless, we remark that, unlike the original version of the Arrovian impossibility, the converse statement is not true: Although every weak dictatorship is quasi IIA, there exists weak dictatorships that are not Pareto optimal.[8] Following this remark, we allow ourselves to the state a slight generalization of this theorem of Baigent (1987), corrected by Campbell and Kelly (2000b)[9]:

Theorem 3.1 *Let $\#A \geq 4$. Within the family of Pareto optimal SWFs, a SWF $\alpha : \Pi^N \to \Re$ is quasi IIA iff a is weakly dictatorial.*

We now explore the effect of being confined to the class of Pareto optimal SWFs. The strict counterpart of $T \in \Theta$ is denoted T^*. Let $\rho : \Theta \longrightarrow 2^{\Re}$ stand for the correspondence which transforms each $T \in \Theta$ over A into a non-empty subset of $\Re$ such that $\rho(T) = \{R \in \Re : xTy \implies xRy, \forall x, y \in A\}$. To have a clearer understanding of ρ, we recall that every $T \in \Theta$ induces an ordered list of "cycles".[10] A set $Y \in 2^A \backslash \{\emptyset\}$ is a *cycle* (with respect to $T \in \Theta$) iff Y can be written as $Y = \{y_1, ..., y_{\#Y}\}$ such that $y_i\ T\ y_{i+1}\ \forall i \in \{1, ..., \#Y - 1\}$ and $y_{\#Y}\ T\ y_1$. The *top-cycle* of $X \in 2^A \backslash \{\emptyset\}$ with respect to $T \in \Theta$ is a cycle $C(X, T) \subseteq X$ such that $y\ T\ {}^*x\ \forall y \in C(X, T), \forall x \in X \backslash C(X, T)$.[11] Now let $A_1 = C(A, T)$ and recursively define $A_i = C(A \backslash \bigcup_{k=1}^{i-1} A_k, T), \forall i \geq 2$. Given the finiteness of A, there exists an integer k such that $A_{k+1} = \emptyset$. So every $T \in \Theta$ induces a unique ordered partition $(A_1, A_2, ..., A_k)$ of A. It follows from the definition of the top-cycle that whenever $i < j$, we have $xT^*y\ \forall x \in A_i$, $\forall y \in A_j$.

Lemma 3.1 *Take any $T \in \Theta$ which induces the ordered partition $(A_1, A_2, ..., A_k)$. Given any A_i and any $x, y \in A_i$, we have $x\ R\ y$ and $y\ R\ x$, $\forall R \in \rho(T)$.*

---

[7]So for any $i \in N$, we have $P_i^{\{x,y\}} = \substack{x \\ y} \iff x\ P_i\ y$.

[8]For example the SWF α where $x\ \alpha(P)\ y\ \forall x, y \in A$ and $\forall P \in \Pi^N$ is a weak dictatorship but not Pareto optimal.

[9]See Footnote 3.

[10]we use the definition of "cycle" as stated by Peris and Subiza (1999).

[11]The top-cycle, introduced by Good (1971) and Schwartz (1972), has been explored in details. Moreover, Peris and Subiza (1999) extend this concept to weak tournaments. In their setting, as $C(X, T)$ is a cycle, $\nexists Y \subset C(X, T)$ with $y\ T^*\ x\ \forall y \in Y, \forall x \in C(X, T) \backslash Y$.

Proof. Take any $T \in \Theta$ which induces the ordered partition $(A_1, A_2, ..., A_k)$. Take any A_i, any for $x, y \in A_i$ and any $R \in \rho(T)$.If $\#A_i = 1$, then x and y coincide, hence $x\ R\ y$ and $y\ R\ x$ holds by the completeness of R. If $\#A_i = 2$, then $x\ T\ y$ and $y\ T\ x$ since A_i is a cycle, which implies $x\ R\ y$ and $y\ R\ x$ since $R \in \rho(T)$. We complete the proof by considering the case $\#A_i = k \geq 3$. Let $A_i = \{x_1, x_2, ..., x_k\}$. Suppose, without loss of generality, $x_1 R\ x_2$ and *not* $x_2\ R\ x_1$. This implies $x_1\ T^*\ x_2$, as $R \in \rho(T)$. Moreover, as A_i is a cycle, $\exists x \in A_i$ such that $x_2\ T\ x$. Let, without loss of generality, $x_2\ T\ x_3$. Thus $x_2\ R\ x_3$ holds by definition of ρ which implies $x_1 R\ x_3$ and *not* $x_3\ R\ x_1$ by the transitivity of R. Again by definition of ρ, we have $x_1\ T^* x_3$. As A_i is a cycle, $\exists j \in \{4, ..., k-1\}$ such that $x_3\ T\ x_j$. Suppose, without loss of generality, $j = 4$. So $x_3\ T\ x_4$, hence $x_3\ R\ x_4$, implying $x_1 R\ x_4$ and *not* $x_4\ R\ x_1$, which in turn implies $x_1 T^*\ x_4$. So, iteratively, $\forall i \in \{4, ..., k-1\}$, we have $x_i\ T\ x_{i+1}$,which implies $x_i\ R\ x_{i+1}$ and moreover $x_1 R\ x_{i+1}$ and *not* $x_{i+1}\ R\ x_1$. Hence, $x_1\ T^*\ x_{i+1}$. But as A_i is a cycle, we have $x_k\ T\ x_1$. So $x_k\ R\ x_1$ holds by definition of ρ. As we also have $x_i\ R\ x_{i+1}, \forall i \in \{1, ..., k-1\}$, $x_2\ R\ x_1$ holds by transitivity of R, which leads to a contradiction. Therefore, $x\ R\ y$ and $y\ R\ x$ for all $x, y \in A_i, \forall R \in \rho(T)$. ■

Thus for any $T \in \Theta$ which induces the ordered partition $(A_1, A_2, ..., A_k)$ and any $R \in \Re$, we have $R \in \rho(T)$ if and only if for any $x, y \in A$

(i) $x, y \in A_i$ for some $A_i \Longrightarrow xRy$ and yRx

and

(ii) $x \in A_i$ and $y \in A_j$ for some A_i, A_j with $i < j \Longrightarrow xRy$.

We now proceed towards characterizing the family of quasi IIA SWFs. Take any aggregation rule $f \in \Phi$ which satisfies IIA. By composing f with ρ, we get a *social welfare correspondence* $\rho \circ f : \Pi^N \longrightarrow 2^{\Re}$ which assigns to each $P \in \Pi^N$ a non-empty subset $\rho(f(P))$ of $\Re$. Clearly, every singleton-valued selection of $\rho \circ f$ is a SWF.[12] Let $\Sigma^f = \{\alpha : \Pi^N \to \Re \mid \alpha$ is a singleton-valued selection of $\rho \circ f\ \}$. We write $\Sigma = \cup_{f \in \Phi} \Sigma^f$. Interestingly, the class of quasi IIA SWFs coincides with Σ.

Theorem 3.2 *A SWF $\alpha : \Pi^N \to \Re$ is quasi IIA iff $\alpha \in \Sigma$.*

Proof. To establish the "only if" part, let $\alpha : \Pi^N \to \Re$ be a quasi IIA SWF. For any distinct $x, y \in A$, we define $f_{\{x,y\}} : \{\begin{smallmatrix} x \\ y \end{smallmatrix}, \begin{smallmatrix} y \\ x \end{smallmatrix}\}^N \to \{\begin{smallmatrix} x \\ y \end{smallmatrix}, \begin{smallmatrix} y \\ x \end{smallmatrix}, xy\}$ as follows: For any $r \in \{\begin{smallmatrix} x \\ y \end{smallmatrix}, \begin{smallmatrix} y \\ x \end{smallmatrix}\}^N$,

$$f_{\{x,y\}}(r) = \begin{cases} \begin{smallmatrix} x \\ y \end{smallmatrix} \text{ if } & x\ \alpha^*(P)\ y \text{ for some } P \in \Pi^N \text{ with } P^{\{x,y\}} = r \\ \begin{smallmatrix} y \\ x \end{smallmatrix} \text{ if } & y\ \alpha^*(P)\ x \text{ for some } P \in \Pi^N \text{ with } P^{\{x,y\}} = r \\ xy \text{ if } & x\ \alpha(P)\ y \text{ and } y\ \alpha(P)\ x \text{ for all } P \in \Pi^N \text{ with } P^{\{x,y\}} = r \end{cases}.$$

As α is quasi IIA, $f_{\{x,y\}}$is well-defined. Thus $f = \{f_{\{x,y\}}\} \in \Phi$. We now show $\alpha(P) \in \rho(f(P))$ $\forall P \in \Pi^N$. Take any $P \in \Pi^N$ and any distinct $x, y \in A$. First let $x\ f^*(P)\ y$. So $f_{\{x,y\}}(P^{\{x,y\}}) = \begin{smallmatrix} x \\ y \end{smallmatrix}$. By definition of $f_{\{x,y\}}$, we have $x\ \alpha^*(Q)\ y$ for some $Q \in \Pi^N$ with $Q^{\{x,y\}} = P^{\{x,y\}}$ which implies $x\ \alpha(P)\ y$ as α is quasi IIA. If $y\ f^*(P)\ x$, then one can similarly $y\ \alpha(P)\ x$. Now, let $x\ f(P)\ y$ and $y\ f(P)\ x$. So, $f_{\{x,y\}}(P^{\{x,y\}}) = xy$ which, by definition of $f_{\{x,y\}}$, implies $x\ \alpha(Q)\ y$ and $y\ \alpha(Q)\ x$ for all $Q \in \Pi^N$ with $Q^{\{x,y\}} = P^{\{x,y\}}$, hence $x\ \alpha(P)\ y$ and $y\ \alpha(P)\ x$. Thus, $x\ f(P)\ y \Longrightarrow x\ \alpha(P)\ y$ for any $x, y \in A$, establishing $\alpha(P) \in \rho(f(P))$.

To establish the "if" part, take any $\alpha \in \Sigma$. So there exists $f \in \Phi$ such that $\alpha(P) \in \rho(f(P))\ \forall P \in \Pi^N$. Suppose α is not quasi IIA. So, $\exists x, y \in A$ and $\exists P, Q \in \Pi^N$ with

[12] We say that $\alpha : \Pi^N \to \Re$ is a singleton-valued selection of $\rho \circ f$ iff $\alpha(P) \in \rho \circ f(P)\ \forall P \in \Pi^N$.

$P^{\{x,y\}} = Q^{\{x,y\}}$such that $x\ \alpha^*(P)\ y$ and $y\ \alpha^*(Q)\ x$. By the definition of ρ we have x $f^*(P)\ y$ and $y\ f^*(Q)\ x$ which implies $f_{\{x,y\}}(P^{\{x,y\}}) = \genfrac{}{}{0pt}{}{x}{y}$ and $f_{\{x,y\}}(Q^{\{x,y\}}) = \genfrac{}{}{0pt}{}{y}{x}$, giving a contradiction as $P^{\{x,y\}} = Q^{\{x,y\}}$, thus showing that α is quasi IIA. ■

By juxtaposing Theorems 3.1 and 3.2, one can conclude that removing the Pareto condition has a dramatic impact, as the class Σ of quasi IIA SWFs is fairly large and allows those where the decision power is not concentrated on a single individual. This positive result prevails when the following weak Pareto condition is imposed: A SWF α is *weakly Pareto optimal* iff given any distinct $x, y \in A$ and any $P \in \Pi^N$ with $x\ P_i\ y\ \forall i \in N$, we have $x\ \alpha(P)\ y$. An aggregation rule $f \in \Phi$ is *weakly Pareto optimal* iff for any $x, y \in A$ and any $r \in \{\genfrac{}{}{0pt}{}{x}{y}, \genfrac{}{}{0pt}{}{y}{x}\}^N$ with $r_i = \genfrac{}{}{0pt}{}{x}{y}\ \forall i \in N$, we have $f_{\{x,y\}}(r) \in \{\genfrac{}{}{0pt}{}{x}{y}, xy\}$. Let Φ^* stand for the set of weakly Pareto optimal and IIA aggregation rules and $\Sigma^* = \cup_{f\in\Phi^*}\Sigma^f$.

Theorem 3.3 *A SWF $\alpha : \Pi^N \to \Re$ is weakly Pareto optimal and quasi IIA iff $\alpha \in \Sigma^*$.*

Proof. To show the "only if" part, take any SWF $\alpha : \Pi^N \to \Re$ which is weakly Pareto optimal and quasi IIA. For any distinct $x, y \in A$, we define $f_{\{x,y\}} : \{\genfrac{}{}{0pt}{}{x}{y}, \genfrac{}{}{0pt}{}{y}{x}\}^N \to \{\genfrac{}{}{0pt}{}{x}{y}, \genfrac{}{}{0pt}{}{y}{x}, xy\}$ as follows: For any $r \in \{\genfrac{}{}{0pt}{}{x}{y}, \genfrac{}{}{0pt}{}{y}{x}\}^N$,

$$f_{\{x,y\}}(r) = \begin{cases} \genfrac{}{}{0pt}{}{x}{y} \text{ if } & x\ \alpha^*(P)\ y \text{ for some } P \in \Pi^N \text{ with } P^{\{x,y\}} = r \\ \genfrac{}{}{0pt}{}{y}{x} \text{ if } & y\ \alpha^*(P)\ x \text{ for some } P \in \Pi^N \text{ with } P^{\{x,y\}} = r \\ xy \text{ if } x\ \alpha(P)\ y \text{ and } y\ \alpha(P)\ x \text{ for all } P \in \Pi^N \text{ with } P^{\{x,y\}} = r \end{cases}.$$

As α is quasi IIA, $f_{\{x,y\}}$is well-defined. Thus $f = \{f_{\{x,y\}}\} \in \Phi$. Suppose, f is not weakly Pareto optimal. So, $\exists x, y \in A$ and $\exists P \in \Pi^N$ with $x\ P_i\ y\ \forall i \in N$ such that $y\ f^*(P)\ x$, implying $f_{\{x,y\}}(P^{\{x,y\}}) = \genfrac{}{}{0pt}{}{y}{x}$. By definition of $f_{\{x,y\}}$, we have $y\ \alpha^*(Q)\ x$ for some $Q \in \Pi^N$ with $Q^{\{x,y\}} = P^{\{x,y\}}$, contradicting that α is weakly Pareto optimal, which establishes $f = \{f_{\{x,y\}}\} \in \Phi^*$. We now show $\alpha(P) \in \rho(f(P))\ \forall P \in \Pi^N$. Take any $P \in \Pi^N$ and any distinct $x, y \in A$. First let $x\ f^*(P)\ y$. So $f_{\{x,y\}}(P^{\{x,y\}}) = \genfrac{}{}{0pt}{}{x}{y}$. By definition of $f_{\{x,y\}}$, we have $x\ \alpha^*(Q)\ y$ for some $Q \in \Pi^N$ with $Q^{\{x,y\}} = P^{\{x,y\}}$ which implies $x\ \alpha(P)\ y$ as α is quasi IIA. If $y\ f^*(P)\ x$, then one can similarly $y\ \alpha(P)\ x$. Now, let $x\ f(P)\ y$ and $y\ f(P)\ x$. So, $f_{\{x,y\}}(P^{\{x,y\}}) = xy$ which, by definition of $f_{\{x,y\}}$, implies $x\ \alpha(Q)\ y$ and $y\ \alpha(Q)\ x$ for all $Q \in \Pi^N$ with $Q^{\{x,y\}} = P^{\{x,y\}}$, hence $x\ \alpha(P)\ y$ and $y\ \alpha(P)\ x$. Thus, $x\ f(P)\ y \Longrightarrow x\ \alpha(P)$ y for any $x, y \in A$, establishing $\alpha(P) \in \rho(f(P))$.

To show the "if" part, take any $\alpha \in \Sigma^*$. So there exists $f \in \Phi^*$ such that $\alpha(P) \in \rho(f(P))$ $\forall P \in \Pi^N$. Take any distinct $x, y \in A$ and any $P \in \Pi^N$ with $x\ P_i\ y\ \forall i \in N$. By the weak Pareto optimality of f, we have $f_{\{x,y\}}(P^{\{x,y\}}) \in \{\genfrac{}{}{0pt}{}{x}{y}, xy\}$, hence $x\ f(P)\ y$, which implies x $\alpha(P)\ y$ by the definition of ρ. Thus, α is weakly Pareto optimal. The "if"part of Theorem 3.2 establishes that α is quasi IIA, completing the proof. ■

4 Concluding Remarks

Within the scope of the preference aggregation problem, we contribute to the understanding of the well-known tension between requiring the pairwise independence of the aggregation rule and the transitivity of the social preference. As Wilson (1972) shows, a SWF $\alpha : \Pi^N \to \Re$ is *non-imposed*[13] and IIA if and only if α is dictatorial or *antidictatorial*[14] or *null*[15]. Thus, aside from these, any aggregation rule which is IIA allows non-transitive social outcomes. In case these outcomes are rendered transitive according to one of the prescriptions made by ρ, we attain a SWF which fails IIA but satisfies quasi IIA. In fact, as Theorem 3.2 states, the class of quasi IIA SWFs coincides with those which can be attained through a selection made out of the social welfare correspondence obtained by the composition of a SWF that is IIA with ρ. This can be interpreted as a positive result, as the class of quasi IIA SWFs is fairly rich and not restricted to those where the decision power is concentrated on one individual. In fact, this class contains SWFs that are both anonymous and neutral.[16] Moreover, as Theorem 3.3 states, this positive result prevails when a weaker version of the Pareto condition is imposed. Thus, we can conclude that the transitivity of the social outcome can be achieved at a cost of reducing IIA to quasi IIA and compromising of the strenght of the Pareto condition - hence an escape from an impossibility of both the Arrow (1951) and Wilson (1972) type.

Another way of looking at the problem is to conceive it as determining the possible "stretchings" of the null rule (which is well-known to be IIA) without violating quasi-IIA. This angle of view advises caution about our optimism on escaping the Arrow/Wilson impossibilities, as this escape imposes indifference in social preference. So it is worth exploring "how far" quasi IIA SWFs are from the null rule. This exploration requires to ask for the minimization of the imposed social indifference. The answer is straightforward for a given aggregation rule $f \in \Phi$: Taking the transitive closure of the social preference is the selection of $\rho \circ f$ which minimizes the imposed social indifference.[17] Nevertheless, the choice of the (non-dictatorial) f that minimizes the imposed social indifference.remains as an interesting open question.[18]

Acknowledgments

We thank Goksel Asan, Nicholas Baigent, Donald Campbell, Semih Koray, Gilbert Laffond, Jean Laine, Ipek Ozkal-Sanver and Jack Stecher for their constructive suggestions. Our research is part of a project entitled "Social Perception - A Social Choice Perspective", supported by Istanbul Bilgi University Research Fund. Remzi Sanver acknowledges the support of the Turkish Academy of Sciences Distinguished Young Scientist Award Program (TUBA-GEBIP). Of course, the authors are responsible from all possible errors.

[13] $\alpha : \Pi^N \to \Re$ is *non-imposed* iff for any $x, y \in A$, there exists $P \in \Pi^N$ with $x\ \alpha(P)\ y$.

[14] α is *anti-dictatorial* iff $\exists i \in N$ such that $x\ P_i\ y$ implies $y\ \alpha^*(P)\ x\ \forall P \in \Pi^N, \forall x, y \in A$.

[15] $\alpha : \Pi^N \to \Re$ is *null* iff $x\ \alpha(P)\ y\ \forall x, y \in A$ and $\forall P \in \Pi^N$.

[16] As a matter of fact, the SWF in Example 2 of Campbell and Kelly (2000b), which shows the failure of Theorem 3.1 for $\#A = 3$, belongs to this class.

[17] By "taking the transitive closure", we mean to replace cycles with indifference classes. Formally speaking, writing $(A_1, A_2, ..., A_k)$ for the ordered partition induced by $f(P) \in \Theta$ at $P \in \Pi^N$, take $\alpha(P) \in \rho(f(P))$ where $x\ \alpha^*(P)\ y\ \forall x \in A_i$ and $\forall y \in A_j$ with $i < j$. One can see Sen (1986) for a general discussion of the "closure methods".

[18] We conjecture, by relying on Dasgupta and Maskin (2008), that this will be the pairwise majority rule.

5 References

Arrow, K. J. (1951) *Social Choice and Individual Values*, John Wiley, New York.

Baigent, N. (1987) "Twitching weak dictators", *Journal of Economics*, Vol. 47, No. 4: 407 - 411.

Barberà, S. (2003), "A theorem on preference aggregation", WP166 CREA-Barcelona Economics.

Blau, J. H. (1971) "Arrow's Theorem with weak independence", *Economica*, Vol. 38, No. 152: 413 - 420.

Campbell, D. E. (1976) "Democratic preference functions", *Journal of Economic Theory*, 12: 259 - 272.

Campbell, D. E. and J. S. Kelly (2000a) "Information and preference aggregation", *Social Choice and Welfare*, 17: 3 - 24.

Campbell, D. E. and J. S. Kelly (2000b) "Weak independence and veto power", *Economics Letters*, 66: 183 - 189.

Campbell, D. E. and J. S. Kelly (2007) "Social welfare functions that satisfy Pareto, anonymity and neutrality but not IIA", *Social Choice and Welfare*, 29: 69 - 82.

Dasgupta, P. and E. Maskin (2008), "On the robustness of majority rule", *Journal of the European Economic Association* (forthcoming).

Denicolo, V. (1998) "Independent decisiveness and the Arrow Theorem", *Social Choice and Welfare*, 15: 563 - 566.

Good, I. J. (1971) "A note on Condorcet sets", *Public Choice*, 10: 97 - 101.

Ozdemir, U. and R. Sanver (2007) "Dictatorial domains in preference aggregation", *Social Choice and Welfare*, 28: 61 - 76.

Peris, J. E. and B. Subiza (1999) "Condorcet choice correspondences for weak tournaments", *Social Choice and Welfare*, 16: 217 - 231.

Schwartz, J. (1972) "Rationality and the myth of maximum", *Nous*, Vol. 6, No. 2: 97 - 117.

Sen, A. (1986) "Social choice theory", in *Handbook of Mathematical Economics (Volume 3),* Eds. KJ Arrow and MD Intriligator, North-Holland.

Wilson, R. (1972) "Social Choice Theory without the Pareto principle", *Journal of Economic Theory*, 5: 478 - 486.

Ceyhun Coban
Department of Economics
Washington University in St. Louis
One Brookings Drive
St. Louis, MO 63130
USA

M. Remzi Sanver
Department of Economics
Istanbul Bilgi University
Inonu Cad. No. 28
Kustepe, 80310
Istanbul, Turkey
Email: sanver@bilgi.edu.tr

An Optimal Single-Winner Preferential Voting System Based on Game Theory

Ronald L. Rivest and Emily Shen

Abstract

We describe an *optimal* single-winner preferential voting system, called the "GT method" because of its use of symmetric two-person zero-sum game theory to determine the winner. Game theory is used not to describe voting as a multi-player game between voters, but rather to define when one voting system is better than another one. The cast ballots determine the payoff matrix, and optimal play corresponds to picking winners optimally.

The GT method is a special case of the "maximal lottery methods" proposed by Fishburn [14], when the preference strength between two candidates is measured by just the margin between them. We suggest that such methods have been somewhat underappreciated and deserve further study.

The GT system, essentially by definition, is *optimal*: no other preferential voting system can produce election outcomes that are preferred more by the voters, on the average, to those of the GT system. We also look at whether the GT system has several standard properties, such as monotonicity, Condorcet consistency, etc. We also briefly discuss a deterministic variant of GT, which we call GTD.

We present empirical data comparing GT and GTD against other voting systems on simulated data.

The GT system is not only theoretically interesting and optimal, but simple to use in practice. We feel that it can be recommended for practical use.

1 Introduction

Voting systems have a rich history and are still being vigorously researched. We refer the reader to surveys and texts, such as Börgers [1], Brams [2], Brams and Fishburn [3], Fishburn [13], Kelly [18], and Tideman [34], for overviews.

The purpose of this paper is to describe a preferential voting system, called the "GT method," to study its properties, and to compare it with some other well-known voting systems.

The GT method is a special case of the "maximal lottery methods" discussed by Fishburn [14] (who references Kreweras [19] as the first to mention them). A *lottery* assigns a probability to each candidate; a lottery method outputs such a lottery, and the election winner is chosen randomly according to those probabilities. Maximal lotteries are those that voters prefer at least as well as any single candidate or any other lottery. The preference strength between two lotteries is the expected value of a social evaluation function applied to the vote differential (margin) between candidates. The GT method has the identity function as the social evaluation function (i.e., the strength of the social preference between two candidates is the vote margin between them).

We suggest that such voting systems with probabilistic output have received insufficient attention, both in the literature and in practice, and that they are really the most natural resolution of the "Condorcet cycle" paradox that plagues preferential voting systems.

More generally, at a high level, the approach is based on a "metric" or "quantitative" approach to comparing two voting systems, which is a nice complement to the more usual "axiomatic" or "property-based" approach common in the literature; the metric approach enables a simple comparison of any two voting systems, given a distribution on profiles.

Finally, the GT method is easy to use in practice; we discuss some implementation details.

The contributions of this paper are as follows:

- We define "relative advantage" as a metric to compare two preferential voting systems.
- We define the GT method as the "optimal" preferential voting system with respect to relative advantage. This includes our proposal for resolving ambiguity when the optimal mixed strategy is not unique.
- We compare the GT method and various voting systems experimentally and show a ranking of these systems, relative to GT.
- We propose a deterministic variant of GT, called GTD, which performs nearly as well as GT, and may be more acceptable to those who object to randomized methods.

2 Preliminaries

Candidates and ballots We assume an election where n voters are to select a single winner from m alternatives ("candidates"). We restrict attention to *preferential voting systems*, where each ballot lists candidates in order of preference. We assume that all ballots are *full* (they list all candidates), but it is a simple extension to allow voters to submit truncated ballots, to write in candidates, or to express indifference between candidates (details omitted).

Profiles, preference and margin matrices, and margin graphs A collection C of (cast) ballots is called a *profile*. A profile is a multi-set; two ballots may list candidates in the same order.

A profile has an associated *preference matrix* N—the $m \times m$ matrix whose (x, y) entry is the number of ballots expressing a preference for candidate x over candidate y. Each entry is nonnegative, and $N(x, y) + N(y, x) = n$, since all ballots are assumed to be full.

It is also useful to work with the *margin matrix* M — the $m \times m$ matrix defined by $M(x, y) = N(x, y) - N(y, x)$, so that $M(x, y)$ is the margin of x over y—that is, the number of voters who prefer x over y minus the number of voters who prefer y over x. The matrix M is anti-symmetric with diagonal 0; for all x, y we have: $M(x, y) = -M(y, x)$.

From the margin matrix M we can construct a directed weighted *margin graph* G whose vertices are the candidates and where there is an edge from x to y weighted $M(x, y)$ whenever $M(x, y) > 0$. If $M(x, y) = M(y, x) = 0$ then voters are, on the whole, indifferent between x and y, and there is no edge between x and y.

Voting system – social choice function A voting system provides a *social choice function* that takes as input a profile of cast ballots and produces as output the name of the election winner. (In some systems the output may be a set of winners.) The social choice function may be *deterministic* or *randomized*. While most but not all voting systems in the literature are deterministic, the GT system is randomized. We also describe a deterministic variant, GTD, of the GT system.

3 Generalized Ties

A *Condorcet winner* is a candidate x who beats every other candidate in a pairwise comparison: for every other candidate y, more voters prefer x to y than prefer y to x. Thus, the margin matrix M has only positive entries in every off-diagonal position of row x. Equivalently, for each other candidate y, the margin graph contains a directed edge from x to y.

If there is no Condorcet winner, we say that there is a "*generalized tie*," since for every candidate x there exists some other candidate y whom voters like at least as much as x.

The interesting question is then: *When there is a generalized tie, how should one do the "tie-breaking" to pick a single winner?*

4 Breaking Ties Using a Randomized Method

We feel strongly that the best way of breaking a generalized tie is to use an appropriate randomized method. Of course, when there is a clear winner (by which we mean a Condorcet winner) then a randomized method is not needed. A randomized method is only appropriate when a tie needs to be broken.

Academic literature on voting systems has often eschewed proposals having a randomized component. For example, Myerson [26, p. 15] says,

> "Randomization confronts democratic theory with the same difficulty as multiple equilibria, however. In both cases, the social choice ultimately depends on factors that are unrelated to the individual voters' preferences (private randomizing factors in one case, public focal factors in the other). As Riker (1982) has emphasized, such dependence on extraneous factors implies that the outcome chosen by a democratic process cannot be characterized as a pure expression of the voters' will."

We would argue that Myerson and Riker have it backwards, since, as we shall see, voting systems can do better at implementing the voters' will if they are randomized.

Arbitrary deterministic tie-breaking rules, such as picking the candidate whose name appears first in alphabetical order, are clearly unfair. And, while much work has gone into devising clever voting systems that break generalized ties in apparently plausible but deterministic manners, the result is nonetheless arguably unfair to some candidates.

The strongest reason for using a randomized tie-breaking method is that for any deterministic voting system there is another voting system whose outcomes are preferred by voters on the average, while there exist randomized voting systems which are not so dominated by another system. This is effectively just a restatement of the minimax theorem, due to von Neumann, that optimal strategies in two-person zero-sum games may need to be randomized.

It is not a new idea to have a voting system that uses randomization, either in theory or in practice. Using a randomized method is in fact a common and sensible way of breaking ties.

Several recent elections have used randomized methods to break ties. In June, 2009, when the city of Cave Creek, Arizona had a tie between two candidates for a city council seat, the two candidates drew cards from a shuffled deck to determine the winner[1]. In November, 2009, the mayor of Wendell, Idaho, was determined by a coin toss, when the challenger and the incumbent were tied. In February, 2010, in Sealy, Texas, dice were used to resolve a tied election for city council membership.

Several previous voting system proposals use randomization to determine the outcome. For example, the "random dictator" voting system [15, 32] picks a random ballot, and uses it to name the winner. This method always uses randomization, not just for tie-breaking. Gibbard [15] proves that if a system is strategy-proof (and satisfies certain natural conditions), then it must be the random dictator method.

[1] "Election at a Draw, Arizona Town Cuts a Deck," NY Times, June 17, 2009.

Sewell et al. [32] propose a randomized voting system based on maximum entropy considerations; this is, however, a social welfare function (it produces a complete ordering, not just a single winner), not a social choice function. Potthoff [27] proposes a randomized method for the case of a three-candidate election with a majority cycle. Laffond et al. [21] propose a randomized method based on game theory for parties to pick platform issues, a situation attributed by Shubik [33] to Downs [8].

Other voting systems, such as the Schulze method [31], use randomization as a final tie-breaker.

5 Optimal Preferential Voting Systems

How should one compare a voting system P against another voting system Q? Here P and Q are (possibly randomized) social choice functions that each take a profile C of cast ballots and produce an election outcome or winner, $P(C)$ or $Q(C)$.

There is a long list of well-studied properties of voting systems, such as monotonicity, consistency, etc.; such studies exemplify the "axiomatic" approach to voting systems. One can certainly ask whether a voting system has these desirable properties. The inference is usually that a system with more desirable properties is the better system. But this approach can sometimes give rather conflicting and inconclusive advice.

Here is a more direct approach:

> A voting system P is said to be better than a voting system Q if voters tend to prefer the outcome of P to the outcome of Q.

How can one make this appealing intuition precise?

Let $\mathcal{C}$ be an assumed probability distribution on the profiles of cast ballots. (The details of $\mathcal{C}$ will turn out to be not so important, since GT is optimal on each profile C separately.)

Suppose we choose a profile C of cast ballots according to the distribution $\mathcal{C}$ and then play a game $G_C(P,Q)$ between P and Q as follows:

- P and Q compute respective election outcomes $x = P(C)$ and $y = Q(C)$.
- The systems are scored as follows: P wins $N(x,y)$ points, and Q wins $N(y,x)$ points.

Note that the net number of points gained by P, relative to the number of points gained by Q, is just the margin $M(x,y) = N(x,y) - N(y,x)$; more voters prefer P's outcome to Q's outcome than the reverse if $M(x,y) > 0$.

Definition 5.1 *We say that the* relative advantage *of voting system P over voting system Q, denoted $\mathbf{Adv}_{\mathcal{C}}(P,Q)$, with respect to distribution $\mathcal{C}$ on profiles, is*

$$\mathbf{Adv}_{\mathcal{C}}(P,Q) = E_{\mathcal{C}}(M(x,y)/\,|C|) \tag{1}$$

where $x = P(C)$ and $y = Q(C)$, where $E_{\mathcal{C}}$ denotes expectation over profiles C chosen according to the distribution $\mathcal{C}$ and with respect to any randomization within P and Q, and where $0/0$ is understood to equal 0 if $|C| = 0$. When $\mathcal{C}$ has all of its support on a single profile C, we write $\mathbf{Adv}_C(P,Q)$.

Definition 5.2 *We say that voting system P is as good as or better than voting system Q (with respect to distribution $\mathcal{C}$ on profiles), if $\mathbf{Adv}_{\mathcal{C}}(P,Q) \geq 0$.*

Definition 5.3 *We say that voting system P is* optimal *if it is as good as or better than every other voting system for any distribution $\mathcal{C}$ on profiles—equivalently, if for every profile C and for every voting system Q we have $\mathbf{Adv}_C(P,Q) \geq 0$.*

Intuitively, P will win more points than Q, on the average, according to the extent that voters prefer P's outcomes to Q's outcomes. If P's outcomes tend to be preferred, then P should be considered to be the better voting system. And if P is as good as or better than any other voting system, for any distribution on profiles, then P is optimal.

Note that if P is as good as or better than Q on every distribution $\mathcal{C}$ on profiles, then P must be as good or better than Q on each particular profile C, and vice versa, so the details of distribution $\mathcal{C}$ don't matter.

6 Game Theory

We now describe how to construct an optimal voting system using game theory.

In the game $G_C(P, Q)$, the margin $M(x, y)$ is the "payoff" received by P from Q when P picks x, and Q picks y, as the winner for the election with profile C. The comparison of two voting systems reduces to considering them as players in a distribution on two-person zero-sum games—one such game for each profile C.

The theory of two-person zero-sum games is long-studied and well understood, and optimal play is well-defined. See, for example, the excellent survey article by Raghavan [28].

The expected payoff for P, when P chooses candidate x with probability p_x and when Q independently chooses candidate y with probability q_y is:

$$\sum_x \sum_y p_x q_y M(x, y) \ . \tag{2}$$

An optimal strategy depends on the margin matrix M. When there is a Condorcet winner, the optimal strategy will always pick the Condorcet winner as the election winner. When there is no Condorcet winner, there is a generalized tie, and the optimal strategy is a mixed strategy. Computing the optimal mixed strategy is not hard; see Section 7. Playing this optimal mixed strategy yields an optimal preferential voting system—no other voting system can produce election outcomes that are preferred more by the voters, on average.

We denote by $\text{supp}(GT(C))$ the set of candidates with nonzero probability in the optimal mixed strategy for the game associated with profile C. (If there is not a unique optimal mixed strategy, GT uses the most "balanced" optimal mixed strategy, as described in Section 7.) Intuitively, $\text{supp}(GT(C))$ is the set of "potential winners" for the election with profile C for the GT voting system. If there is a Condorcet winner x, then $\text{supp}(GT(C)) = \{x\}$; otherwise, the GT winner is chosen randomly from $\text{supp}(GT(C))$ according to the optimal mixed strategy probabilities.

7 Computing Optimal Mixed Strategies

One can solve a two-person zero-sum symmetric game with $m \times m$ payoff matrix M using a simple reduction to linear programming. Each solution to the linear program provides an optimal mixed strategy for the game. (See Raghavan [28, Problem A, page 740] for details.)

When ballots are full and the number of voters is odd, the optimal mixed strategy p^* is uniquely defined (see Laffond et al. [22]). There are other situations for which there is a unique optimal mixed strategy. With a large number of voters, one would expect the optimal mixed strategy to be unique.

In the case when there is not a unique optimal mixed strategy, we propose that GT picks the unique optimal mixed strategy that minimizes the sum of squares $\sum_i p_i^2$; this strategy can be computed easily with standard quadratic programming packages. This approach then gives a well-defined lottery as output, and treats candidates symmetrically.

8 Selecting the Winner

As we have seen, the GT voting system comprises the following steps:

1. [**Margins**] Compute the margin matrix M from the profile C of cast ballots.
2. [**Optimal mixed strategy**] Determine the optimal mixed strategy p^* for the two-person zero-sum game with payoff matrix M.
3. [**Winner selection**] Select the election winner by a randomized method in accordance with the probability distribution p^*. (If there is a Condorcet winner x, then $p^*(x) = 1$ and this step is trivial.)

There are of course details that must be taken care of properly with using a randomized method to select a winner; these details are very similar to those that arise when generating suitable random numbers of post-election audits; see Cordero et al. [6].

GTD—A Deterministic Variant of GT We now describe a deterministic variant of the GT voting system, which we call GTD. The optimal mixed strategy is computed as in GT, but the winner selection then proceeds in a deterministic manner.

Instead of randomly picking a candidate according to this probability distribution, GTD chooses the candidate with the maximum probability in this optimal mixed strategy. (If there is more than one candidate with the maximum probability in the optimal mixed strategy, then the one with the least name alphabetically is chosen.)

The GTD method does not require any randomness—it is a deterministic social choice function. We expect that in practice it would perform as well as the GT method. However, since GTD is deterministic, one cannot prove that it is optimal.

9 Properties of the GT Voting System

Although our focus is on comparing voting systems using "relative advantage" instead of an axiomatic approach, we briefly consider how GT fares with respect to some standard properties.

Optimality. Optimality is perhaps the most important property of the GT voting system. No preferential voting system can produce election outcomes that are preferred more by voters to those of the GT system, on average.

Condorcet winner and loser criteria. Fishburn [14] proves that maximal lotteries satisfy the strong Condorcet property: If the candidates can be partitioned into nonempty subsets A and B such that, for all $a \in A$ and all $b \in B$, more voters prefer a to b than b to a, then the winner will be a candidate in A. This result implies in particular that the GT method will always elect a Condorcet winner, if one exists, and will never elect a Condorcet loser, if one exists. The Condorcet criterion implies the majority criterion. However, as Schulze [31] notes, the Condorcet criterion is incompatible with other desired criteria including consistency [36], participation [25], later-no-help, and later-no-harm [35][2].

Pareto optimality. A voting system satisfies *Pareto optimality* if whenever there exist two candidates x and y such that no voter prefers candidate y to x, and at least one voter prefers x to y, then the voting system never elects y. Fishburn [14] proves that maximal lottery methods satisfy Pareto optimality (and thus GT does).

[2] These results are for deterministic voting systems. The notion of consistency, for example, needs to be redefined for probabilistic voting systems.

Monotonicity. A voting system satisfies *monotonicity* if, if a voter raises a candidate x on her ballot without changing the order of other candidates, then the probability that the voting system elects x does not decrease. The GT system is not monotonic. This can be seen by analyzing the optimal mixed strategy probabilities of the simplest generalized tie, whose margin graph is a three-cycle. (See Fishburn [14] and Kaplansky [17, p. 479].)

Independence of clones. A voting system satisfies the *independence of clones* property if replacing an existing candidate `A` with a set of clones does not change the winning probability for any candidates other than `A`. (Schulze [31, p. 141] notes some of the subtleties in the definition of this property, especially when `A` is already in some sense tied with other candidates.) The GT voting system satisfies independence of clones, for a careful definition of the property. (See the full version of this paper for details.)

Reversal symmetry. A voting system satisfies *reversal symmetry* (see Saari [30]) if it never elects the same candidate as the winner when each voter's preferences are reversed. The GT voting system satisfies reversal symmetry in cases where the GT support consists of a unique candidate, which may be the only cases when it makes sense to consider reversal symmetry.

Manipulability. Our definition of relative advantage allows one to compare two voting systems based on which voting system's outcomes are preferred more by the voters, according to voter preferences as expressed in their ballots. We do not take into consideration whether voters might be voting strategically.

Unfortunately, Gibbard's [15] characterization of strategy-proof randomized voting systems tells us that GT is not strategy-proof. However, the computational hardness of manipulating GT can be studied, although we have no reason to believe that GT is computationally easier or harder to manipulate than other preferential voting systems.

10 Empirical Comparison with Other Voting Systems

The approach we are recommending allows one to compare any two voting systems P, Q on a given distribution $\mathcal{C}$ of profiles, by computing the relative advantage $\mathbf{Adv}_{\mathcal{C}}(P, Q)$ of one system over the other.

We compared seven voting systems: plurality, IRV, Borda, minimax, the Schulze method [31], GTD, and GT. We used the margins variant of minimax and the "winning votes" variant of the Schulze method.

We randomly generated 10,000 profiles for $m = 5$ candidates, as follows. Each profile had $n = 100$ full ballots. Each candidate and each voter was randomly assigned a point on the unit sphere—think of these points as modeling candidates' and voters' locations on Earth. A voter then lists candidates in order of increasing distance from her location. With this "planetary" distribution, about 64.3% of the profiles had a Condorcet winner, and about 77.1% of the 10,000 simulated elections had a unique optimal mixed strategy.

We also tried our experiments under the "impartial culture" distribution (i.e., the uniform distribution). However, under this distribution there were Condorcet winners almost all (about 93%) of the time, so we chose another distribution.

The code we used, and detailed output data, is available at `http://people.csail.mit.edu/rivest/gt` .

Figure 1 gives the cumulative "point advantage" of each of the seven voting systems against each other in our experiment. For example, the "16380" entry in row "Schulze," column "IRV" means that in an average election, the net number of voters preferring the

	plurality	IRV	Borda	minimax	Schulze	GTD	GT
plurality	0	-23740	-31058	-32030	-32128	-32390	-29978
IRV	23740	0	-14148	-16296	-16380	-15892	-13872
Borda	31058	14148	0	-4546	-4654	-5324	-2522
minimax	32030	16296	4546	0	-58	-1436	-174
Schulze	32128	16380	4654	58	0	-1402	-76
GTD	32390	15892	5324	1436	1402	0	10
GT	29978	13872	2522	174	76	-10	0

Figure 1: Cumulative "point advantages" for our main experiment. Row X column Y shows the sum, over 10,000 simulated elections with 100 votes each, of the margin of X's winner over Y's winner. For example, the entry `13872` in row `GT`, column `IRV` means that on average for a random election from our distribution $\mathcal{C}$ on profiles, 1.3872% more of the electorate prefers the GT outcome to the IRV outcome than the reverse; that is, $\mathbf{Adv}_{\mathcal{C}}(GT, IRV) = 1.3872\%$.

Schulze outcome to the IRV outcome is about 1.6380 voters (i.e., 1.6380% of the electorate). That is, $\mathbf{Adv}_{\mathcal{C}}(\text{Schulze}, \text{IRV}) \approx 0.016380$.

With this distribution on profiles, there is a clear improvement in quality of output (as measured by relative advantage compared to GT) as one goes from plurality to IRV to Borda to minimax to Schulze. GT and GTD are perfect by definition in this metric, but Schulze is amazingly close. Although GTD and GT are by definition in a dead heat against each other, GTD appears to be a better competitor against the other systems than GT.

Note that when comparing GT with another voting system, there is no expected net point gain for GT if the other system picks a candidate that is in supp$(GT(C))$. Candidates in supp$(GT(C))$ have the property that playing any one of them has an expected payoff equal to zero (the value of the game) against GT. If the other system plays a candidate outside of supp$(GT(C))$, GT will have an expected net point gain and the other system will have an expected loss.

Figure 2 illustrates the number of times each pair of voting systems produced results that "agree with" each other. The column "GTS" refers to the support of GT; a method "agrees with" GTS if it produces an output that is in the support of GT. In our view, level of agreement with the support of GT is an interesting measure of the quality of the results produced by each voting system. Plurality does quite poorly (agreeing with GTS only 55.15% of the time), as does IRV (72.99%), but minimax (99.15%) and the Schulze method (99.51%) have nearly perfect agreement with the support of GT.

Thus, one can perhaps view the evolution of voting system proposals as a continuing effort to identify candidates that are in the support for the optimal mixed strategy for the associated two-person game, without quite realizing that this is the natural goal. That is, voting systems should be (at the minimum) returning winners that are in supp$(GT(C))$, the set of potential winners for the GT voting system. To do otherwise does not serve the voters as well as can be done. However, since determining the support for the optimal mixed strategy intrinsically involves linear programming, this computation is non-trivial, so we see a variety of quite complex voting system proposals in the literature, which are, in this view, just approximate computations for (a member of) supp$(GT(C))$.

11 Practical Considerations

We believe that the GT voting system is suitable for practical use.

	plurality	IRV	Borda	minimax	Schulze	GTD	GT	GTS
plurality	10000	5557	4107	4356	4366	4335	4262	5515
IRV	5557	10000	5584	6047	6048	5999	5802	7299
Borda	4107	5584	10000	7854	7874	7813	7193	8913
minimax	4356	6047	7854	10000	9953	8869	8232	9915
Schulze	4366	6048	7874	9953	10000	8895	8246	9951
GTD	4335	5999	7813	8869	8895	10000	8377	10000
GT	4262	5802	7193	8232	8246	8377	10000	10000
GTS	5515	7299	8913	9915	9951	10000	10000	10000

Figure 2: Agreement between pairs of voting systems. Row X column Y gives the number of times that method X produced an outcome that agreed with the outcome of method Y, in our 10,000 trials. Here the "GTS method" refers to the support of GT, and a method "agrees with" GTS if it produces an outcome that is in the support of GT. In our view, frequency of agreement with GTS (producing outcomes in the support of GT) is an important measure of the quality of a preferential voting system.

Since the GT voting system depends only on the pairwise preference matrix N, ballot information can be easily aggregated at the precinct level and the results compactly transmitted to central election headquarters for final tabulation; the number of data items that need to be transmitted is only $O(m^2)$, which is much better than for, say, IRV.

Perhaps the only negative aspects with respect to using GT in practice are that (1) its game-theoretic rationale may be confusing to some voters and election officials, (2) it is a randomized method, and may require dice-rolling or other randomized devices in the case of generalized ties, and (3) it is not so clear how to efficiently audit a GT election. (The last aspect is common to many preferential voting systems).

12 Other Related Work

Fishburn [12] gives an excellent overview of voting systems with the Condorcet property.

The idea of using a two-player zero-sum game based on a payoff matrix derived from a profile of ballots is not new; there are several papers that study this and related situations.

Laffond et al. [20] introduce the concept of a "bipartisan set," which is the support of the optimal mixed strategy of a two-player "tournament game." (A tournament game is based on an unweighted complete directed graph (a tournament) where each player picks a vertex, and the player picking x wins one point from the player picking y if there is an edge from x to y.) They show that any tournament game has a unique optimal mixed strategy, and study the properties of its support.

Laffond et al. [21] propose the use of optimal mixed strategies of a zero-sum two-player game in the context of tournament games and "plurality games". (A plurality game is the weighted version of a tournament game and corresponds to the voting situation we consider (assuming no margins are zero); the weight of an edge from x to y is the margin $M(x, y)$.) However, their focus is on the way political parties choose platform issues, whereas our focus is on "competition" between voting systems rather than between political parties. Our work should nonetheless be viewed as further explorations along the directions they propose.

Le Breton [4, p. 190] proves a general version of Laffond et al.'s [20] earlier result, showing that if all edges satisfy certain congruence conditions, then the weighted tournament game has a unique optimal mixed strategy.

Laslier [23] studies the "essential set" (the support of the optimal mixed strategies in a symmetric two-party electoral competition game) with respect to the independence of clones axiom.

Duggan and Le Breton [9] study the "minimal covering set" of a tournament (proposed by Dutta [10] as a choice function on tournaments), and show that it is the same as Shapley's notion of a "weak saddle" for the corresponding tournament game.

De Donder et al. [7] consider various solution concepts for tournament and weighted tournament games and make set-theoretic comparisons between the corresponding social choice functions.

Michael and Quint [24] provide further results characterizing when there exists a unique optimal strategy in tournament and weighted tournament games.

Dutta et al. [11] introduce "comparison functions," which correspond to general skew-symmetric matrices, as a framework for generalizing choice functions on tournaments.

13 Open Problems

There are many aspects of the GT method, and of probabilistic voting systems in general, that deserve further study. Here are a few such open questions:

- For which pairs of voting systems P and Q, and for which distributions $\mathcal{C}$ on profiles, can $\mathbf{Adv}_{\mathcal{C}}(P, Q)$ be analytically determined? Can one show analytically that GTD performs better than GT against some well-known voting system?
- Can one lower bound (for some assumed distribution $\mathcal{C}$ on profiles) the penalty paid for being deterministic, consistent, or monotonic (i.e., in terms of the advantage of GT over systems with the given property)?
- How sensitive are the output probabilities of GT to the input votes? More generally, how resistant is GT to manipulation, for various notions of manipulation of probabilistic voting systems (e.g., that of [5])?
- Is it possible to modify the Schulze method in a straightforward manner so that it always chooses a winner in the support of GT, while retaining its deterministic character and its other desirable properties?
- To what extent would changing the social evaluation function (see Fishburn [14]) change the perceived relative quality of various voting systems (e.g., via simulation results)?

14 Conclusions

We have described the GT voting system for the classic problem of determining the winner of a single-winner election based on voters' preferences expressed as (full or partial) rank-order listings of candidates.

The GT scheme is arguably optimal among preferential voting systems, in the sense that no other voting system P can produce election outcomes that on the average are preferred by voters to those of GT. We feel that optimality is an important criterion for voting systems.

We believe that the GT voting system is suitable for practical use, when preferential voting is desired. When there is a clear (Condorcet) winner, GT elects that winner. When there is no Condorcet winner, GT produces a "best" set of probabilities that can be used in a tie-breaking ceremony. If one is to use preferential ballots, the GT system can be recommended.

Since the GT system shares some potentially confusing properties, such as non-monotonicity, with many other preferential voting systems, election authorities might reasonably consider alternatives to the GT system, such as a non-optimal but monotonic preferential voting system like the Schulze method, or non-preferential voting systems such as approval voting or range voting.

However, we feel that the optimality property of GT makes it worthy of serious consideration when preferential ballots are to be used.

Acknowledgments

We thank the Stanford CS Department and Microsoft Research for hosting the first author on portions of his sabbatical when some of this research was performed. The second author gratefully acknowledges support from a Bell Labs Graduate Research Fellowship. We also thank Josh Benaloh, Steven Brams, James Heather, Gilbert Laffond, Jean-François Laslier, Richard Potthoff, Ariel Procaccia, Peter Ryan, Rob Schapire, Warren Smith, and Vanessa Teague for helpful feedback.

References

[1] C. Börgers. *Mathematics of Social Choice: Voting, Compensation, and Division.* SIAM, 2010.

[2] S. J. Brams. *Mathematics and Democracy.* Princeton, 2008.

[3] S. J. Brams and P. C. Fishburn. Voting procedures. In K. Arrow, A. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, pages 173–236. Elsevier, 2002.

[4] M. Le Breton. On the uniqueness of equilibrium in symmetric two-person zero-sum games with integer payoffs. *Économie Publique*, 2005/2(17):187–195, 2005.

[5] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 781–788, 2003.

[6] A. Cordero, D. Wagner, and D. Dill. The role of dice in election audits. In *Proc. IAVOSS Workshop on Trustworthy Elections (WOTE 2006)*, 2006.

[7] P. De Donder, M. Le Breton, and M. Truchon. Choosing from a weighted tournament. *Math. Social Sciences*, 40(1):85–109, Jul 2000.

[8] Anthony Downs. *An Economic Theory of Democracy.* Harper and Row, 1957.

[9] J. Duggan and M. Le Breton. Dutta's minimal covering sets and Shapley's saddles. *J. Econ. Theory*, 70:257–265, 1996.

[10] B. Dutta. Covering sets and a new Condorcet choice correspondence. *J. Econ. Theory*, 44:63 – 80, 1988.

[11] B. Dutta and J.-F. Laslier. Comparison functions and choice correspondences. *Soc. Choice Welfare*, 16:513–532, 1999.

[12] P. C. Fishburn. Condorcet social choice functions. *SIAM J. Appl. Math.*, 33(3):469–489, Nov 1977.

[13] P. C. Fishburn. Dimensions of election procedures: Analyses and comparisons. *Theory and Decision*, 15(4):371–397, Dec 1983.

[14] P. C. Fishburn. Probabilistic social choice based on simple voting comparisons. *Rev. Economic Studies*, 51(4):683–692, Oct 1984.

[15] A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–681, Apr 1977.

[16] A. Gibbard. Straightforwardness of game forms with lotteries as outcomes. *Econometrica*, 46(3):595–614, May 1978.

[17] Irving Kaplansky. A contribution to Von Neumann's theory of games. *Annals. Math. (Second Series)*, 46(3):474–479, Jul 1945.

[18] J. S. Kelly. Social choice bibliography. *Soc. Choice Welfare*, 8:97–169, 1991.

[19] G. Kreweras. Aggregation of preference orderings. In S. Sternberg, editor, *Math. and Soc. Sciences I*, pages 73–79. Moulton, 1965.

[20] G. Laffond, J.-F. Laslier, and M. Le Breton. The bipartisan set of a tournament game. *Games and Economic Behavior*, 5:182–201, 1993.

[21] G. Laffond, J.-F. Laslier, and M. Le Breton. Social choice mediators. *American Economic Review*, 84(2):448–453, May 1994. Proc. 106th Ann. Mtg. Amer. Economic Assn.

[22] G. Laffond, J.-F. Laslier, and M. Le Breton. A theorem on symmetric two-person zero-sum games. *J. Economic Theory*, 72:426–431, 1997.

[23] J.-F. Laslier. Aggregation of preferences with a variable set of alternatives. *Social Choice and Welfare*, 17:269–282, 2000.

[24] T. S. Michael and T. Quint. Optimal strategies for node selection games on oriented graphs: Skew matrices and symmetric games. *Linear Algebra and its Applications*, 412:77–92, 2006.

[25] H. Moulin. Condorcet's principle implies the no show paradox. *J. of Economic Theory*, 45:53–64, 1988.

[26] R. B. Myerson. Fundamentals of social choice theory, Sep 1996. Discussion paper 1162. Center for Math. Studies in Economics and Mgt. Science. Northwestern Univ., Evanston, IL.

[27] R. F. Potthoff. The problem of the three-way election. *Essays in Probability and Statistics*, pages 603–620, 1969.

[28] T. E. S. Raghavan. Zero-sum two-person games. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory*, volume 2, pages 735–759. Elsevier, 1994.

[29] W. H. Riker. *Liberalism Against Populism*. Freeman, 1982.

[30] D. G. Saari. *Geometry of Voting*. Springer-Verlag, 1994.

[31] M. Schulze. A new monotonic, clone-independent, reversal-symmetric, and condorcet-consistent single-winner election method, Aug 2009. `http://m-schulze.webhop.net/schulze1.pdf`.

[32] R. Sewell, D. MacKay, and I. McLean. Probabilistic electoral methods, representative probability, and maximum entropy. *Voting Matters*, 26:16–38, Jan 2009. `http://www.votingmatters.org.uk/ISSUE26/I26P3.pdf`.

[33] M. Shubik. Voting, or a price system in a competitive market structure. *The American Political Science Review*, 64(1):179–181, Mar 1970.

[34] N. Tideman. *Collective Decisions and Voting: the potential for public choice*. Ashgate, 2006.

[35] D. Woodall. Monotonicity of single-seat preferential election rules. *Discrete Applied Math.*, 77:81–98, 1997.

[36] H. P. Young. Social choice scoring functions. *SIAM J. Appl. Math.*, 28(4):824–838, Jun 1975.

Ronald L. Rivest (Email: `rivest@mit.edu`)
MIT Computer Science and Artificial Intelligence Laboratory
Room 32G-692, 32 Vassar Street
Cambridge, MA 02139 USA

Emily Shen (Email: `eshen@csail.mit.edu`)
MIT Computer Science and Artificial Intelligence Laboratory
Room 32G-626, 32 Vassar Street
Cambridge, MA 02139 USA

Convergence to Equilibria in Plurality Voting

Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nicholas R. Jennings

Abstract

Multi-agent decision problems, in which independent agents have to agree on a joint plan of action or allocation of resources, are central to AI. In such situations, agents' individual preferences over available alternatives may vary, and they may try to reconcile these differences by voting. Based on the fact that agents may have incentives to vote strategically and misreport their real preferences, a number of recent papers have explored different possibilities for avoiding or eliminating such manipulations. In contrast to most prior work, this paper focuses on convergence of strategic behavior to a decision from which no voter will want to deviate.
We consider scenarios where voters cannot coordinate their actions, but are allowed to change their vote after observing the current outcome. We focus on the Plurality voting rule, and study the conditions under which this iterative game is guaranteed to converge to a Nash equilibrium (i.e., to a decision that is stable against further unilateral manipulations).
We show for the first time how convergence depends on the exact attributes of the game, such as the tie-breaking scheme, and on assumptions regarding agents' weights and strategies.

1 Introduction

The notion of strategic voting has been highlighted in research on Social Choice as crucial to understanding the relationship between preferences of a population, and the final outcome of elections. The most widely used voting rule is the Plurality rule, in which each voter has one vote and the winner is the candidate who received the highest number of votes. While it is known that no reasonable voting rule is completely immune to strategic behavior, Plurality has been shown to be particularly susceptible, both in theory and in practice [12, 8]. This makes the analysis of any election campaign—even one where the simple Plurality rule is used—a challenging task. As voters may speculate and counter-speculate, it would be beneficial to have formal tools that would help us understand (and perhaps predict) the final outcome.

Natural tools for this task include the well-studied solution concepts developed for normal form games. While voting games are not commonly presented in this way, several natural formulations have been proposed. Moreover, such formulations are extremely simple in Plurality voting games, where voters only have a few ways available to vote.

While some work has been devoted to the analysis of solution concepts such as *dominant strategies* and *strong equilibria*, this paper concentrates on Nash equilibria (NE). This most prominent solution concept has typically been overlooked, mainly because it appears to be too weak for this problem: there are typically many Nash equilibria in a voting game, but most of them are trivial. For example, if all voters vote for the same candidate, then this is clearly an equilibrium, since any single agent cannot change the result. This means that Plurality is *distorted*, i.e., there can be NE points in which the outcome is not truthful.

The lack of a single prominent solution for the game suggests that in order to fully understand the outcome of the voting procedure, it is not sufficient to consider voters' preferences. The strategies voters' choose to adopt, as well as the information available to them, are necessary for the analysis of possible outcomes. To play an equilibrium strategy for example, voters must know the preferences of others. Partial knowledge is also required in order to eliminate dominated strategies or to collude with other voters.

We consider the other extreme, assuming that voters have initially no knowledge regarding the preferences of the others, and cannot coordinate their actions. Such situations may arise, for example, when voters do not trust one another or have restricted communication abilities. Thus, even if two voters have exactly the same preferences, they may be reluctant or unable to share this information, and hence they will fail to coordinate their actions. Voters may still try to vote strategically, based on their current information, which may be partial or wrong. The analysis of such settings is of particular interest to AI as it tackles the fundamental problem of multi-agent decision making, where autonomous agents (that may be distant, self-interested and/or unknown to one another) have to choose a joint plan of action or allocate resources or goods. The central questions are (i) whether, (ii) how fast, and (iii) on what alternative the agents will agree.

In our (Plurality) voting model, voters start from some announcement (e.g., the truthful one), but can change their votes after observing the current announcement and outcome.[1] The game proceeds in turns, where a single voter changes his vote at each turn. We study different versions of this game, varying tie-breaking rules, weights and policies of voters, and the initial profile. Our main result shows that in order to guarantee convergence, it is necessary and sufficient that voters restrict their actions to natural best replies.

1.1 Related Work

There have been several studies applying game-theoretic solution concepts to voting games, and to Plurality in particular. [7] model a Plurality voting game where candidates and voters play strategically. They characterize all Nash equilibria in this game under the very restrictive assumption that the preference domain is *single peaked*. Another highly relevant work is that of [5], which concentrates on *dominant strategies* in Plurality voting. Their game formulation is identical to ours, and they prove a necessary and sufficient condition on the profile for the game to be dominance-solvable. Unfortunately, their analysis shows that this rarely occurs, making dominance perhaps a too-strong solution concept for actual situations. A weaker concept, though still stronger than NE, is *Strong Equilibrium*. In strong equilibrium no subset of agents can benefit by making a coordinated diversion. A variation of strong equilibrium was suggested by [10], which characterized its existence and uniqueness in Plurality games. Crucially, all aforementioned papers assume that voters have some prior knowledge regarding the preferences of others.

A more complicated model was suggested by [11], which assumes a non-atomic set of voters and some uncertainty regarding the preferences of other voters. Their main result is that every positional scoring rule (e.g., Veto, Borda, and Plurality) admits at least one voting equilibrium. In contrast, our model applies to a finite number of voters, that possess zero knowledge regarding the distribution of other voters' preferences.

Variations of Plurality and other voting rules have been proposed in order to increase resistance to strategic behavior (e.g., [4]). We focus on achieving a stable outcome *taking such behavior into account*.

Iterative voting procedures have also been investigated in the literature. [3] consider voters with different levels of information, where in the lowest level agents are myopic (as we assume as well). Others assume, in contrast, that voters have sufficient information to forecast the entire game, and show how to solve it with backward induction [6, 9]; most relevant to our work, [1] study conditions for convergence in such a model.

2 Preliminaries

2.1 The Game Form

There is a set C of m candidates, and a set V of n voters. A voting rule f allows each voter to submit his preferences over the candidates by selecting an action from a set A (in Plurality, $A = C$). Then, f chooses a non-empty set of winner candidates—i.e., it is a function $f : A^n \to 2^C \setminus \{\emptyset\}$.

[1]A real-world example of a voting interface that gives rise to a similar procedure is the recently introduced poll gadget for Google Wave. See http://sites.google.com/site/polloforwave.

v_1, v_2	a	b	c
a	$(14, 9, 3)$ $\{a\}$	$(10, 13, 3)$ $\{b\}$	$(10, 9, 7)$ $\{a\}$
b	$(11, 12, 3)$ $\{b\}$	$(7, 16, 3)$ $\{b\}$	$(7, 12, 7)$ $\{b\}$
c	$(11, 9, 6)$ $\{a\}$	$(7, 13, 6)$ $\{b\}$	$(7, 9, 10)$ $\{c\}$

Table 1: There is a set $C = \{a, b, c\}$ of candidates with initial scores $(7, 9, 3)$. Voter 1 has weight 3 and voter 2 has weight 4. Thus, $GF_T = \langle \{a, b, c\}, \{1, 2\}, (3, 2), (7, 9, 3)\rangle$. The table shows the outcome vector $\mathbf{s}(a_1, a_2)$ for every joint action of the two voters, as well as the set of winning candidates $GF_T(a_1, a_2)$. In this example there are no ties, and it thus fits both tie-breaking schemes.

Each such voting rule f induces a natural *game form*. In this game form, the strategies available to each voter are A, and the outcome of a joint action is $f(a_1, \ldots, a_n)$. Mixed strategies are not allowed. We extend this game form by including the possibility that only k out of the n voters may play strategically. We denote by $K \subseteq V$ the set of k strategic voters (agents) and by $B = V \setminus K$ the set of $n - k$ additional voters who have already cast their votes, and are not participating in the game. Thus, the outcome is $f(a_1, \ldots, a_k, b_{k+1}, \ldots, b_n)$, where $b_{k+1}, \ldots, b_n$ are fixed as part of the game form. This separation of the set of voters does not affect generality, but allows us to encompass situations where only some of the voters behave strategically.

From now on, we restrict our attention to the Plurality rule, unless explicitly stated otherwise. That is, the winner is the candidate (or a set of those) with the most votes; there is no requirement that the winner gain an absolute majority of votes. We assume each of the n voters has a fixed *weight* $w_i \in \mathbb{N}$. The *initial score* $\hat{s}(c)$ of a candidate c is defined as the total weight of the fixed voters who selected c—i.e., $\hat{s}(c) = \sum_{j \in B : b_j = c} w_j$. The *final score* of c for a given joint action $\mathbf{a} \in A^k$ is the total weight of voters that chose c (including the fixed set B): $s(c, \mathbf{a}) = \hat{s}(c) + \sum_{i \in K : a_i = c} w_i$. We sometimes write $s(c)$ if the joint action is clear from the context. We write $s(c) >_p s(c')$ if either $s(c) > s(c')$ or the score is equal and c has a higher priority (lower index). We denote by PL_R the Plurality rule with randomized tie breaking, and by PL_D the Plurality rule with deterministic tie breaking in favor of the candidate with the lower index. We have that $PL_R(\hat{\mathbf{s}}, \mathbf{w}, \mathbf{a}) = argmax_{c \in C} s(c, \mathbf{a})$, and $PL_D(\hat{\mathbf{s}}, \mathbf{w}, \mathbf{a}) = \{c \in C \text{ s.t. } \forall c' \neq c, s(c, \mathbf{a}) >_p s(c', \mathbf{a})\}$. Note that $PL_D(\hat{\mathbf{s}}, \mathbf{w}, \mathbf{a})$ is always a singleton.

For any joint action, its *outcome vector* $\mathbf{s}(\mathbf{a})$ contains the score of each candidate: $\mathbf{s}(\mathbf{a}) = (s(c_1, \mathbf{a}), \ldots, s(c_m, \mathbf{a}))$. For a tie-breaking scheme T ($T = D, R$) the *Game Form* $GF_T = \langle C, K, \mathbf{w}, \hat{\mathbf{s}}\rangle$ specifies the winner for any joint action of the agents—i.e., $GF_T(\mathbf{a}) = PL_T(\hat{\mathbf{s}}, \mathbf{w}, \mathbf{a})$. Table 1 demonstrates a game form with two weighted manipulators.

2.2 Incentives

We now complete the definition of our voting game, by adding incentives to the game form. Let $\mathcal{R}$ be the set of all strict orders over C. The order $\succ_i \in \mathcal{R}$ reflects the preferences of voter i over the candidates. The vector containing the preferences of all k agents is called a *profile*, and is denoted by $\mathbf{r} = (\succ_1, \ldots, \succ_k)$. The game form GF_T, coupled with a profile $\mathbf{r}$, define a normal form game $G_T = \langle GF_T, \mathbf{r}\rangle$ with k players. Player i prefers outcome $GF_T(\mathbf{a})$ over outcome $GF_T(\mathbf{a}')$ if $GF_T(\mathbf{a}) \succ_i GF_T(\mathbf{a}')$.

Note that for deterministic tie-breaking, every pair of outcomes can be compared. If ties are broken randomly, $\succ_i$ does *not* induce a complete order over outcomes, which are *sets* of candidates. A natural solution is to augment agents' preferences with cardinal utilities, where $u_i(c) \in \mathbb{R}$ is the utility of candidate c to agent i. This definition naturally extends to multiple winners by setting $u_i(W) = \frac{1}{|W|}\sum_{c \in W} u_i(c)$.[2] A utility function u is *consistent* with a preference relation $\succ_i$ if $u(c) > u(c') \Leftrightarrow c \succ_i c'$.

[2]This makes sense if we randomize the final winner from the set W. For a thorough discussion of cardinal and ordinal utilities in normal form games, see [2].

v_1, v_2	a	b	* c
* a	$\{\mathbf{a}\}\ \mathbf{3,2}$	$\{b\}\ 2,1$	* $\{\mathbf{a}\}\ \mathbf{3,2}$
b	$\{b\}\ 2,1$	$\{\mathbf{b}\}\ \mathbf{2,1}$	$\{b\}\ 2,1$
c	$\{a\}\ 3,2$	$\{b\}\ 2,1$	$\{c\}\ 1,3$

Table 2: A game $G_T = \langle GF_T, \mathbf{r}\rangle$, where GF_T is as in Table 1, and $\mathbf{r}$ is defined by $a \succ_1 b \succ_1 c$ and $c \succ_2 a \succ_2 b$. The table shows the ordinal utility of the outcome to each agent (the final score is not shown). **Bold** outcomes are the NE points. Here the truthful vote (marked with *) is also a NE.

Lemma 1. *For any utility function u which is consistent with preference order $\succ_i$, the following holds:*

1. $a \succ_i b \Rightarrow \forall W \subseteq C \setminus \{a,b\},\ u(\{a\} \cup W) > u(\{b\} \cup W)$;
2. $\forall b \in W, a \succ_i b \Rightarrow u(a) > u(\{a\} \cup W) > u(W)$.

The proof is trivial and is therefore omitted. Lemma 1 induces a partial preference order on the set of outcomes, but it is not yet complete if the cardinal utilities are not specified. For instance, the order $a \succ_i b \succ_i c$ does not determine if i will prefer $\{b\}$ over $\{a, c\}$. When utilities are given explicitly, every pair of outcomes can be compared, and we will slightly abuse the notation by using $GF_R(\mathbf{a}) \succ_i GF_R(\mathbf{a}')$ to note that i prefers the outcome of action $\mathbf{a}$ over that of $\mathbf{a}'$.

2.3 Manipulation and Stability

Having defined a normal form game, we can now apply standard solution concepts. Let $G_T = \langle GF_T, \mathbf{r}\rangle$ be a Plurality voting game, and let $\mathbf{a} = (a_{-i}, a_i)$ be a joint action in G_T. We say that $a_i \xrightarrow{i} a'_i$ is an *improvement step* of agent i if $GF_T(a_{-i}, a'_i) \succ_i GF_T(a_{-i}, a_i)$. A joint action $\mathbf{a}$ is a *Nash equilibrium* (NE), if no agent has an improvement step from $\mathbf{a}$ in G_T. That is, no agent can gain by changing his vote, provided that others keep their strategies unchanged. A priori, a game with pure strategies does not have to admit any NE. However, in our voting games there are typically (but not necessarily) many such points.

Now, observe that the preference profile $\mathbf{r}$ induces a special joint action $\mathbf{a}^*$, termed the *truthful vote*, such that $\mathbf{a}^*(\mathbf{r}) = (a^*_1, \ldots, a^*_k)$, where $a^*_i \succ_i c$ for all $c \neq a^*_i$. We also call $\mathbf{a}^*(\mathbf{r})$ the *truthful state* of G_T, and refer to $GF_T(\mathbf{a}^*(\mathbf{r}))$ as the *truthful outcome* of the game. If i has an improvement step in the truthful state, then this is a *manipulation*.[3] Thus, $\mathbf{r}$ cannot be manipulated if and only if $\mathbf{a}^*(\mathbf{r})$ is a Nash equilibrium of $G_T = \langle GF_T, \mathbf{r}\rangle$. However, the truthful vote may or may not be included in the NE points of the game, as can be seen from Table 2.

2.4 Game Dynamics

We finally consider natural *dynamics* in Plurality voting games. Assume that players start by announcing some initial vote, and then proceed and change their votes until no one has objections to the current outcome. It is not, however, clear how rational players would act to achieve a stable decision, especially when there are multiple equilibrium points. However, one can make some plausible assumptions about their behavior. First, the agents are likely to only make improvement steps, and to keep their current strategy if such a step is not available. Thus, the game will end when it first reaches a NE. Second, it is often the case that the initial state is truthful, as agents know that they can reconsider and vote differently, if they are not happy with the current outcome.

We start with a simple observation that if the agents may change their votes simultaneously, then convergence is not guaranteed, even if the agents start with the truthful vote and use best replies—that is, vote for their most preferred candidate out of potential winners in the current round.

[3]This definition of manipulation coincides with the standard definition from social choice theory.

Proposition 2. *If agents are allowed to re-vote simultaneously, the improvement process may never converge.*

Example. The counterexample is the game with 3 candidates $\{a, b, c\}$ with initial scores given by $(0, 0, 2)$. There are 2 voters $\{1, 2\}$ with weights $w_1 = w_2 = 1$ and the following preferences: $a \succ_1 b \succ_1 c$, and $b \succ_2 a \succ_2 c$. The two agents will repeatedly swap their strategies, switching endlessly between the states $\mathbf{a}(\mathbf{r}) = (a, b)$ and (b, a). Note that this example works for both tie-breaking schemes. $\diamondsuit$

We therefore restrict our attention to dynamics where simultaneous improvements are not available. That is, given the initial vote $\mathbf{a}_0$, the game proceeds in steps, where at each step t, a single player may change his vote, resulting in a new state (joint action) $\mathbf{a}_t$. The process ends when no agent has objections, and the outcome is set by the last state. Such a restriction makes sense in many computerized environments, where voters can log-in and change their vote at any time.

In the remaining sections, we study the conditions under which such iterative games reach an equilibrium point from either an arbitrary or a truthful initial state. We consider variants of the game that differ in tie-breaking schemes or assumptions about the agents' weights or behavior. In cases where convergence is guaranteed, we are also interested in knowing how fast it will occur, and whether we can say anything about the identity of the winner. For example, in Table 2, the game will converge to a NE from any state in at most two steps, and the outcome will be a (which happens to be the truthful outcome), unless the players initially choose the alternative equilibrium (b, b) with outcome b.

3 Results

Let us first provide some useful notation. We denote the outcome at time t by $o_t = PL(\mathbf{a}_t) \subseteq C$, and its score by $s(o_t)$. Suppose that agent i has an improvement step at time t, and as a result the winner switched from o_{t-1} to o_t. The possible steps of i are given by one of the following types (an example of such a step appears in parentheses):

type 1 from $a_{i,t-1} \notin o_{t-1}$ to $a_{i,t} \in o_t$; (step 1 in Ex.4a.)

type 2 from $a_{i,t-1} \in o_{t-1}$ to $a_{i,t} \notin o_t$; (step 2 in Ex.4a.)

type 3 from $a_{i,t-1} \in o_{t-1}$ to $a_{i,t} \in o_t$; (step 1 in Ex.4b.),

where inclusion is replaced with equality for deterministic tie-breaking. We refer to each of these steps as a *better reply* of agent i. If $a_{i,t}$ is i's most preferred candidate capable of winning, then this is a *best reply*.[4] Note that there are no best replies of type 2. Finally, we denote by $s_t(c)$ the score of a candidate c *without the vote of the currently playing agent*; thus, it always holds that $s_{t-1}(c) = s_t(c)$.

3.1 Deterministic Tie-Breaking

Our first result shows that under the most simple conditions, the game must converge.

Theorem 3. *Let G_D be a Plurality game with deterministic tie-breaking. If all agents have weight 1 and use best replies, then the game will converge to a NE from any state.*

[4]Any rational move of a myopic agent in the normal form game corresponds to exactly one of the three types of better-reply. In contrast, the definition of best-reply is somewhat different from the traditional one, which allows the agent to choose any strategy that guarantees him a best possible outcome. Here, we assume the improver makes the more natural response by actually voting for o_t. Thus, under our definition, the best reply is always unique.

Proof. We first show that there can be at most $(m-1) \cdot k$ sequential steps of type 3. Note that at every such step $a \xrightarrow{i} b$ it must hold that $b \succ_i a$. Thus, each voter can only make $m-1$ such subsequent steps.

Now suppose that a step $a \xrightarrow{i} b$ of type 1 occurs at time t. We claim that at any later time $t' \geq t$: (I) there are at least two candidates whose score is *at least* $s(o_{t-1})$; (II) the score of a will not increase at t'. We use induction on t' to prove both invariants. Right after step t we have that

$$s_t(b) + 1 = s(o_t) >_p s(o_{t-1}) >_p s_t(a) + 1 \ . \tag{1}$$

Thus, after step t we have at least two candidates with scores of at least $s(o_{t-1})$: $o_t = b$ and $o_{t-1} \neq b$. Also, at step t the score of a has decreased. This proves the base case, $t' = t$.

Assume by induction that both invariants hold until time $t' - 1$, and consider step t' by voter j. Due to (I), we have at least two candidates whose score is at least $s(o_{t-1})$. Due to (II) and Equation (1) we have that $s_{t'}(a) \leq_p s_t(a) <_p s(o_{t-1}) - 1$. Therefore, no single voter can make a a winner and thus a cannot be the best reply for j. This means that (II) still holds after step t'. Also, j has to vote for a candidate c that can beat $o_{t'}$—i.e., $s_{t'}(c) + 1 >_p s(o_{t'}) >_p s(o_{t-1})$. Therefore, after step t' both c and $o_{t'} \neq c$ will have a score of at least $s(o_{t-1})$—that is, (I) also holds. □

The proof also supplies us with a polynomial bound on the rate of convergence. At every step of type 1, at least one candidate is ruled out permanently, and there at most k times a vote can be withdrawn from a candidate. Also, there can be at most mk steps of type 3 between such occurrences. Hence, there are in total at most m^2k^2 steps until convergence. It can be further shown that if all voters start from the truthful state then there are no type 3 steps at all. Thus, the score of the winner never decreases, and convergence occurs in at most mk steps. The proof idea is similar to that of the corresponding randomized case in Theorem 8.

We now show that the restriction to best replies is necessary to guarantee convergence.

Proposition 4. *If agents are not limited to best replies, then: (a) there is a counterexample with two agents; (b) there is a counterexample with an initial truthful vote.*

Example 4a. $C = \{a, b, c\}$. We have a single fixed voter voting for a, thus $\hat{\mathbf{s}} = (1, 0, 0)$. The preference profile is defined as $a \succ_1 b \succ_1 c,\ c \succ_2 b \succ_2 a$. The following cycle consists of better replies (the vector denotes the votes (a_1, a_2) at time t, the winner appears in curly brackets):

$$(b, c)\{a\} \xrightarrow{2} (b, b)\{b\} \xrightarrow{1} (c, b)\{a\} \xrightarrow{2} (c, c)\{c\} \xrightarrow{1} (b, c) \quad \diamondsuit$$

Example 4b. $C = \{a, b, c, d\}$. Candidates a, b, and c have 2 fixed voters each, thus $\hat{\mathbf{s}} = (2, 2, 2, 0)$. We use 3 agents with the following preferences: $d \succ_1 a \succ_1 b \succ_1 c,\ c \succ_2 b \succ_2 a \succ_2 d$ and $d \succ_3 a \succ_3 b \succ_3 c$. Starting from the truthful state (d, c, d) the agents can make the following two improvement steps (showing only the outcome):

$$(2, 2, 3, 2)\{c\} \xrightarrow{1} (2, 3, 3, 1)\{b\} \xrightarrow{3} (3, 3, 3, 0)\{a\} \ ,$$

after which agents 1 and 2 repeat the cycle shown in (4a). $\diamondsuit$

Weighted voters While using the best reply strategies guaranteed convergence for equally weighted agents, this is no longer true for non-identical weights:

Proposition 5. *There is a counterexample with 3 weighted agents that start from the truthful state and use best replies.*

The proof is omitted for the sake of brevity.

However, if there are *only two* weighted voters, either restriction is sufficient:

Theorem 6. *Let G_D be a Plurality game with deterministic tie-breaking. If $k = 2$ and both agents (a) use best replies* **or** *(b) start from the truthful state, a NE will be reached.*

Proof of (6a). Assume there is a cycle, and consider the winners in the first steps: $\{x\} \xrightarrow{1} \{y\} \xrightarrow{2} \{z\}$. Suppose that after step 1 both agents vote for different candidates ($a_{1,2} \neq a_{1,1} = y$). This holds for any later step, as an agent has no reason to vote for the current winner. An agent can never make a step of type 3 (after the first step), since at every step the winner is the candidate that the other agent is voting for. If the first step brings the agents to the same candidate, then in the second step they split again ($a_{2,1} \neq a_{2,2} = z$), and we are back in the previous case. $\square$

Proof of (6b). We show that the score of the winner can only increase. This clearly holds in the first step, which must be of type 1. Once again, we have that the two agents always vote for different candidates, and thus only steps that increase the score can change the identity of the winner. $\square$

3.2 Randomized Tie-Breaking

The choice of tie-breaking scheme has a significant impact on the outcome, especially when there are few voters. A randomized tie-breaking rule has the advantage of being neutral —no specific candidate or voter is preferred over another.

In order to prove convergence under randomized tie-breaking, we must show that convergence is guaranteed for *any* utility function which is consistent with the given preference order. That is, we may only use the relations over outcomes that follow directly from Lemma 1. To disprove, it is sufficient to show that for a specific assignment of utilities, the game forms a cycle. In this case, we say that there is a *weak counterexample*. When the existence of a cycle will follow only from the relations induced by Lemma 1, we will say that there is a *strong counterexample*, since it holds for any profile of utility scales that fits the preferences.

In contrast to the deterministic case, the weighted randomized case does not always converge to a Nash equilibrium or possess one at all, even with (only) two agents.

Proposition 7. *There is a strong counterexample G_R for two weighted agents with randomized tie-breaking, even if both agents start from the truthful state and use best replies.*

Example. $C = \{a, b, c\}$, $\hat{\mathbf{s}} = (0, 1, 3)$. There are 2 agents with weights $w_1 = 5$, $w_2 = 3$ and preferences $a \succ_1 b \succ_1 c$, $b \succ_2 c \succ_2 a$ (in particular, $b \succ_2 \{b, c\} \succ_2 c$). The resulting 3×3 normal form game contains no NE states. $\diamondsuit$

Nevertheless, the conditions mentioned are sufficient for convergence if all agents have the same weight.

Theorem 8. *Let G_R be a Plurality game with randomized tie-breaking. If all agents have weight 1 and use best replies, then the game will converge to a NE from the truthful state.*

Proof. Our proof shows that in each step, the current agent votes for a *less* preferred candidate. Clearly, the first improvement step of every agent must hold this invariant.

Assume, toward deriving a contradiction, that $b \xrightarrow{i} c$ at time t_2 is the first step s.t. $c \succ_i b$. Let $a \xrightarrow{i} b$ at time $t_1 < t_2$ be the previous step of the same agent i.

We denote by $M_t = o_t$ the set of all winners at time t. Similarly, L_t denotes all candidates whose score is $s(o_t) - 1$.

We claim that for all $t < t_2$, $M_t \cup L_t \subseteq M_{t-1} \cup L_{t-1}$, i.e., the set of "almost winners" can only shrink. Also, the score of the winner cannot decrease. Observe that in order to contradict any of these assertions, there must be a step $x \xrightarrow{j} y$ at time t, where $\{x\} = M_{t-1}$ and $y \notin M_{t-1} \cup L_{t-1}$. In that case, $M_t = L_{t-1} \cup \{x, y\} \succ_j \{x\}$, which means either that $y \succ_j x$ (in contradiction to the minimality of t_2) or that y is not a best reply.

From our last claim we have that $s(o_{t_1-1}) \le s(o_{t'})$ for any $t_1 \le t' < t_2$. Now consider the step t_1. Clearly $b \in M_{t_1-1} \cup L_{t_1-1}$ since otherwise voting for b would not make it a winner. We consider the cases for c separately:

Case 1: $c \notin M_{t_1-1} \cup L_{t_1-1}$. We have that $s_{t_1}(c) \le s(o_{t_1-1}) - 2$. Let t' be any time s.t. $t_1 \le t' < t_2$, then $c \notin M_{t'} \cup L_{t'}$. By induction on t', $s_{t'}(c) \le s_{t_1}(c) \le s(o_{t_1-1}) - 2 \le s(o_{t'}) - 2$, and therefore c cannot become a winner at time $t'+1$, and the improver at time $t'+1$ has no incentive to vote for c. In particular, this holds for $t'+1 = t_2$; hence, agent i will not vote for c.

Case 2: $c \in M_{t_1-1} \cup L_{t_1-1}$. It is not possible that $b \in L_{t_1-1}$ or that $c \in M_{t_1-1}$: since $c \succ_i b$ and i plays best reply, i would have voted for c at step t_1. Therefore, $b \in M_{t_1-1}$ and $c \in L_{t_1-1}$. After step t_1, the score of b equals the score of c plus 2; hence, we have that $M_{t_1} = \{b\}$ and $c \notin M_{t_1} \cup L_{t_1}$, and we are back in case 1.

In either case, voting for c at step t_2 leads to a contradiction. Moreover, as agents only vote for a less-preferred candidate, each agent can make at most $m-1$ steps, hence, at most $(m-1) \cdot k$ steps in total. $\square$

However, in contrast to the deterministic case, convergence is no longer guaranteed, if players start from an arbitrary profile of votes. The following example shows that in the randomized tie-breaking setting even best reply dynamics may have cycles, albeit for specific utility scales.

Proposition 9. *If agents start from an arbitrary profile, there is a weak counterexample with 3 agents of weight 1, even if they use best replies.*

Example. There are 4 candidates $\{a,b,c,x\}$ and 3 agents with utilities $u_1 = (5,4,0,3)$, $u_2 = (0,5,4,3)$ and $u_3 = (4,0,5,3)$. In particular, $a \succ_1 \{a,b\} \succ_1 x \succ_1 \{a,c\}$; $b \succ_2 \{b,c\} \succ_2 x \succ_2 \{a,b\}$; and $c \succ_3 \{a,c\} \succ_3 x \succ_3 \{b,c\}$. From the state $\mathbf{a}_0 = (a,b,x)$ with $\mathbf{s}(\mathbf{a}_0) = (1,1,0,1)$ and the outcome $\{a,b,x\}$, the following cycle occurs: $(1,1,0,1)\{a,b,x\} \xrightarrow{2} (1,0,0,2)\{x\} \xrightarrow{3} (1,0,1,1)\{a,x,c\} \xrightarrow{1} (0,0,1,2)\{x\} \xrightarrow{2} (0,1,1,1)\{x,b,c\} \xrightarrow{3} (0,1,0,2)\{x\} \xrightarrow{1} (1,1,0,1)\{a,b,x\}$. $\diamondsuit$

As in the previous section, if we relax the requirement for best replies, there may be cycles even from the truthful state.

Proposition 10. *(a) If agents use arbitrary better replies, then there is a strong counterexample with 3 agents of weight 1. Moreover, (b) there is a weak counterexample with 2 agents of weight 1, even if they start from the truthful state.*

Example 10a. $C = \{a,b,c\}$ with initial score $\hat{\mathbf{s}} = (0,1,0)$. The initial state is $\mathbf{a}_0 = (a,a,b)$—that is, $\mathbf{s}(\mathbf{a}_0) = (2,2,0)$ and the outcome is the winner set $\{a,b\}$. Consider the following cyclic sequence (we write the score vector and the outcome in each step): $(2,2,0)\{a,b\} \xrightarrow{2} (1,2,1)\{b\} \xrightarrow{1} (0,2,2)\{b,c\} \xrightarrow{3} (1,1,2)\{c\} \xrightarrow{2} (2,1,1)\{a\} \xrightarrow{3} (1,2,1)\{b\} \xrightarrow{1} (2,2,0)\{a,b\}$. If the preferences are $a \succ_1 c \succ_1 b$, $b \succ_2 a \succ_2 c$ and $c \succ_3 b \succ_3 a$, then each step is indeed an improvement step for the agent whose index is on top of the arrow. $\diamondsuit$

Example 10b. We use 5 candidates with initial score $(1,1,2,0,0)$, and 2 agents with utilities $u_1 = (5,3,2,8,0)$ and $u_2 = (4,2,5,0,8)$. In particular, $\{b,c\} \succ_1 c$, $\{a,c\} \succ_1 \{a,b,c\}$, and $\{a,b,c\} \succ_2 \{b,c\}$, $c \succ_2 \{a,c\}$, and the following cycle occurs: $(1,1,2,1,1)\{c\} \xrightarrow{1} (1,2,2,0,1)\{b,c\} \xrightarrow{2} (2,2,2,0,0)\{a,b,c\} \xrightarrow{1} (2,1,2,1,0)\{a,c\} \xrightarrow{2} (1,1,2,1,1)\{c\}$. $\diamondsuit$

3.3 Truth-Biased Agents

So far we assumed purely rational behavior on the part of the agents, in the sense that they were indifferent regarding their chosen action (vote), and only cared about the outcome. Thus, for example, if an agent cannot affect the outcome at some round, he simply keeps his *current vote*. This assumption is indeed common when dealing with normal form games, as there is no reason to prefer

Tie breaking	Dynamics / Initial state	Best reply from: Truth	Best reply from: Anywhere	Any better reply from: Truth	Any better reply from: Anywhere	Truth biased
Deterministic	Weighted $(k > 2)$	X (5)	X	X	X	X
	Weighted $(k = 2)$	**V**	**V** (6a)	V (6b)	X (4a)	X
	Non-weighted	**V**	**V** (3)	X (4b)	X	X (11a)
Randomized	Weighted	X (7)	X	X	X	X
	Non-weighted	**V** (8)	X (9)	X (10)	X (10)	X (11b)

Table 3: We highlight cases where convergence is guaranteed. The number in brackets refers to the index of the corresponding theorem (marked with **V**) or counterexample (X). Entries with no index follow from other entries in the table.

one strategy over another if outcomes are the same. However, in voting problems it is typically assumed that voters will vote *truthfully* unless they have an incentive to do otherwise. As our model incorporates both settings, it is important to clarify the exact assumptions that are necessary for convergence.

In this section, we consider a variation of our model where agents always prefer their higher-ranked outcomes, but will vote honestly if the outcome remains the same—i.e., the agents are *truth-biased*. Formally, let $W = PL_T(\hat{\mathbf{s}}, \mathbf{w}, a_i, \mathbf{a}_{-i})$ and $Z = PL_T(\hat{\mathbf{s}}, \mathbf{w}, a'_i, \mathbf{a}_{-i})$ be two possible outcomes of i's voting. Then, the action a'_i is *better than* a_i if either $Z \succ_i W$, or $Z = W$ and $a'_i \succ_i a_i$. Note that with this definition there is a strict preference order over all possible actions of i at every step. Unfortunately, truth-biased agents may not converge even in the simplest settings:

Proposition 11. *There are strong counterexamples for (a) deterministic tie-breaking, and (b) randomized tie-breaking. This holds even with two non-weighted truth-biased agents that use best reply dynamics and start from the truthful state.*

Example 11a. We use 4 candidates with no initial score. The preferences are defined as $c \succ_1 a \succ_1 b \succ_1 d$ and $d \succ_2 b \succ_2 a \succ_2 c$. The reader can easily verify that in the resulting 4×4 game there are no NE states. $\diamondsuit$

Example 11b. There are 4 candidates with initial scores $(0, 0, 1, 2)$. The preference profile is given by $a \succ_1 c \succ_1 d \succ_1 b$, $b \succ_2 d \succ_2 c \succ_2 a$. Consider the following cycle, beginning with the truthful state: $(1, 1, 1, 2) \xrightarrow{1} (0, 1, 2, 2) \xrightarrow{2} (0, 0, 2, 3) \xrightarrow{1} (1, 0, 1, 3) \xrightarrow{2} (1, 1, 1, 2)$. $\diamondsuit$

4 Discussion

We summarize the results in Table 3. We can see that in most cases convergence is not guaranteed unless the agents restrict their strategies to "best replies"—i.e., always select their most-preferred candidate that can win. Also, deterministic tie-breaking seems to encourage convergence more often. This makes sense, as the randomized scheme allows for a richer set of outcomes, and thus agents have more options to "escape" from the current state. Neutrality can be maintained by randomizing a tie-breaking order and publicly announcing it *before* the voters cast their votes.

We saw that if voters are non-weighted, begin from the truthful announcement and use best reply, then they always converge within a polynomial number of steps (in both schemes), but to what outcome? The proofs show that the score of the winner can only increase, and by at most 1 in each iteration. Thus possible winners are only candidates that are either tied with the (truthful) Plurality winner, or fall short by one vote. This means that it is not possible for arbitrarily "bad" candidates to be elected in this process, but does not preclude a competition of more than two candidates. This result suggests that widely observed phenomena such as Duverger's law only apply in situations where voters have a larger amount of information regarding one another's preferences, e.g., via public polls.

Our analysis is particularly suitable when the number of voters is small, for two main reasons. First, it is technically easier to perform an iterative voting procedure with few participants. Second, the question of convergence is only relevant when cases of tie or near-tie are common. An analysis in the spirit of [11] would be more suitable when the number of voters increases, as it rarely happens that a single voter would be able to influence the outcome, and almost any outcome is a Nash equilibrium. This limitation of our formulation is due to the fact that the behaviors of voters encompass only myopic improvements. However, it sometimes makes sense for a voter to vote for some candidate, even if this will not immediately change the outcome (but may contribute to such a change if other voters will do the same).

A new voting rule We observe that the improvement steps induced by the best reply policy are unique. If, in addition, the order in which agents play is fixed, we get a *new voting rule*—Iterative Plurality. In this rule, agents submit their full preference profiles, and the center simulates an iterative Plurality game, applying the best replies of the agents according to the predetermined order. It may seem at first glance that Iterative Plurality is somehow resistant to manipulations, as the outcome was shown to be an equilibrium. This is not possible of course, and indeed agents can still manipulate the new rule by submitting false preferences. Such an action can cause the game to converge to a different equilibrium (of the Plurality game), which is better for the manipulator.

Future work It would be interesting to investigate computational and game-theoretic properties of the new, iterative, voting rule. For example, perhaps strategic behavior is scarcer, or computationally harder. Another interesting question arises regarding possible strategic behavior of the election chairperson: can voters be ordered so as to arrange the election of a particular candidate? This is somewhat similar to the idea of manipulating the agenda. Of course, a similar analysis can be carried out on voting rules other than Plurality, or with variations such as voters that join gradually. Such analyses might be restricted to best reply dynamics, as in most interesting rules the voter strategy space is very large. Another key challenge is to modify our best-reply assumption to reflect non-myopic behavior. Finally, even in cases where convergence is not guaranteed, it is worth studying the *proportion* of profiles that contain cycles.

5 Acknowledgments

The authors thank Michael Zuckerman and Ilan Nehama for helpful discussions. R. Meir and J. S. Rosenschein are partially supported by Israel Science Foundation Grant #898/05. M. Polukarov and N. R. Jennings are supported by the ALADDIN project, jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C548051/1).

References

[1] Stèphane Airiau and Ulle Endriss. Iterated majority voting. In *Proccedings of ADT-09*, pages 38–49. Springer Verlag, 2009.

[2] Tilman Borgers. Pure strategy dominance. *Econometrica*, 61(2):423–430, 1993.

[3] Samir Chopra, Eric Pacuit, and Rohit Parikh. Knowledge-theoretic properties of strategic voting. Presented in JELIA-04, Lisbon, Portugal, 2004.

[4] Vincent Conitzer and Tuomas Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of IJCAI-03*, pages 781–788. Morgan Kaufmann, 2003.

[5] Amrita Dhillon and Ben Lockwood. When are plurality rule voting games dominance-solvable? *Games and Economic Behavior*, 46:55–75, 2004.

[6] Robin Farquharson. *Theory of Voting*. Yale Uni. Press, 1969.

[7] Timothy J. Feddersen, Itai Sened, and Stephen G. Wright. Rational voting and candidate entry under plurality rule. *American Journal of Political Science*, 34(4):1005–1016, 1990.

[8] Robert Forsythe, Thomas Rietz, Roger Myerson, and Robert Weber. An experimental study of voting rules and polls in three-candidate elections. *Int. J. of Game Theory*, 25(3):355–83, 1996.

[9] Richard D. McKelvey and Richard Niemi. A multistage representation of sophisticated voting for binary procedures. *Journal of Economic Theory*, 18:1–22, 1978.

[10] Matthias Messner and Mattias K. Polborn. Robust political equilibria under plurality and runoff rule, 2002. Mimeo, Bocconi University.

[11] Roger B. Myerson and Robert J. Weber. A theory of voting equilibria. *The American Political Science Review*, 87(1):102–114, 1993.

[12] Donald G. Saari. Susceptibility to manipulation. *Public Choice*, 64:21–41, 1990.

Reshef Meir
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Givat Ram, Jerusalem 91904, Israel
Email: `reshef24@cs.huji.ac.il`

Maria Polukarov
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO17 1BJ, United Kingdom
Email: `mp3@ecs.soton.ac.uk`

Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Givat Ram, Jerusalem 91904, Israel
Email: `jeff@cs.huji.ac.il`

Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO17 1BJ, United Kingdom
Email: `nrj@ecs.soton.ac.uk`

The Probability of Safe Manipulation

Mark C. Wilson and Reyhaneh Reyhani

Abstract

The concept of safe manipulation has recently been introduced by Slinko and White. We show how to compute the asymptotic probability that a safe manipulation exists for a given scoring rule under the uniform distribution on voting situations (the so-called Impartial Anonymous Culture). The technique used is computation of volumes of convex polytopes. We present explicit numerical results in the 3-candidate case.

1 Introduction

The Gibbard-Satterthwaite theorem [2, 5] shows that for each nondictatorial social choice function allowing unrestricted preferences of voters over alternatives and such that the $m \geq 3$ alternatives can each win in some profile, there always exists a profile which is unstable. In other words, in the voting game with ordinal utilities given by the voter preferences of that profile, the strategy where all voters express their sincere preference is not a Nash equilibrium, so that at least one voter has incentive to deviate unilaterally by expressing an insincere preference. For common social choice functions, the probability that a single individual can succeed in changing the election result converges to zero as n, the number of voters, tends to ∞. Thus the question of coalitional manipulation is more interesting.

Coalitions must be of fairly large size in order to manipulate effectively. For example, under the IC hypothesis (uniform distribution on profiles) the manipulating coalitions are typically of order $\sqrt{n}$, while they can be considerably larger under other preference distributions [7, 6]. Thus the question of coalition formation becomes important, because there are substantial coordination difficulties to be overcome in order to manipulate successfully.

Slinko and White [8] proposed a simple model for coalition formation, whereby a "leader" publicizes a strategic vote and voters sharing the leader's preference order decide whether to follow this strategy or vote sincerely. As a topic for further research, [8] lists the study of the probability that such an attempt succeeds sometimes and the coalition members never fare worse than with the sincere outcome. The present paper studies this topic for a well-known preference distribution, namely the Impartial Anonymous Culture.

2 Definitions and basic properties

Let $m \geq 1$ be an integer and let $\mathcal{C}$ be a set of size m, the set of **alternatives** (or **candidates**). Let $n \geq 1$ be an integer and let $\mathcal{V}$ be a set of size n, the set of **agents** (or **voters**). Each agent is assumed to have a total order of the alternatives, the agent's **preference order**. An agent a **strongly prefers** alternative i to alternative j if and only if i is strictly above j in a's preference order; if we also allow the possibility $i = j$ then we just use the term **prefers**. There are $M := m!$ possible such preference orders, which we call **types**. We denote the set of all types by $\mathcal{T}$ and the set of all agents of type t by $\mathcal{V}_t$. A multiset from $\mathcal{T}$ with total weight n is a **voting situation**, whereas a function taking $\mathcal{V}$ to $\mathcal{T}$ is a **profile**. Each voting situation corresponds naturally to several profiles, corresponding to the different permutations of the multiset.

Let F be a social choice function, a map that associates an element of $\mathcal{C}$ to each profile. If this map depends only on the voting situation, then the rule is called **anonymous**.

The **Impartial Anonymous Culture** (IAC) is the uniform probability distribution on the set of voting situations. If F is anonymous, then we can compute the probability of an event under IAC simply by counting voting situations. Since voting situations can be encoded by tuples of natural numbers $(n_1, \ldots, n_M)$ with $\sum_i n_i = n$, this amounts to counting lattice points in a subset of a dilated standard simplex.

In the following definitions it is assumed that agents not mentioned continue to vote sincerely.

Definition 1. *A voting situation is* ***manipulable*** *if there is some subset X of voters such that, if all members of X vote insincerely, the result is strongly preferred by all members of X to the sincere outcome. Such a set X is called a* ***manipulating coalition****.*

A voting situation is ***safe*** *for voters of type t if there is some type t' such that for all x with $0 \leq x \leq n_t$, whenever x agents of type t change their vote to t', these agents weakly prefer the resulting outcome to the sincere outcome.*

A voting situation is ***safely manipulable*** *by voters of type t if it is safe for them, and there is some value of x for which the agents concerned strongly prefer the resulting outcome to the sincere outcome.*

There are three main points in the definition of safe manipulation:

- the manipulating coalition consists only of voters of a single type;
- the manipulating strategy is the same for all coalition members;
- the size of this coalition is unknown and there must be no risk of obtaining a worse outcome than the sincere one (through "undershooting" or "overshooting").

Overshooting occurs when the following situation holds. If some number x change from t to t', the result is strongly preferred to the sincere one, but if some number $y > x$ change, the sincere result is strongly preferred to the latter outcome. **Undershooting** is the same, but with $y > x$ replaced by $y < x$. Examples in [8] show that both phenomena can occur. In fact they can both occur in the same voting situation as shown by the following example.

Example 1. *Let $m = 5$ and consider the voting situation with 3 voters having each of the possible preference orders, except the order 12345 which has 4 voters. The scoring rule (see Section 3 for definitions if necessary) with weights $(55, 39, 33, 21, 0)$ yields scores that induce an overall ordering 12345 (meaning candidate 1 wins, candidate 2 is second, etc). Consider voters of type 53124 and the strategy of voting 35241. If 1 voter switches to this strategy, the new winner is candidate 2; if 2 voters switch, then the new winner is candidiate 3; if 3 voters switch, the new winner is candidate 4. This shows that undershooting and overshooting can be possible for the same type and choice of insincere strategy in the same voting situation.*

Remark 1. *We can consider a game in which the set T of types of voters is partitioned into two subsets, T', T''. The set T'' consists of all types of voters whose only action is to vote sincerely, while voters corresponding to types in T' have all possible votes open to them (we do not allow abstention). In the case where $T'' = \emptyset$ and this is common knowledge, we have a fully strategic game. A situation is manipulable if and only if it is not a strong Nash equilibrium of this game.*

When $T' = T_i$ for some fixed type T_i, there is a different game that is easier to analyse. A situation is safe for members of T' if and only if there exists a pure strategy that weakly dominates the sincere strategy, and safely manipulable if and only if there exists a pure strategy that dominates the sincere strategy.

Remark 2. *Note that for each type of voter that ranks the sincere winner lowest, every situation is safe (in fact a stronger statement is true: such voters have nothing to lose by strategic voting, no matter what T' and T'' are). On the other hand, types that rank the sincere winner highest can never manipulate.*

3 Algorithms and polytopes

We restrict to scoring rules. However the method described works more generally (for some rules, much more care may be needed when considering ties).

Scoring rules

Definition 2. *Let $w = (w_1, \ldots, w_m)$ be such that all $w_i \geq 0$, $w_1 \geq w_2 \ldots w_m$ and $w_1 > w_m$. The **scoring rule defined by** w gives the following score to each candidate c:*

$$|c| = \sum_{t \in \mathcal{T}} n_t w_{r(c,t)}$$

*where $r(c, t)$ denotes the rank of c according to type t. The candidates with largest score are the winners. The scores give a **social ordering** of candidates (the value of the associated social welfare function).*

Remark 3. *If a tie occurs for largest score, then a separate tiebreaking procedure is needed in order to obtain a social choice function. This can be a difficult issue, but fortunately when considering asymptotic results under IAC as in this paper, we do not need to consider it further. This is because the set of tied situations has measure zero in the limit as $n \to \infty$.*

We now impose an order on the candidates, and write $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$. The types are then identified with permutations of $\{1, \ldots, m\}$ and can be written in the usual way. We describe the scores by the **scoreboard**, the tuple $s = (|c_1|, \ldots, |c_m|)$ of scores. The group of types acts on the scoreboard w via permuting candidates and we denote the action of t on w by w^t. In terms of our current notation, we have

$$s = \sum_{t \in \mathcal{T}} n_t w^{t^{-1}}.$$

Example 2. *Let $m = 3$ and consider the voting situation in which 6 agents have preference order 312 and 2 agents have order 213. Under the plurality rule given by $w = (1, 0, 0)$, the scoreboard is $(0, 2, 6)$ and c_3 wins, the social ordering being 321. Under the Borda rule given by $(2, 1, 0)$, the scoreboard is $(8, 4, 12)$ and the order of second and third place is reversed, the social ordering being 312. Under the antiplurality rule given by $w = (1, 1, 0)$, the scoreboard is $(8, 2, 6)$ and social ordering is 132. There is no weight vector for which c_2 can win, as c_3 always has a higher score.*

Without loss of generality we assume from now on that the sincere social ordering is $123 \ldots m$.

3.1 When t and t' are specified

Fix types t and t' until further notice. We now describe the set S of safely manipulable voting situations. S is the union $\bigcup_{t \in T} S_t$, where S_t is the set of situations that are safely manipulable by voters of type t. This can be further refined to $S = \bigcup_{t \neq t'} S_{t,t'}$ where $S_{t,t'}$ is the set of situations that are safely manipulable by voters of type t using strategy t'.

To describe $S_{t,t'}$, we use the following basic observations.

Let x denote the number of members in a coalition of type t who vote insincerely and suppose they vote t'. Then the new and old scoreboards are related by

$$s' - s = x\left(w^{(t')^{-1}} - w^{t^{-1}}\right).$$

For brevity we refer to those candidates ranked above candidate 1 by agents of type t as **good**, and those ranked below 1 as **bad**. For example, when $m = 3$ and the social ordering is 123, then according to an agent of type 213, c_2 is good and c_3 is bad. The new outcome is preferred by type t agents if and only if no bad candidate is the new winner. It is strongly preferred if and only if some good candidate is the new winner.

Proposition 1. *When $m = 3$, undershooting can never occur, and overshooting occurs if and only if some bad candidate wins when $x = n_t$.*

Proof. First note that as a function of x, the differences in scores of each alternative between the sincere and strategic voting situation are (linearly) either increasing or decreasing. Thus if candidate i is above candidate j for some x but below for some larger value of x, it will remain below for all even larger values of x. For types 123 and 132, no better result can be achieved by strategic voting; for types 231 and 321, no worse result. The only other cases are types 213 and 312. In each case there is only one good and one bad candidate: once one overtakes the other and the sincere winner, it stays ahead and cannot be subsequently beaten by another candidate of the opposite type. □

Proposition 2. *The following algorithm solves the decision problem for safe manipulation for scoring rules, and runs in polynomial time provided the tiebreaking procedure does.*

Let $|c|_x$ denote the score of candidate c when x agents have switched from t to t', and let L be the set of points of intersection of the graphs of the functions $x \mapsto |c|_x$ for $0 \leq x \leq n_t$. Sort the elements of L. For each interval formed by successive elements, compute the maximum score B of all bad candidates, and the maximum score G of all good candidates. If $B > G$ for any interval (or $B = G$ and the tiebreaking procedure says that a valid manipulation in favour of a bad candidate has occurred) then safe manipulation is not possible; otherwise it is possible.

Proof. The winner is constant on each interval, so we need only check one point in each interval, plus endpoints to deal with ties. There are at most $m(m-1)/2$ intersections of the lines which are the graphs of the functions $x \mapsto |c|_x$ for $0 \leq x \leq n_t$. The condition on maximum good and bad scores can be checked for each interval in time proportional to m. □

Corollary 1. *When $m = 3$, we need only calculate which candidate wins when $x = n_t$, and safe manipulation is possible if and only if the winner is good.*

3.2 The general case

When at least one of t and t' is not specified, there are obviously more possibilities, and a brute force approach that simply tries each pair (t, t') in turn will work. However, we can clearly do better than this.

There are some values of t for which S_t is empty. This means that no matter what the situation and the differences in the sincere scores, safe manipulation is impossible by type t. For example, every t for which the sincere winner 1 is ranked first has no incentive to manipulate. Other types have incentive but as we see in Example 3, S_t may still be empty.

For those t for which S_t is nonempty, we can still remove strategies t' for which $S_{t,t'}$ is empty. Similarly, we can express the union defining S_t with as few terms as possible. This is done by discarding dominated strategies (in any particular voting situation, even more strategies may be dominated, but we consider here those that are never worth including for any situation). For example, any type that ranks a bad candidate ahead of a good one is dominated by the type that differs only by transposing those two candidates. Thus all good candidates should be ranked ahead of all bad ones. The sincere winner should not be ranked ahead of any good candidate for the same reason. Furthermore, each strategy that does not allow some good candidate to catch the sincere winner should be rejected, as should each strategy that further advantages a bad candidate higher in the social ordering over all good candidates.

Example 3 ($m = 3$)**.** *Consider type* 312*. The only possibly undominated strategy that we need to consider, according to the above discussion, is* 321*. However* 321 *cannot lead to successful manipulation, as it increases the score of* 2 *and not of* 3*. Thus type* 312 *cannot manipulate at all, let alone safely. Types* $231, 213$ *and* 321 *have respectively the strategies* $321, 231, 231$ *available.*

Example 4. *When* $m = 4$*, the strategies that are worth considering in some situation are as follows. For any type starting with* 1*, only the sincere strategy. For any type ending with* 1*, any strategy that keeps* 1 *at the bottom. For types starting* 41*, only the sincere strategy; for types starting* 31*, any strategy that lowers* 1 *while keeping* 3 *at the top and not promoting* 2*; for types starting* 21*, any strategy that lowers* 1*, keeping* 2 *first. For types ranking* 1 *third, transpose the two good candidates.*

When there are very few distinct entries in w, there are many fewer strategies to consider. The extreme cases are plurality ($w = (1, 0, \ldots, 0)$) and antiplurality $(1, 1, \ldots, 1, 0)$). For plurality (respectively antiplurality), safe manipulation is possible by a type t voter if and only if it is possible by ranking some good candidate first (respectively some bad candidate last). The player is indifferent between the different strategies satisfying this criterion (if the good candidate is fixed) and the analysis does not distinguish between them, so we can assume that any such voter uses a standard strategy that makes a chosen good candidate the favoured one and orders the others by increasing value of index. Thus, for example, for $m = 3$ under plurality we consider 213 and 312 as possible values for t'.

We have so far expressed S_t in terms of a union of $S_{t,t'}$ which is as small as possible. However the terms in the union may not be disjoint. For example, with $m \geq 4$ a voter of type ranking c_1 last may use any of the $(m-1)! - 1$ insincere strategies that leave c_1 at the bottom (when $m = 3$ there is only one such strategy).

To compute the final probability of safe manipulation, we need to compute the volume of the union of all S_t. This union is in general not disjoint even for $m = 3$, as the following example shows.

Example 5. *Let* $m = 3$ *and consider the voting situation with* 3 *agents having preference* 123*,* 2 *having preference* 231 *and* 2 *having preference* 321*. Under the plurality rule, the last two types can each manipulate safely.*

We use inclusion-exclusion to compute the volume of the union. The number of terms in the inclusion-exclusion formula is $2^p - 1$ where p is the number of types involved.

4 Numerical results

We restrict to $m = 3$ and some selected scoring rules including the commonly studied plurality, Borda ($w = (2, 1, 0)$), and antiplurality.

For a situation in which the sincere result is 123, types $123, 132$ and 312 cannot manipulate safely. We need deal with only the remaining types, each of which has only one insincere strategy to consider. The linear systems in question are as follows. We denote $w_i - w_j$ by w_{ij}.

The fact that 123 is the sincere result is expressed as $|c_1| \geq |c_2| \geq |c_3|$. This translates to

$$\begin{aligned} 0 &\leq n_1 w_{12} + n_2 w_{13} + n_3 w_{21} + n_4 w_{31} + n_5 w_{23} + n_6 w_{32} \\ 0 &\leq n_1 w_{23} + n_2 w_{32} + n_3 w_{13} + n_4 w_{12} + n_5 w_{31} + n_6 w_{21} \\ n_i &\geq 0 \text{ for all } i \\ n &= n_1 + \cdots + n_6. \end{aligned}$$

For type 213, safe manipulation is possible if and only the following additional conditions are satisfied.

$$\begin{aligned} |c_2| &\geq |c_1| - n_3 w_{23} \\ |c_2| &\geq |c_3| + n_3 w_{23} \end{aligned}$$

which simplifies to the following system.

$$\begin{aligned} 0 &\geq n_1 w_{12} + n_2 w_{13} + n_3 w_{31} + n_4 w_{31} + n_5 w_{23} + n_6 w_{32} \\ 0 &\leq n_1 w_{23} + n_2 w_{32} + n_3 w_{12} + n_4 w_{12} + n_5 w_{31} + n_6 w_{21} \end{aligned}$$

Every voting situation can be represented in this way up to a permutation of alternatives.

Thus the asymptotic probability under IAC that type 213 can safely manipulate is given by the ratio of the volume of the "strategic" polytope to that of the "sincere" polytope. A completely analogous method works for other types. The volumes can be computed using standard software as described in [9, 3].

The results for several voting rules are shown in Table 1. The column labelled "P(manip)" gives the asymptotics probability of a voting situation begin manipulable (possibly by a coalition of more than one type) and was computed using the methods in [4] (note that the results for plurality, antiplurality and Borda have been computed exactly elsewhere [9]). Note that the ordering of rules according to their susceptibility to manipulation and the corresponding order for safe manipulation differ. Also the entries in the last column, giving conditional probabilities, are decreasing. This last fact is not surprising in hindsight and probably not dependent on the culture IAC. For example, plurality allows only one type of member in a minimal manipulating coalition, and such members have nothing to lose, so manipulation is possible if and only if it is safely possible. At the other extreme, only one type of voter can manipulate under antiplurality, but whether this is safe or not depends strongly on the voting situation.

The Borda rule is often criticized for its susceptibility to manipulation. While it is still the most manipulable here by both measures, it is clear that many manipulable situations under Borda require unsafe manipulations. The plurality rule seems the least manipulable when complicated coalitions are used, but its advantage disappears when safety is considered. These results, which of course depend on the particular distribution IAC, nevertheless indicate that when communication is restricted, traditional ratings of voting rules may need to be revised.

Table 2 shows the probability that a given rule is safely manipulable by all of the individual types listed. We see for example that type 213 has the most manipulating power under the $(3, 2, 0)$ rule, whereas 231 and 321 are strongest under plurality. Note that, for

Table 1: Asymptotic probability under IAC of a situation being (safely) manipulable.

scoring rule	P(manip)	P(safely)	P (safely | manip)
Plurality	0.292	0.292	1.00
(3,1,0)	0.422	0.322	0.76
Borda	0.502	0.347	0.69
(3,2,0)	0.535	0.330	0.62
(10,9,0)	0.533	0.264	0.49
Antiplurality	0.525	0.222	0.42

Table 2: Asymptotic probability under IAC of safe manipulation by various types

scoring rule	213	231	321	213, 231	213, 321	231, 321	213, 231, 321
Plurality	0.0000000	0.156250	0.246528	0.000000	0.000000	0.111111	0.000000
(3,1,0)	0.178369	0.086670	0.216913	0.000080	0.104229	0.053084	0.000067
Borda	0.225000	0.047950	0.196759	0.000033	0.093542	0.027400	0.000024
(3,2,0)	0.239297	0.020019	0.152812	0.000007	0.070438	0.010926	0.000005
(10,9,0)	0.234375	0.001687	0.051107	0.000000	0.022681	0.000866	0.000000
Antiplurality	0.2222222	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

example, there is an appreciable probability that both types 213 and 321 can manipulate safely. If each proceeds, ignoring the other, the result may no longer be safe. On the other hand, if both 231 and 321 try simultaneously to manipulate safely, the cancellation effect means that they are less likely to be disappointed.

5 Further discussion

The uniform distribution on profiles (the Impartial Culture hypothesis) has been used in many analyses in voting theory, because of its analytical tractability. However, for the asymptotic study of safe manipulation it seems less useful. This is because under IC for scoring rules, much weight is placed on situations that are nearly tied: a typical situation has almost equal numbers of each type, and the differences between the scores are of order $\sqrt{n}$. Thus as $n \to \infty$ the probability that, for example, a voter of type 321 can safely manipulate will approach 1 rapidly, while the probability that a type 213 can do so will approach 0 rapidly.

The inclusion-exclusion procedure used is probably exponential in m, since the number p of types used seems to grow linearly in m (we have not formally proved this). Thus a better algorithm is needed for large m.

As pointed out by the referee, the argument of Section 3.2 involve a monotonicity property that should be satisfied by more than just the scoring rules, but we have not pursued such a generalization here, leaving it for possible future work.

The literature on safe manipulation is very small still - our literature search turned up only one preprint of unknown publication status, dealing with complexity issues (though a similar idea was apparently used in [1] without explicit mention). However the basic model is attractive and some obvious generalizations should be investigated. For example, we can use a probability distribution to model the number of followers, instead of considering the worst case outcome, and thereby consider whether strategic voting even with lack of coordination can lead to better outcomes in the sense of expected utility.

We thank the referee for several useful comments.

References

[1] Pierre Favardin and Dominique Lepelley. Some further results on the manipulability of social choice rules. *Soc. Choice Welf.*, 26(3):485–509, 2006.

[2] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41:587–601, 1973.

[3] Dominique Lepelley, Ahmed Louichi, and Hatem Smaoui. On Ehrhart polynomials and probability calculations in voting theory. *Soc. Choice Welf.*, 30(3):363–383, April 2008.

[4] Geoffrey Pritchard and Mark C. Wilson. Exact results on manipulability of positional voting rules. *Soc. Choice Welf.*, 29(3):487–513, 2007.

[5] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *J. Econom. Theory*, 10(2):187–217, 1975.

[6] Arkadii Slinko. How large should a coalition be to manipulate an election? *Math. Social Sci.*, 47(3):289–293, 2004.

[7] Arkadii Slinko. How the size of a coalition affects its chances to influence an election. *Soc. Choice Welf.*, 26(1):143–153, 2006.

[8] Arkadii Slinko and Shaun White. Is it ever safe to vote strategically? Technical Report 563, University of Auckland Department of Mathematics Report Series, 2008.

[9] Mark C. Wilson and Geoffrey Pritchard. Probability calculations under the IAC hypothesis. *Math. Social Sci.*, 54(3):244–256, 2007.

Mark C. Wilson
Department of Computer Science
University of Auckland
Email: `mcw@cs.auckland.ac.nz`

Reyhaneh Reyhani
Department of Computer Science
University of Auckland
Email: `rrey015@aucklanduni.ac.nz`

Socially Desirable Approximations for Dodgson's Voting Rule[1]

Ioannis Caragiannis, Christos Kaklamanis,
Nikos Karanikolas, and Ariel D. Procaccia

Abstract

In 1876 Charles Lutwidge Dodgson suggested the intriguing voting rule that today bears his name. Although Dodgson's rule is one of the most well-studied voting rules, it suffers from serious deficiencies, both from the computational point of view—it is $\mathcal{NP}$-hard to approximate the Dodgson score to logarithmic factors—and from the social choice point of view—it fails basic social choice desiderata such as monotonicity and homogeneity.

In a previous paper [Caragiannis et al., SODA 2009] we have asked whether there are approximation algorithms for Dodgson's rule that are monotonic or homogeneous. In this paper we give definitive answers to these questions. We design a monotonic exponential-time algorithm that yields a 2-approximation to the Dodgson score, while matching this result with a tight lower bound. We also present a monotonic polynomial-time $\mathcal{O}(\log m)$-approximation algorithm (where m is the number of alternatives); this result is tight as well due to a complexity-theoretic lower bound. Furthermore, we show that a slight variation on a known voting rule yields a monotonic, homogeneous, polynomial-time $\mathcal{O}(m \log m)$-approximation algorithm, and establish that it is impossible to achieve a better approximation ratio even if one just asks for homogeneity. We complete the picture by studying several additional social choice properties; for these properties, we prove that algorithms with an approximation ratio that depends only on m do not exist.

1 Introduction

Social choice theory is concerned with aggregating the preferences of a set of n *agents* over a set of m *alternatives*. It is often assumed that each agent holds a private ranking of the alternatives; the collection of agents' rankings is known as a *preference profile*. The preference profile is reported to a *voting rule*, which then singles out the winning alternative.

When there are two alternatives (and an odd number of agents), majority voting is unanimously considered a perfect method of selecting the winner. However, when there are at least three alternatives it is sometimes unclear which alternative is best. In the Eighteenth Century the marquis de Condorcet, perhaps the founding father of the mathematical theory of voting, suggested a solution by extending majority voting to multiple alternatives [10]. An alternative x is said to beat alternative y in a *pairwise election* if a majority of agents prefer x to y, i.e., rank x above y. An alternative that beats every other alternative in a pairwise election is easy to accept as the winner of the entire election; in the modern literature such an alternative is known as a *Condorcet winner*. Unfortunately, there are preference profiles for which no alternative is a Condorcet winner.

Almost a century after Condorcet, a refinement of Condorcet's ideas was proposed by Charles Lutwidge Dodgson (today better known by his pen name Lewis Carroll), despite apparently being unfamiliar with Condorcet's work [5]. Dodgson proposed selecting the alternative "closest" to being a Condorcet winner, in the following sense. The *Dodgson score* of an alternative x is the number of exchanges between adjacent alternatives in the

[1]A slightly different version of the paper appeared in *Proceedings of the 10th ACM Conference on Electronic Commerce (EC 10).*

agents' rankings that must be introduced in order for x to become a Condorcet winner (see Section 2 for an example). A *Dodgson winner* is an alternative with minimum Dodgson score.

Although Dodgson's rule is intuitively appealing, it has been heavily criticized over the years for failing to satisfy desirable properties that are considered by social choice theorists to be extremely basic. Most prominent among these properties are *monotonicity* and *homogeneity*; a voting rule is said to be *monotonic* if it is indifferent to pushing a winning alternative upwards in the preferences of the agents, and is said to be *homogeneous* if it is invariant under duplication of the electorate. In fact, several authors have commented that it is somewhat unfair to attribute the abovementioned rule to Dodgson, since Dodgson himself seems to have questioned it due to its serious defects (see, e.g., the papers by Tideman [21, p. 194] and Fishburn [11, p. 474]).

To make matters worse, the rise of computational complexity theory, a century after the conception of Dodgson's rule, has made it clear that it suffers from yet another serious deficiency: it is intractable to single out the winner of the election. Indeed, it is the first voting rule where winner determination was known to be $\mathcal{NP}$-hard [4]; even the computation of the Dodgson score of a given alternative is $\mathcal{NP}$-hard. The question of the exact complexity of winner determination under Dodgson's rule was resolved by Hemaspaandra et al. [13]: it is complete for the class Θ_2^p. These results have sparked great interest in Dodgson's rule among computer scientists, making it "one of the most studied voting rules in computational social choice" [6].

In previous work with numerous colleagues [8], we have largely taken the computational complexity point of view by considering the computation of the Dodgson score as an optimization problem. Among other results, we have given two polynomial-time algorithms that guarantee an approximation ratio of $\mathcal{O}(\log m)$ to the Dodgson score (where m is the number of alternatives); this bound is asymptotically tight with respect to polynomial-time algorithms (unless $\mathcal{P} = \mathcal{NP}$).

Taking the social choice point of view, our main conceptual contribution in [8] was the suggestion that an algorithm that approximates the Dodgson score is a voting rule in its own right in the sense that it naturally induces a voting rule that selects an alternative with minimum score according to the algorithm. Hence, such algorithms should be evaluated not only by their computational properties (e.g., approximation ratio and complexity) but also by their social choice properties (e.g., monotonicity and homogeneity). In other words, they should be "socially desirable". This issue was very briefly explored in the foregoing paper: we have shown that one of our two approximation algorithms satisfies a weak flavor of monotonicity, whereas the other does not. Both algorithms, as well as Dodgson's rule itself, are neither monotonic (in the usual sense) nor homogeneous, but this does not preclude the existence of monotonic or homogeneous approximation algorithms for the Dodgson score. Indeed, we have asked whether there exist such algorithms that yield a good approximation ratio [8, p. 1064].

In the following, we refer to algorithms approximating the Dodgson score (as well as to the voting rules they induce) using the term *Dodgson approximations*. A nice property that Dodgson approximations enjoy is that a finite approximation ratio implies *Condorcet-consistency*, i.e., a Condorcet winner (if one exists) is elected as the unique winner. One might wish for approximations of the Dodgson ranking (i.e., the ranking of the alternatives with respect to their Dodgson scores) directly instead of approximating the Dodgson score. Unfortunately, it is known that distinguishing whether an alternative is the Dodgson winner or in the last $\mathcal{O}(\sqrt{m})$ positions in the Dodgson ranking is $\mathcal{NP}$-hard [8]. This extreme inapproximability result provides a complexity-theoretic explanation of the discrepancies that have been observed in the social choice literature when comparing Dodgson's rule to simpler polynomial-time voting rules (see the discussion in [8]) and implies that, as long as we care about efficient algorithms, reasonable approximations of the Dodgson ranking are impossible. However, the cases where the ranking is hard to approximate are cases where the alternatives have very similar Dodgson scores. We would argue that in those cases it is not crucial, from Dodgson's point of view, which alternative is elected, since they are all almost equally close to being Condorcet winners. Put another way, if the Dodgson score is a measure of an alternative's quality, the goal is simply to elect a good alternative according to this measure.

Our results and techniques. In this paper we give definitive (and mostly positive) answers to the questions raised above; our results are tight. Due to lack of space, all proofs have been omitted.

In Section 3 we study monotonic Dodgson approximations. We first design an algorithm that we denote by M. Roughly speaking, this algorithm "monotonizes" Dodgson's rule by explicitly defining a winner set for each given preference profile, and assigning an alternative to the winner set if it is a Condorcet winner in some preference profile such that the former profile is obtained from the latter by pushing the alternative upwards. We prove the following result.

Theorem 3.1. *M is a monotonic Dodgson approximation with an approximation ratio of 2.*

We furthermore show that there is no monotonic Dodgson approximation with a ratio smaller than 2 (Theorem 3.2), hence M is optimal among monotonic Dodgson approximations. Note that the lower bound is independent of computational assumptions, and, crucially, computing an alternative's score under M requires exponential time. This is to be expected since the Dodgson score is computationally hard to approximate within a factor better than $\Omega(\log m)$ [8].

It is now natural to ask whether there is a monotonic *polynomial-time* Dodgson approximation with an approximation ratio of $\mathcal{O}(\log m)$. We give a positive answer to this question as well. Indeed, we design a Dodgson approximation denoted by Q, and establish the following result.

Theorem 3.3. *Q is a monotonic polynomial-time Dodgson approximation with an approximation ratio of $\mathcal{O}(\log m)$.*

The result relies on monotonizing an existing Dodgson approximation that is based on linear programming. The main obstacle is to perform the monotonization in polynomial time rather than looking at an exponential number of profiles, as described above. Our main tool is the notion of *pessimistic estimator*, which allows the algorithm to restrict its attention to a single preference profile. Pessimistic estimators are obtained by solving a linear program that is a variation on the one that approximates the Dodgson score.

In Section 4 we turn to homogeneity. We consider Tideman's *simplified Dodgson rule* [22, pages 199-201], which was designed to overcome the deficiencies of Dodgson's rule. The former rule is computable in polynomial time, and is moreover known to be monotonic and homogeneous. By scaling the score given by the simplified Dodgson rule we obtain a rule, denoted Td', that is identical as a voting rule, and moreover has the following properties.

Theorem 4.1. *Td' is a monotonic, homogeneous, polynomial-time Dodgson approximation with an approximation ratio of $\mathcal{O}(m \log m)$.*

Note that the Dodgson score can be between 0 and $\Theta(nm)$, so this bound is far from trivial. The analysis is tight when there is an alternative that is tied against many other alternatives in pairwise elections (and hence has relatively high Dodgson score), whereas another alternative strictly loses in pairwise elections to few alternatives (so it has relatively low Dodgson score). By homogeneity the former alternative must be elected, since its score does not scale when the electorate is replicated (we elaborate in Section 4). This intuition leads to the following result which applies to any (even exponential-time) homogeneous Dodgson approximation.

Theorem 4.2. *Any homogeneous Dodgson approximation has approximation ratio at least* $\Omega(m \log m)$.

In particular the homogeneous upper bound given in Theorem 4.1 (which is achieved by an algorithm that is moreover monotonic and efficient) is asymptotically tight. The heart of our construction is the design of a preference profile such that an alternative is tied against $\Omega(m)$ other alternatives; this is equivalent to a construction of a family of subsets of a set U, $|U| = m$, such that each element of U appears in roughly half the subsets but the minimum cover is of size $\Omega(\log m)$.

In order to complete the picture, in Section 5 we discuss some other, less prominent, social choice properties not satisfied by Dodgson's rule [22, Chapter 13]: combinativity, Smith consistency, mutual majority, invariant loss consistency, and independence of clones. We show that any Dodgson approximation that satisfies one of these properties has an approximation ratio of $\Omega(nm)$ (in the case of the former two properties) or $\Omega(n)$ (in the case of the latter three). An $\Omega(nm)$ ratio is a completely trivial one, but we also consider an approximation ratio of $\Omega(n)$ to be impractical, as the number of agents n is very large in almost all settings of interest.

Discussion. Our results with respect to monotonicity are positive across the board. In particular, we find Theorem 3.1 surprising as it indicates that Dodgson's lack of monotonicity can be circumvented by slightly modifying the definition of the Dodgson score; in a sense this suggests that Dodgson's rule is not fundamentally far from being monotonic. Theorem 3.3 provides a striking improvement over the main result of [8]. Indeed, if one is interested in computationally tractable algorithms then an approximation ratio of $\mathcal{O}(\log m)$ is optimal; the theorem implies that we can additionally satisfy monotonicity without (asymptotically) increasing the approximation ratio. Our monotonization techniques may be of independent interest.

Our results regarding homogeneity, Theorem 4.1 and Theorem 4.2, can be interpreted both positively and negatively. Consider first the case where the number of alternatives m is small (e.g., in political elections). A major advantage of Theorem 4.1 is that it concerns Tideman's simplified Dodgson rule, which is already recognized as a desirable voting rule, as it is homogeneous, monotonic, Condorcet-consistent, and resolvable in polynomial time. The theorem lends further justification to this rule by establishing that it always elects an alternative relatively close (according to Dodgson's notion of distance) to being a Condorcet winner, that is, the spirit of Dodgson's ideas is indeed preserved by the "simplification" and (due to Theorem 4.2) this is accomplished in the best possible way.

Viewed negatively, when the number of alternatives is large (an extreme case is a multiagent system where the agents are voting over joint plans), Theorem 4.2 strengthens the criticism against Dodgson's rule: not only is the rule itself nonhomogeneous, but any (even exponential-time computable) conceivable variation that tries to roughly preserve Dodgson's notion of proximity to Condorcet is also nonhomogeneous. We believe that both interpretations of the homogeneity results are of interest to social choice theorists as well as computer scientists.

As an aside, note that almost all work in *algorithmic mechanism design* [18] seeks *truthful* approximation algorithms, that is, algorithms such that the agents cannot benefit by lying. However, it is well known that in the standard social choice setting, truthfulness cannot be achieved. Indeed, the Gibbard-Satterthwaite Theorem [12, 19] (see also [17]) implies that any minimally reasonable voting rule is not truthful. Therefore, social choice theorists strive for other socially desirable properties, and in particular the ones discussed above. To avoid confusion, we remark that although notions of monotonicity are often studied in mechanism design as ways of obtaining truthfulness (see, e.g., [3]), in social choice theory monotonicity is a very basic desirable property in its own right (and has been so long before mechanism design was conceived).

Future work. In the future, we envision the extension of our agenda of socially desirable approximation algorithms to other important voting rules. Positive results in this direction would provide us with tools to circumvent the deficiencies of known voting rules without sacrificing their core principles; negative results would further enhance our understanding of such deficiencies. Note that these questions are relevant even with respect to tractable voting rules that do not satisfy certain properties, but seem especially interesting in the context of voting and rank aggregation rules that are hard to compute, e.g., Kemeny's and Slater's rules [1, 9, 15]. The work in this direction might involve well-known tractable, Condorcet-consistent, monotonic, and homogeneous rules such as Copeland and Maximin (see, e.g., [22]) in the same way that we use Tideman's simplified Dodgson rule in the current paper.

2 Preliminaries

We consider a set of *agents* $N = \{0, 1, \ldots, n-1\}$ and a set of *alternatives* A, $|A| = m$. Each agent has linear preferences over the alternatives, that is, a ranking over the alternatives. Formally, the preferences of agent i are a binary relation $\succ_i$ over A that satisfies irreflexivity, asymmetry, transitivity and totality; given $x, y \in A$, $x \succ_i y$ means that i prefers x to y. We let $\mathcal{L} = \mathcal{L}(A)$ be the set of linear preferences over A. A *preference profile* $\succ = \langle \succ_0, \ldots, \succ_{n-1} \rangle \in \mathcal{L}^n$ is a collection of preferences for all the agents. A *voting rule* (also known as a *social choice correspondence*) is a function $f : \mathcal{L}^n \to 2^A \setminus \{\emptyset\}$ from preference profiles to nonempty subsets of alternatives, which designates the winner(s) of the election.

Let $x, y \in A$, and $\succ \in \mathcal{L}^n$. We say that x beats y in a *pairwise election* if $|\{i \in N : x \succ_i y\}| > n/2$, that is, if a (strict) majority of agents prefer x to y. A *Condorcet winner* is an alternative that beats every other alternative in a pairwise election. The *Dodgson score* of an alternative $x \in A$ with respect to a preference profile $\succ \in \mathcal{L}^n$, denoted $\mathrm{sc}_D(x, \succ)$, is the number of swaps between adjacent alternatives in the individual rankings that are required in order to make it a Condorcet winner. A *Dodgson winner* is an alternative with minimum Dodgson score.

Consider, for example, the profile $\succ$ in Table 1; in this example $N = \{0, \ldots, 4\}$, $A = \{a, b, c, d, e\}$, and the ith column is the ranking reported by agent i. Alternative a loses in pairwise elections to b and e (two agents prefer a to b, one agent prefers a to e). In order to become a Condorcet winner, four swaps suffice: swapping a and e, and then a and b, in the ranking of agent 1 (after the swaps the ranking becomes $a \succ_1 b \succ_1 e \succ_1 c \succ_1 d$), and swapping a and d, and then a and e, in the ranking of agent 4. Agent a cannot be made a Condorcet winner with fewer swaps, hence we have $\mathrm{sc}_D(a, \succ) = 4$ in this profile. However, in the profile of Table 1 there is a Condorcet winner, namely agent b, hence b is the Dodgson winner with $\mathrm{sc}_D(b, \succ) = 0$.

Given a preference profile $\succ \in \mathcal{L}^n$ and $x, y \in A$, the *deficit* of x against y, denoted $\mathrm{defc}(x, y, \succ)$, is the number of additional agents that must rank x above y in order for x to beat y in a pairwise election. Formally,

$$\mathrm{defc}(x, y, \succ) \quad = \quad \max\left\{0, \left\lceil \frac{n+1}{2} \right\rceil - |\{i \in N : x \succ_i y\}| \right\}.$$

In particular, if x beats y in a pairwise election then it holds that $\mathrm{defc}(x, y, \succ) = 0$. Note that

0	1	2	3	4
a	b	e	e	b
b	e	b	c	e
c	a	c	d	d
d	c	a	a	a
e	d	d	b	c

Table 1: An example of the Dodgson score. For this profile $\succ$, it holds that $\mathrm{sc}_D(b, \succ) = 0$, $\mathrm{sc}_D(a, \succ) = 4$.

if n is even and x and y are tied, that is, $|\{i \in N : x \succ_i y\}| = n/2$, then $\mathrm{defc}(x, y, \succ) = 1$. For example, in the profile of Table 1 we have that $\mathrm{defc}(a, b, \succ) = 1$, $\mathrm{defc}(a, c, \succ) = 0$, $\mathrm{defc}(a, d, \succ) = 0$, $\mathrm{defc}(a, e, \succ) = 2$.

We consider algorithms that receive as input an alternative $x \in A$ and a preference profile $\succ \in \mathcal{L}^n$, and return a score for x. We denote the score returned by an algorithm V on the input which consists of an alternative $x \in A$ and a profile $\succ \in \mathcal{L}^n$ by $\mathrm{sc}_V(x, \succ)$. We call such an algorithm V a *Dodgson approximation* if $\mathrm{sc}_V(x, \succ) \geq \mathrm{sc}_D(x, \succ)$ for every alternative $x \in A$ and every profile $\succ \in \mathcal{L}^n$. We also say that V has an *approximation ratio* of ρ if $\mathrm{sc}_D(x, \succ) \leq \mathrm{sc}_V(x, \succ) \leq \rho \cdot \mathrm{sc}_D(x, \succ)$, for every $x \in A$ and every $\succ \in \mathcal{L}^n$. A Dodgson approximation naturally induces a voting rule by electing the alternative(s) with minimum score. Hence, when we say that a Dodgson approximation satisfies a social choice property we are referring to the voting rule induced by the algorithm. Observe that the voting rule induced by a Dodgson approximation with finite approximation ratio is *Condorcet-consistent*, i.e., it always elects a Condorcet winner as the sole winner if one exists.

Let us give an example. Consider the algorithm V that, given an alternative $x \in A$ and a preference profile $\succ \in \mathcal{L}^n$, returns a score of $\mathrm{sc}_V(x, \succ) = m \cdot \sum_{y \in A \setminus \{x\}} \mathrm{defc}(x, y, \succ)$. It is easy to show that this algorithm is a Dodgson approximation and, furthermore, has approximation ratio at most m. Indeed, it is possible to make x beat y in a pairwise election by pushing x to the top of the preferences of $\mathrm{defc}(x, y, \succ)$ agents, and this requires at most $m \cdot \mathrm{defc}(x, y, \succ)$ swaps. By summing over all $y \in A \setminus \{x\}$, we obtain an upper bound of $\mathrm{sc}_V(x, \succ)$ on the Dodgson score of x. On the other hand, given $x \in A$, for every $y \in A \setminus \{x\}$ we require $\mathrm{defc}(x, y, \succ)$ swaps that push x above y in the preferences of some agent in order for x to beat y in a pairwise election. Moreover, these swaps do not decrease the deficit against any other alternative. Therefore, $\sum_{y \in A \setminus \{x\}} \mathrm{defc}(x, y, \succ) \leq \mathrm{sc}_D(x, \succ)$, and by multiplying by m we get that $\mathrm{sc}_V(x, \succ) \leq m \cdot \mathrm{sc}_D(x, \succ)$.

3 Monotonicity

In this section we present our results on monotonic Dodgson approximations. A voting rule is *monotonic* if a winning alternative remains winning after it is pushed upwards in the preferences of some of the agents. Dodgson's rule is known to be non-monotonic (see, e.g., [6]). The intuition is that if an agent ranks x directly above y and y above z, swapping x and y may not help y if it already beats x, but may help z defeat x.

As a warm-up we observe that the Dodgson approximation mentioned at the end of the previous section is monotonic as a voting rule. Indeed, consider a preference profile $\succ$ and a winning alternative x. Pushing x upwards in the preference of some of the agents can neither increase its score (since its deficit against any other alternative does not increase) nor decrease the score of any other alternative $y \in A \setminus \{x\}$ (since the deficit of y against any alternative in $A \setminus \{x, y\}$ remains unchanged and its deficit against x does not decrease).

3.1 Monotonizing Dodgson's Voting Rule

In the following we present a much stronger result. Using a natural monotonization of Dodgson's voting rule, we obtain a monotonic Dodgson approximation with approximation ratio at most 2. The main idea is to define the winning set of alternatives for a given profile first and then assign the same score to the alternatives in the winning set and a higher score to the non-winning alternatives. Roughly speaking, the winning set is defined so that it contains the Dodgson winners for the given profile as well as the Dodgson winners of other profiles that are necessary so that monotonicity is satisfied.

More formally, we say that a preference profile $\succ' \in \mathcal{L}^n$ is a *y-improvement* of $\succ$ for some alternative $y \in A$ if $\succ'$ is obtained by starting from $\succ$ and pushing y upwards in the preferences of some of the agents. In particular a profile is a y-improvement of itself for any alternative $y \in A$.

We monotonize Dodgson's voting rule as follows. Let M denote the new voting rule we are constructing. We denote by $W(\succ)$ the set of winners of M for profile $\succ \in \mathcal{L}^n$. Let $\Delta = \max_{y \in W(\succ)} \mathrm{sc}_D(y, \succ)$. The voting rule M assigns a score of $\mathrm{sc}_M(y, \succ) = \Delta$ to each alternative $y \in W(\succ)$ and a score of

$$\mathrm{sc}_M(y, \succ) = \max\{\Delta + 1, \mathrm{sc}_D(y, \succ)\}$$

to each alternative $y \notin W(\succ)$. All that remains is to define the set of winners $W(\succ)$ for profile $\succ$. This is done as follows: for each preference profile $\succ^* \in \mathcal{L}^n$ and each Dodgson winner y^* at $\succ^*$, include y^* in the winner set $W(\succ')$ of each preference profile $\succ' \in \mathcal{L}^n$ that is a y^*-improvement of $\succ^*$.

Theorem 3.1. *M is a monotonic Dodgson approximation with an approximation ratio of 2.*

In general, the Dodgson approximation M is computable in exponential time. However, it can be implemented to run in polynomial time when m is a constant; in this special case the number of different profiles with n agents is polynomial and the Dodgson score can be computed exactly in polynomial time [4].

The next statement shows that the voting rule M is the best possible monotonic Dodgson approximation. Note that it is not based on any complexity assumptions and, hence, it holds for exponential-time Dodgson approximations as well.

Theorem 3.2. *A monotonic Dodgson approximation cannot have an approximation ratio smaller than 2.*

3.2 A Monotonic Polynomial-Time $\mathcal{O}(\log m)$-Approximation Algorithm

In the following we present a monotonic polynomial-time Dodgson approximation that achieves an approximation ratio of $\mathcal{O}(\log m)$. Given the $\Omega(\log m)$ inapproximability bound for the Dodgson score [8], this rule is asymptotically optimal with respect to polynomial-time algorithms. To be precise, it is optimal within a factor of 4, assuming that problems in $\mathcal{NP}$ do not have quasi-polynomial-time algorithms.

In general, there are two main obstacles that we have to overcome in order to implement the monotonization in polynomial time. First, the computation of the Dodgson score and the decision problem of detecting whether a given alternative is a Dodgson winner on a particular profile are $\mathcal{NP}$-hard problems [4]. We overcome this obstacle by using a polynomial-time Dodgson approximation R instead of the Dodgson score itself. Even in this case, given a profile, we still need to be able to detect whether an alternative $y \in A$ is the winner according to R at some profile of which the current profile is a y-improvement; if this is the case, y should be included in the winning set. Note that, in general, this requires checking an exponential number of profiles in order to determine the winning set of the current one. We tackle this second obstacle using the notion of *pessimistic estimators*; these are quantities defined in terms of the current profile only and are used to identify its winning alternatives.

In order to define the algorithm R that we will monotonize we consider an alternative definition of the Dodgson score for an alternative $z^* \in A$ and a profile $\succ \in \mathcal{L}^n$. Define the set $S_k^{\succ_i}$ to be the set of alternatives z^* bypasses as it is pushed k positions upwards in the preference of agent i. Denote by $\mathcal{S}^{\succ_i}$ the collection of all possible such sets for agent i, i.e.,

$$\mathcal{S}^{\succ_i} = \{S_k^{\succ_i} : k = 1, ..., r_i(z^*, \succ) - 1\},$$

where $r_i(z^*, \succ)$ denotes the rank of alternative z^* in the preferences of agent $i \in N$ (e.g., the most and least preferred alternatives have rank 1 and m, respectively). Let $\mathcal{S} = \bigcup_{i \in N} \mathcal{S}^{\succ_i}$. Then, the problem of computing the Dodgson score of alternative z^* on the profile $\succ$ is equivalent to selecting sets from $\mathcal{S}$ of minimum total size so that at most one set is selected among the ones in $\mathcal{S}^{\succ_i}$ for each agent $i \in N$ and each alternative $z \in A \setminus \{z^*\}$ appears in at least $\mathrm{defc}(z^*, z, \succ)$ selected sets. This can be expressed by the following integer linear program:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i \in N} \sum_{k=1}^{r_i(z^*,\succ)-1} k \cdot \mathbf{x}\left(S_k^{\succ_i}\right) \qquad (1)\\
\text{subject to} \quad & \forall z \in A \setminus \{z^*\}, \\
& \sum_{i \in N} \sum_{S \in \mathcal{S}^{\succ_i} : z \in S} \mathbf{x}(S) \geq \mathrm{defc}(z^*, z, \succ) \\
& \forall i \in N, \ \sum_{S \in \mathcal{S}^{\succ_i}} \mathbf{x}(S) \leq 1 \\
& \forall S \in \mathcal{S}, \ \mathbf{x}(S) \in \{0, 1\}
\end{aligned}$$

The binary variable $\mathbf{x}(S)$ indicates whether the set $S \in \mathcal{S}$ is selected ($\mathbf{x}(S) = 1$) or not ($\mathbf{x}(S) = 0$). Now, consider the LP relaxation of the above ILP in which the last constraint is relaxed to $\mathbf{x}(S) \geq 0$. We define the voting rule R that sets $\mathrm{sc}_R(z^*, \succ)$ equal to the optimal value of the LP relaxation multiplied by H_{m-1}, where H_k is the kth harmonic number. In [8] it is shown that

$$\mathrm{sc}_D(y, \succ) \leq \mathrm{sc}_R(y, \succ) \leq H_{m-1} \cdot \mathrm{sc}_D(y, \succ)$$

for every alternative $y \in A$, i.e., R is a Dodgson approximation with an approximation ratio of H_{m-1}.

We now present a new voting rule Q by monotonizing R. The voting rule Q defines a set of alternatives $W(\succ)$ that is the set of winners on a particular profile $\succ$. Then, it sets $\mathrm{sc}_Q(y, \succ) = 2 \cdot \mathrm{sc}_R(y^*, \succ)$ for each alternative $y \in W(\succ)$, where y^* is the winner according to the voting rule R. In addition, it sets $\mathrm{sc}_Q(y, \succ) = 2 \cdot \mathrm{sc}_R(y, \succ)$ for each alternative $y \notin W(\succ)$.

In order to define the set $W(\succ)$ we will use another (slightly different) linear program defined for two alternatives $y, z^* \in A$ and a profile $\succ \in \mathcal{L}^n$. The new LP has the same set of constraints as the relaxation of (1) used in the definition of $\mathrm{sc}_R(z^*, \succ)$ and the following objective function:

$$\text{minimize} \quad \sum_{i \in N} \sum_{k=1}^{r_i(z^*,\succ)-1} k \cdot \mathbf{x}\left(S_k^{\succ_i}\right) + \sum_{i \in N : y \succ_i z^*} \sum_{k=1}^{r_i(z^*,\succ)-r_i(y,\succ)-1} \mathbf{x}\left(S_k^{\succ_i}\right) \qquad (2)$$

We define the *pessimistic estimator* $\mathrm{pe}(z^*, y, \succ)$ for alternative $z^* \in A$ with respect to another alternative $y \in A \setminus \{z^*\}$ and a profile $\succ \in \mathcal{L}^n$ to be equal to the objective value of LP (2) multiplied by H_{m-1}. As will become apparent shortly, the pessimistic estimator $\mathrm{pe}(z^*, y, \succ')$ upper-bounds the score of alternative z^* under R on every profile $\succ$ such that $\succ'$ is a y-improvement of $\succ$, hence the pessimism with respect to estimating the score of z^*. These pessimistic estimators will be our main tool in order to monotonize R.

We are now ready to complete the definition of the voting rule Q. The set $W(\succ)$ is defined as follows. First, it contains all the winners according to voting rule R. An alternative y that is not a winning alternative according to R is included in the set $W(\succ)$ if $\mathrm{pe}(z, y, \succ) \geq \mathrm{sc}_R(y, \succ)$ for every alternative $z \in A \setminus \{y\}$.

Theorem 3.3. *Q is a monotonic polynomial-time Dodgson approximation with an approximation ratio of $2H_{m-1}$.*

4 Homogeneity

In this section we present our results on homogeneous Dodgson approximations. A voting rule is *homogeneous* if duplicating the electorate, that is, duplicating the preference profile, does not change the outcome of the election. An example (due to Brandt [6]) that demonstrates that Dodgson's rule fails homogeneity can be found in Table 2. The intuition is that if alternatives x and y are tied in a pairwise election, the deficit of x against y does not increase by duplicating the profile, whereas if x strictly loses to y in a pairwise election then the deficit scales with the number of copies.

$\times 2$	$\times 2$	$\times 2$	$\times 2$	$\times 2$	$\times 1$	$\times 1$
d	b	c	d	a	a	d
c	c	a	b	b	d	a
a	a	b	c	c	b	b
b	d	d	a	d	c	c

Table 2: An example that demonstrates that Dodgson's rule does not satisfy homogeneity. A column headed by $\times k$ represents k identical agents. In the above profile, a is the Dodgson winner with a score of 3. By duplicating the electorate three times we obtain a profile in which the winner is d with a score of 6.

4.1 The Simplified Dodgson Rule

Tideman [22, pages 199-201] defines the following simplified Dodgson rule and proves that it is monotonic and homogeneous. Consider a profile $\succ \in \mathcal{L}^n$. If an alternative is a Condorcet winner, then this alternative is the sole winner. Otherwise, the simplified Dodgson rule assigns a score to each alternative and the alternative with the minimum score wins. According to the simplified Dodgson rule, the score of an alternative x is

$$\mathrm{sc}_{\mathrm{Td}}(x, \succ) \quad = \sum_{y \in A \setminus \{x\}} \max\{0, n - 2 \cdot |\{i \in N : x \succ_i y\}|\}.$$

Observe that $\mathrm{sc}_{\mathrm{Td}}(x, \succ)$ can be smaller than the Dodgson score of x and, hence, this definition does not correspond to a Dodgson approximation. For example, in profiles with an even number of agents, $\mathrm{sc}_{\mathrm{Td}}(x, \succ)$ is 0 when x is tied against some alternatives and beats the rest. Hence, we present an alternative definition of the simplified Dodgson rule as a

Dodgson approximation by scaling the original definition. If an alternative x is a Condorcet winner, then it has score $\mathrm{sc}_{\mathrm{Td}'}(x, \succ) = 0$. Otherwise:

$$\mathrm{sc}_{\mathrm{Td}'}(x, \succ) \quad = \quad m \cdot \mathrm{sc}_{\mathrm{Td}}(x, \succ) + m(\log m + 1).$$

It is clear that this alternative definition is equivalent to the original one of the simplified Dodgson rule, in the sense that it elects the same set of alternatives. It is also clear that $\mathrm{sc}_{\mathrm{Td}}(x, \succ)$ can be computed in polynomial time, and, as mentioned above, Td is known to be monotonic and homogeneous. Hence, in order to prove the following theorem it is sufficient to prove that Td′ is a Dodgson approximation and to bound its approximation ratio.

Theorem 4.1. *Td′ is a monotonic, homogeneous, polynomial-time Dodgson approximation with an approximation ratio of $\mathcal{O}(m \log m)$.*

4.2 Lower Bound

We next show that Td′ is the asymptotically optimal homogeneous Dodgson approximation by proving a matching lower bound on the approximation ratio of homogeneous Dodgson approximations. The lower bound is not based on any complexity assumptions and holds for exponential-time Dodgson approximations as well. This is quite striking since, as stated in Theorem 4.1, Td′ is also monotonic and polynomial-time.

Theorem 4.2. *Any homogeneous Dodgson approximation has approximation ratio at least $\Omega(m \log m)$.*

The proof is based on the construction of a preference profile with an alternative $b \in A$ that defeats some of the alternatives in pairwise elections, and is tied against many others. Hence, it has a high Dodgson score. On the other hand, there is a second alternative that has a Dodgson score of two, simply because it has a deficit of two against another alternative. In order to obtain a good approximation ratio, the algorithm must not select b in this profile. However, when the profile is replicated, the Dodgson score of b does not increase: it is still tied against the same alternatives. In contrast, the Dodgson score of the other alternatives scales with the number of copies. By homogeneity, we cannot select b in the replicated profile, which yields the lower bound.

We can think of an agent as the subset of alternatives that are ranked above b. If b is tied against an alternative, then that alternative is a member of exactly half the subsets. The argument used in the proof of Theorem 4.1 implies that there is always a cover of logarithmic size; the proof of Theorem 4.2 establishes that this bound is tight. Indeed, the combinatorial core of the theorem's proof is the construction of a set cover instance with the following properties: each element of the ground set appears in roughly half the subsets, but every cover requires a logarithmic number of subsets. This (apparently novel) construction is due to Noga Alon [2].

5 Additional Properties

In this section we briefly summarize our results with respect to several additional social choice properties that are not satisfied by Dodgson's rule. In general, our lower bounds with respect to these properties are at least linear in n, the number of agents. Since n is almost always large, these results should strictly be interpreted as impossibility results, that is, normally an upper bound of $\mathcal{O}(n)$ is not useful. We now (informally) formulate the five properties in question; for more formal definitions the reader is referred to [22].

We say that a voting rule satisfies *combinativity* if, given two preference profiles where the rule elects the same winning set, the rule would also elect this winning set under the profile obtained from appending one of the original preference profiles to the other. Note that combinativity implies homogeneity.

A *dominating set* is a nonempty set of alternatives such that each alternative in the set beats every alternative outside the set in pairwise elections. The *Smith set* is the unique inclusion-minimal dominating set. A voting rule satisfies *Smith consistency* if winners under the rule are always contained in the Smith set.

We say that a voting rule satisfies *mutual majority consistency* if, given a preference profile where more than half the agents rank a subset of alternatives $X \subseteq A$ above $A \setminus X$, only alternatives from X can be elected. A voting rule satisfies *invariant loss consistency* if an alternative that loses to every other alternative in pairwise elections cannot be elected. Clearly, mutual majority consistency implies invariant loss consistency.

Independence of clones was introduced by Tideman [21]; see also the paper by Schulze [20]. For ease of exposition we use a slightly weaker definition previously employed by Brandt [6]; since we are proving a lower bound, a weaker definition only strengthens the bound. Given a preference profile, two alternatives $x, y \in A$ are considered *clones* if they are adjacent in the rankings of all the agents, that is, their order with respect to every alternative in $A \setminus \{x, y\}$ is identical everywhere. A voting rule is *independent of clones* if a losing alternative cannot be made a winning alternative by introducing clones.

We have the following theorem.

Theorem 5.1. *Let V be a Dodgson approximation. If V satisfies combinativity or Smith consistency, then its approximation ratio is at least $\Omega(nm)$. If V satisfies mutual majority consistency, invariant loss consistency, or independence of clones, then its approximation ratio is at least $\Omega(n)$.*

References

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 684–693, 2005.

[2] N. Alon. Private communication, 2009.

[3] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

[4] J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[5] D. Black. *Theory of Committees and Elections*. Cambridge University Press, 1958.

[6] F. Brandt. Some remarks on Dodgson's voting rule. *Mathematical Logic Quarterly*, 55(4):460–463, 2009.

[7] F. Brandt, F. Fischer, and P. Harrenstein. The computational complexity of choice sets. *Mathematical Logic Quarterly*, 55(4): 444-459, 2009.

[8] I. Caragiannis, J. A. Covey, M. Feldman, C. M. Homan, C. Kaklamanis, N. Karanikolas, A. D. Procaccia, and J. S. Rosenschein. On the approximability of Dodgson and Young elections. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1058–1067, 2009.

[9] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 776–782, 2006.

[10] Marquis de Condorcet. Essai sur l'application de l'analyse à la probabilité de décisions rendues à la pluralité de voix. Imprimerie Royal, 1785. Facsimile published in 1972 by Chelsea Publishing Company, New York.

[11] P. C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–487, 1977.

[12] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[13] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[14] C. Homan and L. A. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson election winners. *Journal of Heuristics*, 15(4):403–423, 2009.

[15] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 95–103, 2007.

[16] J. C. McCabe-Dansted, G. Pritchard, and A. M. Slinko. Approximability of Dodgson's rule. *Social Choice and Welfare*, 31(2):311–330, 2008.

[17] N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9. Cambridge University Press, 2007.

[18] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1–2):166–196, 2001.

[19] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[20] M. Schulze. A new monotonic and clone-independent single-winner election method. *Voting Matters*, 17:9–19, 2003.

[21] N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[22] N. Tideman. *Collective Decisions and Voting*. Ashgate, 2006.

Ioannis Caragiannis
Department of Computer Engineering
and Informatics & RACTI
University of Patras, Greece
Email: `caragian@ceid.upatras.gr`

Christos Kaklamanis
Department of Computer Engineering
and Informatics & RACTI
University of Patras, Greece
Email: `kakl@ceid.upatras.gr`

Nikos Karanikolas
Department of Computer Engineering
and Informatics & RACTI
University of Patras, Greece
Email: `nkaranik@ceid.upatras.gr`

Ariel D. Procaccia
School of Engineering and Applied Sciences
Harvard University, USA
Email: `arielpro@seas.harvard.edu`

Approximation Algorithms and Mechanism Design for Minimax Approval Voting[1]

Ioannis Caragiannis, Dimitris Kalaitzis, and Evangelos Markakis

Abstract

We consider approval voting elections in which each voter votes for a (possibly empty) set of candidates and the outcome consists of a set of k candidates for some parameter k, e.g., committee elections. We are interested in the minimax approval voting rule in which the outcome represents a compromise among the voters, in the sense that the maximum distance between the preference of any voter and the outcome is as small as possible. This voting rule has two main drawbacks. First, computing an outcome that minimizes the maximum distance is computationally hard. Furthermore, any algorithm that always returns such an outcome provides incentives to voters to misreport their true preferences.

In order to circumvent these drawbacks, we consider approximation algorithms, i.e., algorithms that produce an outcome that approximates the minimax distance for any given instance. Such algorithms can be considered as alternative voting rules. We present a polynomial-time 2-approximation algorithm that uses a natural linear programming relaxation for the underlying optimization problem and deterministically rounds the fractional solution in order to compute the outcome; this result improves upon the previously best known algorithm that has an approximation ratio of 3. We are furthermore interested in approximation algorithms that are resistant to manipulation by (coalitions of) voters, i.e., algorithms that do not motivate voters to misreport their true preferences in order to improve their distance from the outcome. We complement previous results in the literature with new upper and lower bounds on strategyproof and group-strategyproof algorithms.

1 Introduction

Approval voting is a very popular voting protocol mainly used for committee elections [2]. In such a protocol, the voters are allowed to vote for, or approve of, as many candidates as they like. In the last three decades, many scientific societies and organizations have adopted approval voting for their council elections. The solution concept that has been used in almost all such elections in practice is the minisum solution, *i.e.*, output the committee which, when seen as a 0/1-vector, minimizes the sum of the Hamming distances to the ballots. We assume throughout the paper that the committee should be of some predefined size k. Then the minisum solution consists of the k candidates with the highest number of approvals.

This solution however may ignore some voters' preferences in certain instances and does not take fairness issues into account. We demonstrate this with the following example with four voters, five candidates, and $k = 2$. Each row represents the preference of the corresponding voter. The minisum solution contains the candidates $\{a, b\}$. The distances of the voters from this outcome are 1, 0, 2, and 5 for voters 1, 2, 3, and 4, respectively (counting the number of alternatives in which the voter disagrees with the outcome). Instead, the solution $\{a, c\}$ has distances 3, 2, 2, and 3, respectively, and suggests a better compromise among the voters since everybody is relatively close to the outcome.

[1]A slightly different version of the paper appeared in *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 10).* This work was partially supported by the European Union under the Euro-NF Network of Excellence and Cost Action IC0602 "Algorithmic Decision Theory", and by a "Caratheodory" basic research grant from the University of Patras.

	a	b	c	d	e
1	1	1	0	0	1
2	1	1	0	0	0
3	1	1	1	1	0
4	0	0	1	1	1

Recently, a new voting rule, the *minimax solution*, was introduced as a means to achieve a compromise between the voters' preferences [3]. The minimax solution picks the k candidates for which the maximum (Hamming) distance of any voter from the outcome is minimized. Since this rule minimizes the disagreement with the least satisfied voter, it tends to result in outcomes that are more widely acceptable than the minisum solution. On the negative side the minimax solution has two main drawbacks that prevent its applicability: (i) the problem of computing the minimax solution is NP-hard, and (ii) voters may have incentives to misreport their preference in order to improve the distance of their true preference from the outcome. Our main goal in this paper is to tackle these issues by resorting to approximation algorithms.

Approximation algorithms tackle the computational hardness of an optimization problem by producing (in polynomial-time) solutions provably close to optimal ones for any problem instance; see [9] for a coverage of early work in the field. We refer to the optimization problem of computing the minimax solution as k-minimax approval. [4] present a 3-approximation algorithm for the problem; given an instance, the algorithm produces a solution (i.e., a set of k candidates) so that its distance from any voter's preference is at most 3 times the maximum distance of the voters from the minimax solution. The algorithm is very simple to describe and we will refer to it here as the k-completion algorithm: it arbitrarily picks a voter and computes a set of k candidates which has minimum distance from this voter. An immediate question is whether algorithms with better approximation ratios exist. Another interesting question is whether we can have good approximations by non-dictatorial algorithms. Note that the k-completion algorithm is dictatorial as it is based only on one voter's preferences.

The issue of resistance to manipulation is the very subject of Mechanism Design; see [6] for an introduction to the field. In our context, it translates to algorithms for k-minimax approval which, given a profile, compute an approximate solution in such a way that no single voter or a coalition of voters have any incentive to misreport their preferences in order to decrease their distance from the outcome. The corresponding properties of resistance to manipulation by single voters and coalitions of voters are known as strategyproofness and group-stratefyproofness, respectively. [4] prove that the minimax solution is not resistant to manipulation while the k-completion algorithm is. They also pose the question of computing the best possible bound on the approximation ratio of algorithms that are resistant to manipulation. This question falls within the line of research on mechanisms without monetary transfers [8] and, in particular, approximate mechanism design without money [7].

We make progress in both directions. Concerning the approximability of k-minimax approval by polynomial-time algorithms, we first establish a connection between the property of Pareto-efficiency and approximability. As a corollary, we obtain that Minisum (i.e., the algorithm that returns a minisum solution) has approximation ratio at most $3 - \frac{2}{k+1}$ for k-minimax approval. Our strongest result in this direction is an algorithm based on linear programming that achieves an improved approximation ratio of 2; this is a significant improvement compared to the previously best known algorithms. The algorithm is based on rounding the fractional solution of a natural linear programming relaxation for k-minimax approval. This result is the best possible that can be obtained using the particular LP relaxation which has an integrality gap of 2.

In the direction of algorithms resistant to manipulation, we observe that a variation of Minisum is strategyproof and present a Pareto-efficient refinement of the k-completion algorithm. Due to Pareto-efficiency, the latter algorithm has approximation ratio $3 - \frac{2}{k+1}$ as well. We also present the first inapproximability results for algorithms that are resistant to manipulation, making progress on the question posed in [4]. In particular, we present a lower bound of $2 - \frac{2}{k+1}$ on the approximation ratio of any strategyproof algorithm and a negative result which states that a slightly stronger notion of group-strategyproofness cannot be achieved by algorithms with approximation ratio different than $3-\frac{2}{k+1}$ and infinity. Our lower bounds are not based on any computational complexity assumption and, hence, hold for exponential-time algorithms as well.

2 Notation and Definitions

We fix some notation used in the following. We typically use n to denote the number of voters and m for the number of candidates. We denote the set of candidates by A. A preference is simply a subset of A. A profile P is a tuple $P = (P_1, ..., P_n)$ where P_i denotes the preference of voter i (i.e., the set of candidates she approves). Throughout the paper we make the reasonable assumption that $n > k$. When this is not explicitly mentioned (e.g., in some lower bound proofs), we can complete the profile by adding indifferent voters (that approve no candidate).

We extend the notion of (Hamming) distance to subsets of A as follows. We say that the distance of two sets Q and T is the total number of candidates in which they differ, i.e.,

$$d(Q,T) = |Q \setminus T| + |T \setminus Q| = |Q| + |T| - 2|Q \cap T|.$$

Note that this is precisely the Hamming distance of the sets, when seen as binary vectors where the ith coordinate of each vector equals 1 if the ith candidate belongs to the set.

3 Approximation Algorithms

We begin by establishing a connection between Pareto-efficiency and low approximation ratio.

Definition 1. Given a profile P, a size-k set $K \subseteq A$ is called Pareto-efficient with respect to P if there is no other size-k set $K' \subseteq A$ such that $d(K', P_{i^*}) < d(K, P_{i^*})$ for some voter i^* and $d(K', P_i) \leq d(K, P_i)$ for any other voter i. An algorithm for k-minimax approval is Pareto-efficient if, on any input profile P, its outcome is Pareto-efficient with respect to P.

The next lemma significantly extends the class of 3-approximation algorithms for minimax-approval and will be proved very useful later. Interestingly, Minisum is Pareto-efficient; the proof follows by the definition of Pareto-efficiency and the fact that Minisum minimizes the sum of the distances of the outcome from the voters.

Lemma 2. *Any Pareto-efficient algorithm for k-minimax approval has approximation ratio at most* $3 - \frac{2}{k+1}$.

Proof. Let P be a profile and let O and K be the minimax solution and the outcome returned by a non-optimal Pareto-efficient algorithm on input P. Let $OPT = \max_i\{d(O, P_i)\}$. We will show that $d(K, P_i)/OPT \leq 3 - \frac{2}{k+1}$ for every voter i.

First assume that $OPT \geq k + 1$. Then, by applying the triangle inequality, we obtain

$$\begin{aligned}\frac{d(K,P_i)}{OPT} &\leq \frac{d(K,O) + d(O,P_i)}{OPT} \\ &\leq 1 + 2k/OPT \leq 3 - \frac{2}{k+1}\end{aligned}$$

for each voter i. The second inequality follows since the distance of any two size-k sets is at most $2k$ and $d(O, P_i) \leq OPT$.

Now, assume that $OPT < k + 1$. Since the solution returned by the algorithm is non-optimal for the particular profile P, there exists a voter i^* such that $d(K, P_{i^*}) < d(O, P_{i^*})$. Indeed, if this was not the case, then K would not be Pareto-efficient with respect to P. By the definition of the distance, we observe that $d(K, P_{i^*})$ has the same parity with $d(O, P_{i^*})$, and the above argument implies that

$$d(K, P_{i^*}) \leq d(O, P_{i^*}) - 2.$$

Now, using this observation and by applying the triangle inequality twice, we have

$$\begin{aligned}
\frac{d(K, P_i)}{OPT} &\leq \frac{d(K, P_{i^*}) + d(P_{i^*}, P_i)}{OPT} \\
&\leq \frac{d(K, P_{i^*}) + d(O, P_{i^*}) + d(O, P_i)}{OPT} \\
&\leq \frac{2d(O, P_{i^*}) + d(O, P_i) - 2}{OPT} \\
&\leq 3 - \frac{2}{OPT} \leq 3 - \frac{2}{k+1}
\end{aligned}$$

for any voter i. This completes the proof. ☐

We now present an algorithm based on linear programming. On an input profile P, the algorithm uses the following equivalent integer linear program for k-minimax approval.

$$\begin{aligned}
\text{minimize} \quad & q \\
\text{subject to:} \quad & \forall i \in N, q + 2 \sum_{a \in P_i} x_a \geq k + |P_i| \\
& \sum_{a \in A} x_a = k \\
& \forall a \in A, x_a \in \{0, 1\} \\
& q \geq 0
\end{aligned}$$

The variable x_a denotes whether candidate a is included in the solution ($x_a = 1$) or not ($x_a = 0$). The first constraint essentially lower-bounds the value of variable q by the maximum distance of a voter from the size-k set that consists of the candidates included in the solution. The LP-based algorithm solves the LP relaxation in which the integrality constraint has been relaxed to $0 \leq x_a \leq 1$. In this way, a fractional solution is obtained with the x-variables having values in $[0, 1]$. Then, the algorithm includes the candidates with the k largest x-variables in the final solution (by breaking ties arbitrarily).

Theorem 3. *The LP-based algorithm has approximation ratio at most* 2.

Proof. Consider the application of the LP-based algorithm on a profile P. Denote by (q^*, x^*) the optimal fractional solution of the LP and let K be the outcome of the LP-based algorithm. We will show that, for each voter i, her preference P_i has distance at most $2q^*$ from the set K. Since q^* is a lower bound on the cost of the optimal integral solution for the particular instance of k-minimax approval, we will have obtained the desired 2-approximation bound.

Denote by Y_i the set of candidates in the preference of voter i that belong to the set K, i.e., $Y_i = P_i \cap K$. Let j be a voter whose preference P_j has maximum distance from K. The first constraint of the LP implies that

$$q^* \quad \geq \quad k + |P_j| - 2 \sum_{a \in P_j} x_a$$

and, using the fact that the x-variables of the LP are upper-bounded by 1 (due to the third LP constraint), we obtain that $q^* \geq |k - |P_j||$. Observe that if $|Y_j| = \min\{k, |P_j|\}$, then $d(K, P_j) = |k - |P_j||$, i.e., the solution of the algorithm is optimal in this case. So, in the following, we assume that $|Y_j| < \min\{k, |P_j|\}$.

For the sake of contradiction, assume that $d(K, P_j) > 2q^*$. By the definition of distance and the first LP constraint, we obtain

$$k + |P_j| - 2|Y_j| \quad > \quad 2q^* \geq 2\left(k + |P_j| - 2 \sum_{a \in P_j} x_a^*\right)$$

and, equivalently,

$$0 \quad > \quad k + |P_j| + 2|Y_j| - 4 \sum_{a \in P_j} x_a^*. \tag{1}$$

Since none of the candidates in $P_j \setminus Y_j$ was selected in the solution, this means that the x-variables corresponding to the $k - |Y_j|$ candidates in $K \setminus Y_j$ are not smaller than any x-variable corresponding to a candidate in $P_j \setminus Y_j$, i.e., for each candidate a in $K \setminus Y_j$, it holds that $x_a^* \geq \max_{a' \in P_j \setminus Y_j}\{x_{a'}^*\}$. By summing over all candidates in $K \setminus Y_j$, we have

$$\begin{aligned} \sum_{a \in K \setminus Y_j} x_a^* \quad &\geq \quad (k - |Y_j|) \max_{a' \in P_j \setminus Y_j}\{x_{a'}\} \\ &\geq \quad \frac{k - |Y_j|}{|P_j| - |Y_j|} \sum_{a' \in P_j \setminus Y_j} x_{a'}^*. \end{aligned} \tag{2}$$

By the definition of set Y_j, we have that every candidate of $K \setminus Y_j$ also belongs to $A \setminus P_j$. Hence

$$\sum_{a \in A \setminus P_j} x_a^* \quad \geq \quad \sum_{a \in K \setminus Y_j} x_a^*. \tag{3}$$

Furthermore, using the third LP constraint, we have

$$\sum_{a \in P_j \setminus Y_j} x_a^* = \sum_{a \in P_j} x_a^* - \sum_{a \in Y_j} x_a^* \geq \sum_{a \in P_j} x_a^* - |Y_j|. \tag{4}$$

Putting (2), (3), and (4) together, we have

$$\sum_{a \in A \setminus P_j} x_a^* \quad \geq \quad \frac{k - |Y_j|}{|P_j| - |Y_j|} \sum_{a \in P_j} x_a^* - \frac{|Y_j|(k - |Y_j|)}{|P_j| - |Y_j|}.$$

Now, observe that the left hand side in the above inequality satisfies (due to the second LP constraint)

$$\sum_{a \in A \setminus P_j} x_a^* \quad = \quad \sum_{a \in A} x_a^* - \sum_{a \in P_j} x_a^* = k - \sum_{a \in P_j} x_a^*.$$

Hence, the above inequality yields

$$k - \sum_{a \in P_j} x_a^* \geq \frac{k - |Y_j|}{|P_j| - |Y_j|} \sum_{a \in P_j} x_a^* - \frac{|Y_j|(k - |Y_j|)}{|P_j| - |Y_j|},$$

and, equivalently,

$$\sum_{a \in P_j} x_a^* \leq \frac{k|P_j| - |Y_j|^2}{k + |P_j| - 2|Y_j|}. \tag{5}$$

Now, (1) and (5) yield to the following contradiction:

$$\begin{aligned} 0 &> k + |P_j| + 2|Y_j| - 4\frac{k|P_j| - |Y_j|^2}{k + |P_j| - 2|Y_j|} \\ &= \frac{(k - |P_j|)^2}{k + |P_j| - 2|Y_j|} \geq 0, \end{aligned}$$

We conclude that $d(K, P_j) \leq 2q^*$ as desired. □

Given that the rounding in the LP-based algorithm is performed in an extremely simple way, one might hope that a more clever rounding could yield an improved algorithm. Unfortunately, the particular LP relaxation has an integrality gap of 2 and well-known arguments from the theory of approximation algorithms [9] imply that this is the best possible bound that can be obtained using the particular LP relaxation. Consider a profile with at least $2k$ candidates and denote by A' a size-$2k$ set of candidates. There are sufficiently many voters so that each one approves a different set of k candidates from A'. Clearly, for any k-size subset Q of A', there exists a voter whose preference does not include any of the candidates in Q. Hence, the minimax solution on the particular instance has cost at least $2k$. The claim follows by observing that the solution with the x-variables set to $1/2$ and $q = k$ satisfies the constraints of the LP relaxation.

4 Resistance to Manipulation

Let us first formally define strategyproofness in our setting. Given a profile P and an algorithm R, we denote by $R(P)$ the outcome of the algorithm on profile P. We also denote by P_{-i} the preferences of all voters besides i. Hence, we can also write P as (P_{-i}, P_i). Strategyproofness means that no voter i has an incentive to unilaterally change her preference so as to reduce the distance of P_i from the outcome of the algorithm.

Definition 4. An algorithm R is strategyproof (SP) if for any voter i, for any profile P, and for any $P'_i \subseteq A$:

$$d(P_i, R(P_{-i}, P_i)) \leq d(P_i, R(P_{-i}, P'_i)).$$

We begin with an example demonstrating that the minimax solution is not SP. Consider the profile at the left table below with $k = 2$; a similar example is presented in [4]. In this

	a	b	c
1	1	1	0
2	0	1	1
3	0	1	0

	a	b	c
1	1	1	0
2	0	0	1
3	0	1	0

profile, the sets $\{a, b\}$ and $\{b, c\}$ are those with distance at most 2 from all voters. Assume that $\{a, b\}$ is the minimax solution returned for the particular profile (the other case is symmetric). Now, assume that voter 2 has $\{c\}$ as her preference (see the right table). Now, the only set that has distance at most 2 from each voter's preference is $\{b, c\}$, i.e., exactly the preference of voter 2 in the first profile. This implies that voter 2 has an incentive to misreport her preference as $\{c\}$ instead of $\{b, c\}$ and demonstrates that minimax is not SP. The same example can show that the LP-based algorithm is not SP either.

Note that both solutions mentioned above are minisum solutions as well. This implies that Minisum is not SP in general. However, we can introduce a simple tie-breaking rule which assigns distinct ids to the candidates and ties for the last positions of the outcome are resolved by selecting the candidates with the smallest id. Then, Minisum equipped with the smallest-id-first tie-breaking rule can be easily proved to be strategyproof. Note that the particular assumption on the way ties are broken does not affect the Pareto-efficiency of Minisum. We summarize the discussion on Minisum to the following statement. Compared to the k-completion algorithm, Minisum is certainly non-dictatorial.

Theorem 5. *Minisum with the smallest-id tie-breaking rule is SP and has approximation ratio at most* $3 - \frac{2}{k+1}$ *for k-minimax approval.*

Let us remark here that the fact that a variation of Minisum is SP indicates that k-minimax approval is sufficiently restricted as a setting since well-known impossibility results state that, in general, strategyproofness is only achievable by dictatorial algorithms; see [6].

In the following, we present a lower bound on the approximation ratio of SP algorithms. We outline the main argument with the following example with $k = 1$ (we essentially adapt to our model an argument used in [7] in a slightly different context).

Consider the application of an SP algorithm on the following profile with $k = 1$. Without

	a_1	a_2	a_3	a_4	a_5	a_6
1	1	1	1	0	0	0
2	0	0	0	1	1	1

loss of generality, let $\{a_1\}$ be the outcome of the algorithm for this profile (the other cases can be handled symmetrically). Now consider the profile below. Again, the outcome should

	a_1	a_2	a_3	a_4	a_5	a_6
1	1	0	0	0	0	0
2	0	0	0	1	1	1

be the same otherwise voter 1 would have an incentive to misreport her preference from $\{a_1\}$ to $\{a_1, a_2, a_3\}$ and improve her distance from the outcome returned by the algorithm; this would violate strategyproofness. The maximum distance in the second profile is 4. The minimax solution approves one of the three rightmost candidates and has maximum distance 2. Hence, the approximation ratio is 2 in this case.

The extension of this argument for higher values of k yields a slightly weaker lower bound.

Theorem 6. *Any SP algorithm for k-minimax approval has approximation ratio at least* $2 - \frac{2}{k+1}$.

Proof. Consider a profile with $m \geq 4k$ candidates and two voters 1 and 2 that approve the disjoint size-$2k$ sets P_1 and P_2, respectively. Let K be the outcome of an SP algorithm on this particular profile. Assume that $P_1 \cap K \leq k/2$ (the other case is handled similarly). Now, consider the profile in which voter 1 approves the set P_1 and voter 2 approves the set K. We argue that the outcome of the algorithm is again K. Indeed, if this was not the case and the outcome was a set $K' \neq K$, voter 2 would have an incentive to misreport her preference as P_2 instead of K in order to decrease the distance of her true preference from the outcome. The distance of voters 1 and 2 from the outcome in the second profile is $d(K, P_1) = 3k - 2|K \cap P_1|$ and 0, respectively.

Let t be an integer such that

$$\frac{3k - 2|K \cap P_1| - 2}{4} \leq t \leq \frac{3k - 2|K \cap P_1| + 2}{4}.$$

Since $|K \cap P_1| \leq k/2$ and $|P_1| = 2k$, it holds that $t \leq |P_1 \setminus K|$. Consider the size-k set O which consists of the alternatives in $K \cap P_1$, t alternatives from $P_1 \setminus K$, and $k - |K \cap P_1| - t$ alternatives from $K \setminus P_1$. We have

$$d(O, K) = 2t \leq \frac{3k - 2|K \cap P_1| + 2}{2}$$

and

$$d(O, P_1) = 3k - 2|K \cap P_1| - 2t \leq \frac{3k - 2|K \cap P_1| + 2}{2}.$$

Hence, the approximation ratio of the algorithm for the second profile is at least

$$\begin{aligned} \frac{3k - 2|K \cap P_1|}{\max\{d(O,K), d(O,P_1)\}} &\geq 2 - \frac{4}{3k - 2|K \cap P_1| + 2} \\ &\geq 2 - \frac{2}{k+1}. \end{aligned}$$

The last inequality follows since $|K \cap P_1| \leq k/2$. □

We now move to stronger notions of resistance to manipulation. For a set (or coalition) of voters S, we denote by P_{-S} the preferences of the voters not in S.

Definition 7. An algorithm R is group-strategyproof (GSP) if for any coalition S of voters, and for any profile P, there is no profile P'_S of the voters in S such that:

$$d(P_i, R(P_{-S}, P_S)) > d(P_i, R(P_{-S}, P'_S)) \; \forall i \in S.$$

It is not hard to see that Minisum is not GSP. In contrast, the k-completion algorithm can be easily implemented so that it is GSP. The reason for this is that a coalition that does not contain the dictator cannot affect the outcome and the dictator has no incentive to participate in any coalition since her distance from the outcome is anyway minimum. We present a refinement of the k-completion algorithm which can be proved to be simultaneously GSP and Pareto-efficient. Then, Lemma 2 implies that its approximation ratio is at most $3 - \frac{2}{k+1}$. The algorithm uses an ordering of the voters with the dictator being first and an ordering of the candidates. Now, we can think of a candidate a as a binary vector z_a such that the i-th coordinate of the vector is 1 if voter i approves candidate a and 0 otherwise. For each candidate a, it computes its score as

$$\mathrm{sc}(a) = \sum_{i=1}^{n} z_a(i) \cdot 2^{n-i}$$

and picks the k candidates with highest scores by breaking ties according to the candidate ordering.

The Pareto-efficiency and strategyproofness of this algorithm become apparent by the following interpretation of its execution. Initially, it considers all possible size-k sets as possible outcomes. Among them, it keeps the ones that have the same minimum distance from the preference of voter 1. Then, among them, it keeps the ones that have the same minimum distance from the preference of voter 2, and so on. After considering voter n, it returns as an outcome one among the sets kept at that point.

Our last result concerns a stronger definition of group-strategyproofness.

Definition 8. An algorithm R is strongly group-strategyproof (strongly GSP) if for any coalition S of voters, and for any profile P, there is no profile P'_S of the voters in S such that:

$$d(P_i, R(P_{-S}, P_S)) \geq d(P_i, R(P_{-S}, P'_S)) \; \forall i \in S$$

with strict inequality for at least one voter of S.

The rationale behind this concept is that we demand the algorithm to be resistant to coalitions in which some voters may change their preference profile in order to help other members of the coalition (without necessarily gaining something for themselves). We make a connection between Pareto-efficiency and strong group-strategyproofness. We show that the former property is necessary in order to guarantee the existence of good approximation algorithms satisfying the latter. Of course, it is not sufficient. For example, minisum is Pareto-efficient but not even GSP. We also point out that this property is not necessary for group-strategyproofness since there are implementations of the k-completion algorithm that are not Pareto-efficient.

Lemma 9. *Any strongly GSP algorithm for k-minimax approval that has finite approximation ratio is Pareto-efficient.*

Proof. Consider a strongly GSP algorithm with finite approximation ratio. First observe that in each profile in which all voters approve the same set S of k candidates, the algorithm must return S as the outcome. If this is not the case for some profile of this kind, then the approximation ratio would be infinite.

Assume now that the algorithm returns a size-k set K on some profile which is not Pareto-efficient. Then, there exists another size-k set K' such that $d(K', P_{i^*}) < d(K, P_{i^*})$ for some voter i^* and $d(K', P_i) \leq d(K, P_i)$ for any other voter i. Now, the voters have an incentive to misreport the set K' and improve their distance from the outcome. □

Lemmas 2 and 9 imply that if a strongly GSP algorithm with finite approximation ratio exists, then it must have approximation ratio at most $3-\frac{2}{k+1}$. We complement this corollary with the following tight lower bound.

Theorem 10. *Any strongly GSP algorithm for k-minimax approval has approximation ratio at least $3 - \frac{2}{k+1}$.*

Proof. Consider an algorithm with approximation ratio strictly better than $3 - \frac{2}{k+1}$. We will actually prove that it is manipulable by two voters. Consider the profile with $3k + 1$ candidates and $3k + 1$ voters in which the preference of voter i contains only candidate i. Denote by K the outcome of the algorithm for the particular profile. Let i^* be a voter that has a candidate not in K in her preference and consider the profile in which voter i^* approves the $2k+1$ candidates outside K. Now, since the algorithm has approximation ratio strictly better than $3-2/(k+1)$, the outcome for the new profile should include a candidate i' not in K. Hence, voters i^* and i' have an incentive to manipulate the algorithm; voter i^* misreports her preference and does not decrease her distance and voter i' strictly decrease her distance from the outcome. □

Together with the above discussion, Theorem 10 leads to the following interesting statement.

Corollary 11. *Strongly GSP algorithms for k-minimax approval have at most two possible values for their approximation ratio: it can be either exactly $3 - \frac{2}{k+1}$ or infinity.*

5 Discussion

As a conclusion, let us discuss an interesting (but not obvious at first glance) relation of k-minimax approval to facility location problems; see [8, 9] and the references therein. In facility location, we are given agents located at the nodes of a network and the objective is to locate a facility at a node so that the maximum distance of any agent to the facility is minimized. k-minimax approval can be thought of as a facility location problem on a hypercubic network. Recall that a hypercube of dimension m has 2^m nodes each associated with a distinct 0/1 vector. An edge connects two nodes if their vectors differ in exactly one coordinate. So, k-minimax approval on a profile with n voters and m candidates can be thought of as a facility location instance with n agents (corresponding to the voters) located at some nodes of a hypercube of dimension m (the vector of such a node corresponds to the preference of a voter) with the objective being to put a facility to a node with exactly k 1s in its vector (corresponding to a size-k set of candidates) so that the maximum distance of any agent from the facility is minimized.

Besides this relation, the restriction on the type of nodes where the facility can be placed differentiates significantly k-minimax approval from standard facility location so that the best known approximation algorithm (implicit in [5]) for facility location on the hypercube does not carry over to our model. Furthermore, from the resistance to manipulation viewpoint, an important property of the standard facility location setting is single-peakedness in the agents preferences in the sense that the location of the agent is her mostly preferred location for the facility. This property does not hold in our model as there may be several among the possible locations an agent may prefer the most. A consequence of this peculiarity is that strategyproofness does not imply group-strategyproofness in k-minimax approval, in contrast to what is the case for single-peaked agent preferences [1] in facility location settings. We have demonstrated this when we observed that (a variation of) Minisum is SP but not GSP.

Our work leaves several challenging questions open. Concerning the approximability of k-minimax approval there is no known lower bound on the approximation ratio of polynomial-time algorithms besides the NP-hardness of the problem. It is interesting either to find such a lower bound or obtain a polynomial-time approximation scheme (PTAS), i.e., an algorithm that can achieve an approximation guarantee $1 + \epsilon$ for any constant $\epsilon > 0$ at the expense of a (possibly exponential) dependence of its running time on $1/\epsilon$. Progress in either direction will significantly improve our understanding of k-minimax approval. Experimental results in [4] provide evidence that local-search algorithms might have very low approximation ratios. Interestingly, we have a lower bound (very close to 3) for a natural and broad class of local-search algorithms that includes the ones considered in that paper; details will appear in the final version of the paper. As far as resistance to manipulation is concerned, our work leaves an intriguing gap between the upper bound of $3 - \frac{2}{k+1}$ and the lower bound of $2 - \frac{2}{k+1}$ on the approximation ratio of SP or GSP algorithms for k-minimax approval when $k \geq 2$. Furthermore, detecting whether strongly GSP algorithms with finite approximation ratio exist or not is of interest; here, we have made several unsuccessful attempts in both directions.

References

[1] Barberà, S.; Berga, D; and Moreno, B. 2010. Individual versus group strategy-proofness: When do they coincide? *Journal of Economic Theory*, in press.

[2] Brams, S. J.; Fishburn, P. C. 2007. *Approval Voting*, 2nd edition, Springer.

[3] Brams, S. J.; Kilgour, D. M.; and Sanver, M. R. 2007. A minimax procedure for electing committees. *Public Choice*, 132(3-4): 401-420.

[4] LeGrand, R.; Markakis, E.; and Mehta, A. 2007. Some results on approximating the minimax solution in approval voting. *Proceedings of 6th AAMAS*, 1193-1195.

[5] Li, M.; Ma, B.; and Wang, S. 1999. Finding similar regions in many strings. *Proceedings of 31st ACM STOC*, 473-482.

[6] Nisan, N. 2007. Introduction to mechanism design (for computer scientists). *Algorithmic Game Theory*, Cambridge University Press, 209-241.

[7] Procaccia, A. D.; and Tennenholtz, M. 2009. Approximate mechanism design without money. *Proceedings of 10th ACM EC*, 177-186.

[8] Schummer J.; and Vohra, R. V. 2007. Mechanism design without money. *Algorithmic Game Theory*, Cambridge University Press, 209-241.

[9] Vazirani, V. V. 2001. *Approximation algorithms.* Springer.

Ioannis Caragiannis
Department of Computer Engineering
and Informatics & RACTI
University of Patras, Greece
Email: caragian@ceid.upatras.gr

Dimitris Kalaitzis
Department of Computer Engineering
and Informatics & RACTI
University of Patras, Greece
Email: kalaitzis@cti.gr

Evangelos Markakis
Department of Informatics
Athens University of Economics
& Business, Greece
Email: markakis@gmail.com

Group-Strategyproof Irresolute Social Choice Functions

Felix Brandt

Abstract

We axiomatically characterize the class of pairwise irresolute social choice functions that are group-strategyproof according to Kelly's preference extension. The class is narrow but contains a number of appealing Condorcet extensions such as the *minimal covering set* and the *bipartisan set*, thereby answering a question raised independently by Barberà (1977) and Kelly (1977). These functions furthermore encourage participation and thus do not suffer from the no-show paradox (under Kelly's extension).

1 Introduction

One of the central results in social choice theory is that every social choice function (SCF)—a function mapping individual preferences to a collective choice—is susceptible to strategic manipulation (Gibbard, 1973; Satterthwaite, 1975). However, the classic result by Gibbard and Satterthwaite only applies to *resolute*, i.e., single-valued, SCFs. The notion of a resolute SCF is rather restricted and artificial.[1] For example, consider a situation with two voters and two alternatives such that each voter prefers a different alternative. The problem is not that a resolute SCF has to pick a single alternative (which is a well-motivated practical requirement), but that it has to pick a single alternative *based on the individual preferences alone* (see, e.g., Kelly, 1977). As a consequence, resoluteness is at variance with such elementary notions as neutrality and anonymity.

In order to remedy this shortcoming, Gibbard (1977) strengthened his impossibility to social choice functions that yield probability distributions over the set of alternatives rather than single alternatives. While this impossibility result is sweeping, it makes relatively strong assumptions on the voters' preferences. In contrast to the traditional setup in social choice theory, which usually only involves ordinal preferences, Gibbard's result relies on the axioms of von Neumann and Morgenstern (1947) (or an equivalent set of axioms) in order to compare lotteries over alternatives.[2]

The gap between Gibbard and Satterthwaite's theorem for resolute social choice functions and Gibbard's theorem for probabilistic social choice functions has been filled by a number of impossibility results with varying underlying notions of how to compare sets of alternatives with each other (e.g., Barberà, 1977; Kelly, 1977; Gärdenfors, 1976; Duggan and Schwartz, 2000). In this paper, we will be concerned with the weakest (and therefore least controversial) preference extension from alternatives to sets due to Kelly (1977). According to this definition, a set of alternatives is preferred to another set of alternatives if all elements of the former are preferred to all elements of the latter. Barberà (1977) and Kelly (1977) have shown independently that, for more than two alternatives, all social choice functions that are rationalizable via a binary preference relation are manipulable. Kelly (1977) concludes his paper by contemplating that "one plausible interpretation of such a theorem is that, rather than demonstrating the impossibility of reasonable strategy-proof social choice

[1]For example, Gärdenfors (1976) refers to resolute SCFs as "unnatural" and Kelly (1977) calls them "unreasonable."

[2]Gibbard (1978) later strengthened his impossibility theorem by generalizing it to choice mechanisms that do not necessarily take preference relations as inputs.

functions, it is part of a critique of the regularity [rationalizability] conditions" and Barberà (1977) states that "whether a nonrationalizable collective choice rule exists which is not manipulable and always leads to nonempty choices for nonempty finite issues is an open question." Also referring to nonrationalizable choice functions, Kelly (1977) writes: "it is an open question how far nondictatorship can be strengthened in this sort of direction and still avoid impossibility results."

In this paper, we characterize a class of social choice functions that cannot be manipulated by groups of voters who misrepresent their strict preferences. As a corollary of this characterization, all monotonic social choice functions that satisfy the *strong superset property* are group-strategyproof. The strong superset property goes back to early work by Chernoff (1954) (see also Bordes, 1979; Aizerman and Aleskerov, 1995) and requires that choice sets are invariant under the removal of unchosen alternatives. It has recently been used to characterize so-called set-rationalizable choice functions (Brandt and Harrenstein, 2009). The class of social choice functions satisfying the strong superset property is narrow but contains appealing Condorcet extensions such as *weak closure maximality* (also known as the *top cycle*, *GETCHA*, or the *Smith set*), the *minimal covering set*, the *bipartisan set*, and their generalizations (see Bordes, 1976; Laslier, 1997; Dutta and Laslier, 1999; Laslier, 2000).[3] Strategyproofness (according to Kelly's preference extension) thus draws a sharp line within the space of social choice functions as many established social choice functions (such as plurality, Borda's rule, and all weak Condorcet extensions) are known to be manipulable (Taylor, 2005) (and also fail to satisfy the strong superset property (Brandt and Harrenstein, 2009)). We furthermore show that our characterization is complete for *pairwise* social choice functions, i.e., social choice functions whose outcome only depends on the comparisons between pairs of alternatives.

Kelly's conservative preference extension has previously been primarily invoked in impossibility theorems because it is independent of the voters' attitude towards risk and the mechanism that eventually picks a single alternative from the choice set. Its interpretation in positive results, such as in this paper, is more debatable. Gärdenfors (1979) has shown that Kelly's extension is appropriate in a probabilistic context when voters are unaware of the lottery that will be used to pick the winning alternative. (Whether they are able to attach utilities to alternatives or not does not matter.) Alternatively, one can think of an independent chairman or a black-box that picks alternatives from choice sets in a way that prohibits a meaningful prior distribution. Whether these assumptions can be reasonably justified or such a device can actually be built is open to discussion. In particular, the study of distributed protocols or computational selection devices that justify Kelly's extension appears to be promising.

Remarkably, the robustness of the minimal covering set and the bipartisan set with respect to strategic manipulation also extends to *agenda manipulation.* The strong superset property precisely states that a social choice function is resistant to adding and deleting losing alternatives (see also the discussion by Bordes, 1983). Moreover, both choice rules are composition-consistent, i.e., they are strongly resistant to the introduction of clones (Laffond et al., 1993b, 1996).[4] Scoring rules like plurality and Borda's rule are prone to both types of agenda manipulation (Laslier, 1996; Brandt and Harrenstein, 2009) as well as to strategic manipulation.

We conclude the paper by pointing out that voters can never benefit from abstaining strategyproof pairwise SCFs. This does not hold for *resolute* Condorcet extensions, which

[3]If we assume an odd number of voters with strict preferences, the *tournament equilibrium set* (Schwartz, 1990) and the *minimal extending set* (Brandt, 2009) are conjectured to satisfy the strong superset property. Whether this is indeed the case depends on a certain graph-theoretic conjecture (Laffond et al., 1993a; Brandt, 2009).

[4]In addition to these attractive properties, the minimal covering set and the bipartisan set can be computed efficiently using non-trivial algorithms (Brandt and Fischer, 2008).

is commonly known as the *no-show paradox* (Moulin, 1988).

2 Related Work

Apart from the mentioned theorems by Barberà (1977) and Kelly (1977), there are numerous impossibility results concerning strategyproofness based on other—stronger—types of preferences over sets (see, e.g., Gärdenfors, 1976; Duggan and Schwartz, 2000; Barberà et al., 2001; Ching and Zhou, 2002; Sato, 2008; Umezawa, 2009), many of which are surveyed by Taylor (2005) and Barberà (2010). To the best of our knowledge, Jimeno et al. (2009) provide the only extension of Moulin's theorem on abstention for resolute Condorcet extensions (Moulin, 1988) to irresolute SCFs. Interestingly, they use stronger assumptions on preferences over sets and therefore obtain a negative result whereas our result is positive.

Inspired by early work by Bartholdi, III et al. (1989), recent research in computer science investigated how to use computational hardness—namely NP-hardness—as a barrier against manipulation (see, e.g., Conitzer and Sandholm, 2003; Conitzer et al., 2007; Faliszewski et al., 2009). However, NP-hardness is a worst-case measure and it would be much preferred if manipulation is hard on average. Recent negative results on the hardness of typical cases have cast doubt on this strand of research (see, e.g., Conitzer and Sandholm, 2006; Friedgut et al., 2008; Walsh, 2009), but more work remains to be done to settle the question completely. The current state of affairs is surveyed by Faliszewski and Procaccia (2010). If computational protocols or devices can be used to justify Kelly's extension by making "unpredictable" random selections, this might be an interesting alternative application of computational techniques to obtain strategyproofness.

3 Preliminaries

In this section, we provide the terminology and notation required for our results. We will use the standard model of social choice functions with a variable agenda (see, e.g., Taylor, 2005).

3.1 Social Choice Functions

Let U be a universe of alternatives over which voters entertain preferences. The preferences of voter i are represented by a complete preference relation $R_i \subseteq U \times U$.[5] We have $a \; R_i \; b$ denote that voter i values alternative a at least as much as alternative b. In compliance with conventional notation, we write P_i for the strict part of R_i, i.e., $a \; P_i \; b$ if $a \; R_i \; b$ but not $b \; R_i \; a$. Similarly, I_i denotes i's indifference relation, i.e., $a \; I_i \; b$ if both $a \; R_i \; b$ and $b \; R_i \; a$. The set of all preference relations over the universal set of alternatives U will be denoted by $\mathcal{R}(U)$. The set of *preference profiles*, i.e., finite vectors of preference relations, will be denoted by $\mathcal{R}^*(U)$. The typical element of $\mathcal{R}^*(U)$ is $R = (R_1, \ldots, R_n)$ and the typical set of voters is $N = \{1, \ldots, n\}$.

Any subset of U from which alternatives are to be chosen is a *feasible set* (sometimes also called an *issue* or *agenda*). Throughout this paper we assume the set of feasible subsets of U to be given by $\mathcal{F}(U)$, the set of finite and non-empty subsets of U, and generally refer to finite non-empty subsets of U as feasible sets. Our central object of study are *social choice*

[5] Transitivity of individual preferences is not necessary for our results to hold. In fact, Theorem 2 is easier to prove for general—possibly intransitive—preferences. Theorem 3, on the other hand, would require a more cumbersome case analysis for transitive preferences.

functions, i.e., functions that map the individual preferences of the voters and a feasible set to a set of socially preferred alternatives.[6]

Definition 1. A *social choice function (SCF)* is a function $f : \mathcal{R}^*(U) \times \mathcal{F}(U) \to \mathcal{F}(U)$ such that $f(R, A) \subseteq A$ and $f(R, A) = f(R', A)$ for all feasible sets A and preference profiles R, R' such that $R|_A = R'|_A$.

A *Condorcet winner* is an alternative a that, when compared with every other alternative, is preferred by more voters, i.e., $|\{i \in N \mid a \; R_i \; b\}| > |\{i \in N \mid b \; R_i \; a\}|$ for all alternatives $b \neq a$. An SCF is called a *Condorcet extension* if it uniquely selects the Condorcet winner whenever one exists.

The following notational convention will turn out to be useful throughout the paper. For a given preference profile R, $R_{i:(a,b)}$ denotes the preference profile $(R_1, \ldots, R_{i-1}, R'_i, R_{i+1}, \ldots, R_n)$ where $R'_i = R_i \cup \{(a, b)\}$ if $b \; P_i \; a$ and $R'_i = R_i \setminus \{(b, a)\}$ otherwise. That is, $R_{i:(a,b)}$ is identical to R except that alternative a is (weakly) strengthened with respect to b within voter i's preference relation.

A standard property of SCFs that is often considered is monotonicity. An SCF is monotonic if a chosen alternative remains in the choice set when it is strengthened in individual preference relations while leaving everything else unchanged.

Definition 2. An SCF f is *monotonic* if for all feasible sets A, preference profiles R, voters i, and alternatives $a, b \in A$, $a \in f(R, A)$ implies $a \in f(R'_{i:(a,b)}, A)$.

The strong superset property requires that a choice set is invariant under the removal of unchosen alternatives (Chernoff, 1954; Bordes, 1979; Aizerman and Aleskerov, 1995).

Definition 3. An SCF f satisfies the *strong superset property* (SSP) if for all feasible sets A, B and preference profiles R such that $f(R, A) \subseteq B \subseteq A$, $f(R, A) = f(R, B)$.

An SCF satisfies set-independence if the choice set is invariant under modifications of the preference profile with respect to unchosen alternatives (Laslier (1997) used the natural analog of this definition in the context of tournament solutions).

Definition 4. An SCF f satisfies *set-independence* if for all feasible sets A, preference profiles R, voters i, and alternatives $a, b \in A \setminus f(R, A)$, $f(R, A) = f(R_{i:(a,b)}, A)$.

The following proof is adapted from Laslier (1997), who showed the equivalent statement for tournament solutions.

Proposition 1. *Monotonicity and SSP imply set-independence.*

Proof. We show that every monotonic SCF f that satisfies SSP also satisfies set-independence. Let A be a feasible set, R a preference profile, i a voter, and $a, b \in A \setminus f(R, A)$. Furthermore, let $R' = R_{i:(b,a)}$. In case $a \in f(R', A)$, monotonicity yields a contradiction because a is strengthened in R but $a \notin f(R, A)$. Therefore, $a \notin f(R', A)$. SSP implies that $f(R, A) = f(R, A \setminus \{a\})$ and $f(R', A) = f(R', A \setminus \{a\})$. Moreover, $f(R, A \setminus \{a\}) = f(R', A \setminus \{a\})$ since R and R' are completely identical on $A \setminus \{a\}$. Hence, $f(R, A) = f(R', A)$ and f satisfies set-independence. $\square$

[6]This definition incorporates an independence condition that Bordes (1976) refers to as *independence of irrelevant alternatives (IIA)* and that resembles Arrow's IIA condition for social welfare functions.

3.2 Strategyproofness

An SCF is manipulable if one or more voters can misrepresent their preferences in order to obtain a more preferred outcome. Whether one choice set is preferred to another depends on how the preferences over individual alternatives are to be extended to sets of alternatives. In the absence of information about the mechanism that eventually picks a single alternative from any choice set, preferences over choice sets are typically obtained by the conservative extension $\widehat{R}_i$ (Barberà, 1977; Kelly, 1977), where for any pair of feasible sets A and B and preference relation R_i,

$$A \ \widehat{R}_i \ B \text{ if and only if } a \ R_i \ b \text{ for all } a \in A \text{ and } b \in B.$$

Clearly, in all but the simplest cases, $\widehat{R}_i$ is incomplete, i.e., many pairs of feasible sets are incomparable. $\widehat{P}_i$ denotes the strict part of relation $\widehat{R}_i$, i.e., $A \ \widehat{P}_i \ B$ if and only if $A \ \widehat{R}_i \ B$ and $a \ P_i \ b$ for at least one pair of $a \in A$ and $b \in B$.

Definition 5. An SCF f is *manipulable by a group of voters* $G \subseteq N$ if there exists a feasible set A and preference profiles R, R' with $R_i = R'_i$ for all $i \notin G$ such that

$$f(R', A) \ \widehat{P}_i \ f(R, A) \text{ for all } i \in G.$$

An SCF is *strategyproof* if it is not manipulable by single voters. An SCF is *group-strategyproof* if it is not manipulable by any group of voters.

It will turn out that many SCFs that fail to be strategyproof can only be manipulated by breaking ties strategically, i.e., voters can obtain a more preferred outcome by only misrepresenting their *indifference* relation. In many settings, for instance when the choice infrastructure requires a strict ranking of the alternatives, this may be deemed acceptable. Please observe that letting voters misrepresent their indifference relation is a weaker requirement than simply assuming that voters have linear preferences, which is often made in other results on strategyproofness (see, e.g., Taylor, 2005). Accordingly, we obtain the following definition.

Definition 6. An SCF is *strongly manipulable by a group of voters* $G \subseteq N$ if there exists a feasible set A and preference profiles R, R' with $R_i = R'_i$ for all $i \notin G$ and $I_i \subseteq I'_i$ for all $i \in G$ such that

$$f(R', A) \ \widehat{P}_i \ f(R, A) \text{ for all } i \in G.$$

An SCF is *weakly group-strategyproof* if it is not strongly manipulable by any group of voters.

In other words, every strongly manipulable SCF admits a manipulation in which voters only misrepresent their strict preferences.[7]

4 Results

We will present three main results. First, we show that no Condorcet extension is group-strategyproof. The proof of this claim, however, relies on breaking ties strategically. We therefore study *weak* group-strategyproofness and obtain a much more positive characterization result. Finally, we show that the two conditions used in our characterization are necessary and sufficient in the case of pairwise SCFs.

[7]Besides characterizing a class of SCFs that does not admit a strong manipulation, Theorem 2 shows something stronger about this class: In every manipulation where voters misrepresent strict preferences as well as indifferences, modifying the strict preferences is not necessary. The same outcome can be obtained by only misrepresenting the indifference relation.

2	...	2	1	...	1	1
$a_2, \ldots, a_m$	...	$a_1, \ldots, a_{m-1}$	$a_3, \ldots, a_m$	...	$a_1, \ldots, a_{m-2}$	$a_2, \ldots, a_{m-1}$
			a_1	...	a_{m-1}	a_m
a_1	...	a_m	a_2	...	a_m	a_1

Table 1: Preference profile R for $3m$ voters where $A = \{a_1, \ldots, a_m\}$

4.1 Manipulation of Condorcet Extensions

We begin by showing that all Condorcet extensions are weakly manipulable, which strengthens previous results by Gärdenfors (1976) and Taylor (2005) who showed the same statement for a weaker notion of manipulability and weak Condorcet extensions, respectively.[8]

Theorem 1. *Every Condorcet extension is manipulable when there are more than two alternatives.*

Proof. Let $A = \{a_1, \ldots, a_m\}$ with $m \geq 3$ and consider the preference profile R given in Table 1. For every alternative a_i, there are two voters who prefer every alternative to a_i and who are indifferent between the other alternatives. Moreover, there is one voter for every alternative a_i who ranks a_{i+1} below a_i and prefers every other alternative to both of them. Again, the voter is completely indifferent between these other alternatives.

Since $f(R, A)$ yields a non-empty choice set, there has to be some $1 \leq i \leq m$ such that $a_i \in f(R, A)$. Let $j = ((i - 2) \mod m) + 1$. Now, let R' be identical to R, except that the preferences of voter $2i - 1$ (i.e., the first voter who ranks a_i last) changed such that $a_j \; P'_{2i-1} \; a_k$ for all $k \neq j$. Furthermore, let R'' be identical to R, except that the preferences of voters $2i-1$ and $2i$ (i.e., the first two voters who rank a_i last) changed such that $a_j \; P''_{2i} \; a_k$ for all $k \neq j$.

In the case that $a_i \notin f(R', A)$, voter $2i - 1$ can manipulate as follows. Suppose R is the true preference profile. Then, the least favorable alternative of voter $2i - 1$ is chosen (possibly among other alternatives). He can misstate his preferences as in R' such that a_i is not chosen. Since he is indifferent between all other alternatives, $f(R', A) \; \widehat{P}_{2i-1} \; f(R, A)$.

If $a_i \in f(R', A)$, voter $2i$ can manipulate similarly. Suppose R' is the true preference profile. Again, the least favorable alternative of voter $2i$ is chosen. By misstating his preferences as in R'', he can assure that one of his preferred alternatives, namely a_j, is selected exclusively. This is the case because a_j is the Condorcet winner in R''. Hence, $f(R'', A) \; \widehat{P}'_{2i} \; f(R', A)$. □

4.2 Weakly Group-Strategyproof SCFs

The previous statement showed that no Condorcet extension is group-strategyproof. For our characterization of *weakly* group-strategyproof SCFs, we require set-independence and a new property that we call *set-monotonicity*. Set-monotonicity requires that a choice set should be invariant under the strengthening of chosen alternatives with respect to unchosen ones.

[8] A weak Condorcet winner is an alternative that is preferred by *at least* as many voters than any other alternative in pairwise comparisons. In contrast to Condorcet winners, weak Condorcet winners need not be unique. An SCF is called a *weak Condorcet extension* if it chooses the set of weak Condorcet winners whenever this set is non-empty. A large number of reasonable Condorcet extensions (including the minimal covering set and the bipartisan set) are not weak Condorcet extensions. Taylor (2005) calls the definition of weak Condorcet extensions "really quite strong" and refers to Condorcet extensions as "much more reasonable."

Definition 7. An SCF f is *set-monotonic* if for all feasible sets A, preference profiles R, voters i, and alternatives $a \in f(R, A)$, $b \in A \setminus f(R, A)$, $f(R, A) = f(R_{i:(a,b)}, A)$.

The conjunction of set-independence and set-monotonicity is stronger than monotonicity.

Proposition 2. *Set-independence and set-monotonicity imply monotonicity.*

Proof. Let f be a set-monotonic SCF, A a feasible set, R a preference profile, i a voter, and $a, b \in A$ such that $a \in f(R, A)$. Furthermore, let $R' = R_{i:(a,b)}$. Clearly, in case $b \notin f(R, A)$, set-monotonicity implies that $f(R', A) = f(R, A)$ and thus $a \in f(R', A)$. If, on the other hand, $b \in f(R, A)$, assume for contradiction that $a \notin f(R', A)$. If $b \in f(R', A)$, b is strengthened with respect to outside alternative a when moving from R' to R, and set-monotonicity again implies that $f(R, A) = f(R', A)$. Otherwise, if $b \notin f(R', A)$, it follows from set-independence that $f(R, A) = f(R', A)$, a contradiction. □

Set-monotonicity can be connected to existing well-established properties via the following proposition, whose proof runs along the same lines as that of Proposition 1.

Proposition 3. *Monotonicity and SSP imply set-monotonicity.*

Proof. We show that every monotonic SCF f that satisfies SSP also satisfies set-monotonicity. Let A be a feasible set, R a preference profile, i a voter, $a \in f(R, A)$, and $b \in A \setminus f(R, A)$. Furthermore, let $R' = R_{i:(a,b)}$. In case $b \in f(R', A)$, monotonicity yields a contradiction because b is strengthened in R but $b \notin f(R, A)$. Therefore, $b \notin f(R', A)$. SSP implies that $f(R, A) = f(R, A \setminus \{b\})$ and $f(R', A) = f(R', A \setminus \{b\})$. Moreover, $f(R, A \setminus \{b\}) = f(R', A \setminus \{b\})$ because R and R' are completely identical on $A \setminus \{b\}$. As a consequence, $f(R, A) = f(R', A)$ and f satisfies set-monotonicity. □

We are now ready to state the main result of this section.

Theorem 2. *Every SCF that satisfies set-monotonicity and set-independence is weakly group-strategyproof.*

Proof. Let f be an SCF that satisfies set-monotonicity and set-independence and assume for contradiction that f is not weakly group-strategyproof. Then, there has to be a feasible set A, a group of voters $G \subseteq N$, and two preference profiles R and R' with $R_i = R'_i$ for all $i \notin G$ and $I_i \subseteq I'_i$ for all $i \in G$ such that $f(R', A) \; \widehat{P}_i \; f(R, A)$ for all $i \in G$. We choose R and R' such that the union of the symmetric differences of individual preferences $\bigcup_{i \in N} (R_i \setminus R'_i) \cup (R'_i \setminus R_i)$ is inclusion-minimal, i.e., we look at a "smallest" counterexample in the sense that R and R' coincide as much as possible. Let $f(R, A) = X$ and $f(R', A) = Y$. Now, consider a pair of alternatives $a, b \in A$ such that, for some $i \in G$, $a \; P_i \; b$ and $b \; R'_i \; a$, i.e., voter i misrepresents his strict preference relation by strengthening b. The following case analysis will show that no such a and b exist, which implies that R and R' and consequently X and Y are identical, a contradiction.

Case 1 ($a, b \notin X$): It follows from set-independence that $R_{i:(b,a)}$ and R' yield a smaller counterexample since $f(R_{i:(b,a)}, A) = f(R, A) = X$.

Case 2 ($a, b \notin Y$): It follows from set-independence that R and $R'_{i:(a,b)}$ yield a smaller counterexample since $f(R'_{i:(a,b)}, A) = f(R', A) = Y$.

Case 3 ($a \in X$ and $b \in Y$): $Y \; \widehat{P}_i \; X$ implies that $b \; R_i \; a$, a contradiction.

Case 4 ($a \notin X$ and $b \in X$): It follows from set-monotonicity that $f(R_{i:(b,a)}, A) = f(R, A) = X$. Consequently, $R_{i:(b,a)}$ and R' constitute a smaller counterexample.

Case 5 ($a \in Y$ and $b \notin Y$): It follows from set-monotonicity that $f(R'_{i:(a,b)}, A) = f(R', A) = Y$. Consequently, R and $R'_{i:(a,b)}$ constitute a smaller counterexample.

It is easily verified that this analysis covers all possible cases. Hence, R and R' have to be identical, which concludes the proof. $\square$

As mentioned above, when assuming that voters have strict preferences, weak strategyproofness can be replaced with strategyproofness in Theorem 2.

Theorem 2 and Propositions 1 and 3 entail the following useful corollary.

Corollary 1. *Every monotonic SCF that satisfies SSP is weakly group-strategyproof.*

As mentioned in the introduction, there are few—but nevertheless quite attractive—SCFs that satisfy monotonicity and SSP, namely the top cycle, the minimal covering set, and the bipartisan set.[9]

4.3 Weakly Group-Strategyproof Pairwise SCFs

In this section, we identify a natural and well-known class of SCFs for which the characterization given in the previous section is complete. A SCF f is said to be based on pairwise comparisons (or simply *pairwise*) if, for all preference profiles R, R' and feasible sets A, $f(R, A) = f(R', A)$ if and only if

$$|\{i \in N \mid a\ P_i\ b\}| - |\{i \in N \mid b\ P_i\ a\}| = |\{i \in N \mid a\ P'_i\ b\}| - |\{i \in N \mid b\ P'_i\ a\}| \text{ for all } a, b \in A.$$

In other words, the outcome of a pairwise SCF only depends on the comparisons between pairs of alternatives (see, e.g., Young, 1974; Zwicker, 1991). The class of pairwise SCFs is quite natural and contains a large number of well-known voting rules such as Kemeny's rule, Borda's rule, Maximin, ranked pairs, and all rules based on simple majority rule (e.g., the Slater set, the uncovered set, the Banks set, the minimal covering set, and the bipartisan set). We now show that set-monotonicity and set-independence are *necessary* for the strategyproofness of pairwise SCFs.

Theorem 3. *Every weakly strategyproof pairwise SCF satisfies set-monotonicity and set-independence.*

Proof. We need to show that every pairwise SCF that fails to satisfy set-monotonicity or set-independence is strongly manipulable. Suppose SCF f does not satisfy set-monotonicity or set-independence. In either case, there exists a feasible set A, a preference profile R, a voter i, and two alternatives $a, b \in A$ with $a\ R_i\ b$ and $a \notin f(R, A) = X$ such that $f(R', A) = Y \neq X$ where $R' = R_{i:(b,a)}$. Let R_{n+1}, R_{n+2}, and R'_{n+2} be preference relations with indifferences between all pairs of alternatives except

$$\begin{aligned}
&x\ P_{n+1}\ y \text{ for all } (x, y) \in (((X \setminus Y) \times Y) \cup (X \times (Y \setminus X))),\\
&y\ P_{n+2}\ x \text{ for all } (x, y) \in (((X \setminus Y) \times Y) \cup (X \times (Y \setminus X))) \setminus \{(b, a)\},\\
&a\ R_{n+2}\ b \text{ if and only if } a\ R_i\ b,\\
&b\ R_{n+2}\ a \text{ if and only if } b\ R_i\ a,\\
&y\ P'_{n+2}\ x \text{ for all } (x, y) \in (((X \setminus Y) \times Y) \cup (X \times (Y \setminus X))) \setminus \{(b, a)\},\\
&a\ R'_{n+2}\ b \text{ if and only if } a\ R'_i\ b, \text{ and}\\
&b\ R'_{n+2}\ a \text{ if and only if } b\ R'_i\ a.
\end{aligned}$$

[9] SSP and monotonicity do not completely characterize weak strategyproofness. SCFs that satisfy set-monotonicity and set-independence but fail to satisfy SSP can easily be constructed.

We now define two preference profiles with $n+2$ voters where voter i is indifferent between a and b and the crucial change in preference between a and b has been moved to voter $n+2$. Let

$$S = (R_1, \ldots, R_{i-1}, R_i \cup \{(b,a)\}, R_{i+1}, \ldots, R_n, R_{n+1}, R_{n+2}) \text{ and}$$
$$S' = (R_1, \ldots, R_{i-1}, R_i \cup \{(b,a)\}, R_{i+1}, \ldots, R_n, R_{n+1}, R'_{n+2}).$$

Observe that all preferences between alternatives other than a and b cancel out each other in the preference relations of voter $n+1$ and $n+2$. It thus follows from the definition of pairwise SCFs that $f(S,A) = f(R,A) = X$ and $f(S',A) = f(R',A) = Y$. If $X \cup Y \neq \{a,b\}$ or $a\ P_i\ b$, we have $Y\ \widehat{P}_{n+2}\ X$ and f can be manipulated by voter $n+2$ at preference profile S by misstating his strict preference $a\ P_{n+2}\ b$ as $a\ I'_{n+2}\ b$. If, on the other hand, $X \cup Y = \{a,b\}$ and $a\ I_i\ b$, we have $X\ \widehat{P}'_{n+2}\ Y$ and f can be manipulated by voter $n+2$ at preference profile S' (by misstating his strict preference $b\ P'_{n+2}\ a$ as $a\ I_{n+2}\ b$). Hence, f is strongly manipulable. $\square$

We can now completely characterize weak group-strategyproofness of pairwise SCFs using these two properties.

Corollary 2. *A pairwise SCF is weakly group-strategyproof if and only if it satisfies set-monotonicity and set-independence.*

This shows that many pairwise SCFs are not weakly group-strategyproof because they are known to fail set-independence (Laslier, 1997). Notable exceptions are the top cycle, the minimal covering set, and the bipartisan set mentioned above.

Brams and Fishburn (1983) introduced a particularly natural variant of strategic manipulation where voters obtain a more preferred outcome by abstaining the election. A SCF is said to satisfy *participation* if voters are never better off by abstaining. A common criticism of Condorcet extensions is that they do not satisfy participation and thus suffer from the so-called *no-show paradox* (Moulin, 1988). However, Moulin's proof strongly relies on resoluteness. Irresolute Condorcet extensions that satisfy participation do exist and, in the case of pairwise SCFs, there is a close connection between strategyproofness and participation as shown by the following simple observation.[10]

Proposition 4. *Every strategyproof pairwise SCF satisfies participation.*

Proof. Let f be a pairwise SCF that fails participation, i.e., there exists a feasible set A, a preference profile R, and a preference relation R_{n+1} such that $f(R,A)\ \widehat{P}_{n+1}\ f((R_1, \ldots, R_n, R_{n+1}), A)$. Let furthermore R'_{n+1} be a preference relation that expresses complete indifference over all alternatives. Since f is pairwise, $f((R_1, \ldots, R_n, R'_{n+1}), A) = f(R,A)$ and f can be manipulated at profile $(R_1, \ldots, R_n, R_{n+1})$ by voter $n+1$ because by changing his preferences to R'_{n+1} he obtains the more preferred outcome $f(R,A)$. $\square$

It follows that all SCFs satisfying set-monotonicity and set-independence, which includes the Condorcet extensions mentioned earlier, satisfy participation according to Kelly's preference extension.

Acknowledgements

I am grateful to Markus Brill and Paul Harrenstein for helpful discussions. This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/3-3 and BR 2312/7-1.

[10] Proposition 4 holds for any preference extension, not just Kelly's.

References

M. Aizerman and F. Aleskerov. *Theory of Choice*, volume 38 of *Studies in Mathematical and Managerial Economics*. North-Holland, 1995.

S. Barberà. Manipulation of social decision functions. *Journal of Economic Theory*, 15(2): 266–278, 1977.

S. Barberà. Strategy-proof social choice. Barcelona Economics Working Paper 420, Barcelona Graduate School of Economics, 2010.

S. Barberà, B. Dutta, and A. Sen. Strategy-proof social choice correspondences. *Journal of Economic Theory*, 101(2):374–394, 2001.

J. Bartholdi, III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

G. Bordes. Consistency, rationality and collective choice. *Review of Economic Studies*, 43 (3):451–457, 1976.

G. Bordes. Some more results on consistency, rationality and collective choice. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*, chapter 10, pages 175–197. 1979.

G. Bordes. On the possibility of reasonable consistent majoritarian choice: Some positive results. *Journal of Economic Theory*, 31:122–132, 1983.

S. Brams and P. Fishburn. Paradoxes of preferential voting. *Mathematical Magazine*, 56(5): 207–214, 1983.

F. Brandt. Minimal stable sets in tournaments. Technical report, http://arxiv.org/abs/0803.2138, 2009. Presented at the 9th International Meeting of the Society of Social Choice and Welfare.

F. Brandt and F. Fischer. Computing the minimal covering set. *Mathematical Social Sciences*, 56(2):254–268, 2008.

F. Brandt and P. Harrenstein. Set-rationalizable choice and self-stability. Technical report, http://arxiv.org/abs/0910.3580, 2009.

H. Chernoff. Rational selection of decision functions. *Econometrica*, 22:422–443, 1954.

S. Ching and L. Zhou. Multi-valued strategy-proof social choice rules. *Social Choice and Welfare*, 19:569–580, 2002.

V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 781–788, 2003.

V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 627–634. AAAI Press, 2006.

V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007.

J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard-Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

B. Dutta and J.-F. Laslier. Comparison functions and choice correspondences. *Social Choice and Welfare*, 16(4):513–532, 1999.

P. Faliszewski and A. D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 2010. Forthcoming.

P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer-Verlag, 2009.

E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *Proceedings of the 49th Symposium on Foundations of Computer Science (FOCS)*, pages 243–249. IEEE Computer Society Press, 2008.

P. Gärdenfors. Manipulation of social choice functions. *Journal of Economic Theory*, 13(2): 217–228, 1976.

P. Gärdenfors. On definitions of manipulation of social choice functions. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.

A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3): 665–681, 1977.

A. Gibbard. Straightforwardness of game forms with lotteries as outcomes. *Econometrica*, 46(3):595–614, 1978.

J. L. Jimeno, J. Pérez, and E. García. An extension of the Moulin No Show Paradox for voting correspondences. *Social Choice and Welfare*, 33(3):343–459, 2009.

J. Kelly. Strategy-proofness and social choice functions without single-valuedness. *Econometrica*, 45(2):439–446, 1977.

G. Laffond, J.-F. Laslier, and M. Le Breton. More on the tournament equilibrium set. *Mathématiques et sciences humaines*, 31(123):37–44, 1993a.

G. Laffond, J.-F. Laslier, and M. Le Breton. The bipartisan set of a tournament game. *Games and Economic Behavior*, 5:182–201, 1993b.

G. Laffond, J. Lainé, and J.-F. Laslier. Composition-consistent tournament solutions and social choice functions. *Social Choice and Welfare*, 13:75–93, 1996.

J.-F. Laslier. Rank-based choice correspondences. *Economics Letters*, 52:279–286, 1996.

J.-F. Laslier. *Tournament Solutions and Majority Voting*. Springer-Verlag, 1997.

J.-F. Laslier. Aggregation of preferences with a variable set of alternatives. *Social Choice and Welfare*, 17:269–282, 2000.

H. Moulin. Condorcet's principle implies the no show paradox. *Journal of Economic Theory*, 45:53–64, 1988.

S. Sato. On strategy-proof social choice correspondences. *Social Choice and Welfare*, 31: 331–343, 2008.

M. A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

T. Schwartz. Cyclic tournaments and cooperative majority voting: A solution. *Social Choice and Welfare*, 7:19–29, 1990.

A. D. Taylor. *Social Choice and the Mathematics of Manipulation.* Cambridge University Press, 2005.

M. Umezawa. Coalitionally strategy-proof social choice correspondences and the Pareto rule. *Social Choice and Welfare*, 33:151–158, 2009.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 2nd edition, 1947.

T. Walsh. Where are the really hard manipulation problems? The phase transition in manipulating the veto rule. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 324–329, 2009.

H. P. Young. An axiomatization of Borda's rule. *Journal of Economic Theory*, 9:43–52, 1974.

W. S. Zwicker. The voter's paradox, spin, and the Borda count. *Mathematical Social Sciences*, 22:187–227, 1991.

Felix Brandt
Department of Informatics
Technische Universität München
85748 Garching bei München, Germany
brandtf@in.tum.de

Sum of Us: Strategyproof Selection from the Selectors

Noga Alon, Felix Fischer, Ariel D. Procaccia, and Moshe Tennenholtz

Abstract

We consider directed graphs over a set of n agents, where an edge (i, j) is taken to mean that agent i supports or trusts agent j. Given such a graph and an integer $k \leq n$, we wish to select a subset of k agents that maximizes the sum of indegrees, i.e., a subset of k most popular or most trusted agents. At the same time we assume that each individual agent is only interested in being selected, and may misreport its outgoing edges to this end. This problem formulation captures realistic scenarios where agents choose among themselves, which can be found in the context of Internet search, social networks like Twitter, or reputation systems like Epinions.

Our goal is to design mechanisms without payments that map each graph to a k-subset of agents to be selected and satisfy the following two constraints: *strategyproofness*, i.e., agents cannot benefit from misreporting their outgoing edges, and *approximate optimality*, i.e., the sum of indegrees of the selected subset of agents is always close to optimal. Our first main result is a surprising impossibility: for $k \in \{1, \dots, n-1\}$, no deterministic strategyproof mechanism can provide a finite approximation ratio. Our second main result is a randomized strategyproof mechanism with an approximation ratio that is bounded from above by four for any value of k, and approaches one as k grows.

1 Introduction

One of the most well-studied settings in social choice theory concerns a set of *agents* (also known as *voters* or *individuals*) and a set of *alternatives* (also known as *candidates*). The agents express their preferences over the alternatives, and these are mapped by some function to a winning alternative or set of winning alternatives. In one prominent variation, each agent must select a subset of alternatives it approves; this setting is known as *approval voting* [5].

We consider the special case of approval voting when the set of agents and the set of alternatives coincide; this for example occurs when the members of an organization use approval voting to elect a president or a committee from among their numbers.[1] We model this situation by a directed graph on the set of agents. An edge from agent i to agent j means that agent i approves, votes for, trusts, or supports agent j. Our goal is to select a subset of k "best" agents for a given graph; we will elaborate on what we mean by "best" momentarily.

The fact that agents and alternatives coincide allows us to make additional assumptions about agents' preferences. Indeed, we will assume that each agent is only interested in whether it is among those selected, that is, it receives utility one if selected and zero otherwise. We will see, however, that our results in fact hold for any setting where agents give their own selection priority over that of their approved candidates. This assumption, which is very reasonable in practice, is discussed in more detail in Section 5.

[1] Approval voting is employed in this exact context for example by scientific organizations such as the American Mathematical Society (AMS), the Institute of Electrical and Electronics Engineers (IEEE), the Game Theory Society (GTS), and the International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

A (deterministic) *k-selection mechanism* is a function that maps a given graph on the set of agents to a k-subset of selected agents. We also consider *randomized* k-selection mechanisms, which randomly select a subset. The outgoing edges in the underlying graph G are private information of the respective agent. Fixing a mechanism f, the agents play the following game. Each of them reports to the mechanism a set of outgoing edges, which might differ from the true set. The reported edges induce a graph G', and the mechanism selects the subset $f(G')$. We say that a mechanism is *strategyproof (SP)* if an agent cannot benefit from misreporting its outgoing edges, that is, cannot increase its chances of being selected, even if it has complete information about the rest of the graph. We further say that a mechanism is *group strategyproof (GSP)* if even a coalition of agents cannot all gain from misreporting their outgoing edges.

What remains to be specified is what we mean by selecting the "best" agents. In this paper, we measure the quality of a set of agents by their total number of incoming edges, i.e., the sum of their indegrees. The goal of the mechanism designer is to optimize this target function. Note that this goal is in a sense orthogonal to the agent's interests, which may make the design of good SP mechanisms difficult.

In addition to traditional voting settings, this model also captures different problems in networked environments. Consider for example an Internet search setting, where agents correspond to web sites and edges represent hyperlinks. Given this graph, a search engine must return a set of the, say, ten top web sites. Put another way, the top web sites are selected based on the votes cast by other web sites in the form of hyperlinks. Each specific web site, or more accurately its webmaster, is naturally concerned with appearing at the top of the search results, and to this end may add or remove hyperlinks at will.

A second motivating example can be found in the context of social networks. While some social networks, like Facebook (`http://facebook.com`), correspond to undirected graphs, there are many examples with unilateral connections. Each user of the reputation system Epinions (`http://epinions.com`) has a "Web of Trust", that is, the user unilaterally chooses which other users to trust. Another prominent example is the social network Twitter (`http://twitter.com`), which of late has become wildly popular; a Twitter user may choose which other users to "follow." In "directed" social networks, choosing a k-subset with maximum overall indegree simply means selecting the k most popular or most trusted users. Applications include setting up a committee, recommending a trusted group of vendors, targeting a group for an advertising campaign, or simply holding a popularity contest. The last point may seem pure fantasy, but, indeed, celebrity users of Twitter have recently held a race to the milestone of one million followers; the dubious honor ultimately went to actor Ashton Kutcher. Clearly Mr. Kutcher could increase the chance of being selected by not following any other users, that is, reporting an empty set of outgoing edges.

Since a mechanism that selects an optimal subset (in terms of total indegree) is clearly not SP, we will resort to approximate optimality. More precisely, we seek SP mechanisms that give a good approximation, in the usual sense, to the total indegree. Crucially, approximation is *not* employed in this context to circumvent computational complexity (as the problem of selecting an optimal subset is obviously tractable), but in order to sufficiently broaden the space of acceptable mechanisms to include SP ones.

Context and related work. The work in this paper falls squarely into the realm of *approximate mechanism design without money*, an agenda recently introduced by some of us (Procaccia and Tennenholtz [24]), building on earlier work (for example by Dekel et al. [9]). This agenda advocates the design of SP approximation mechanisms *without payments* for structured, and preferably computationally tractable, optimization problems. Indeed, while almost all the work in the field of *algorithmic mechanism design* [23] considers mechanisms that are allowed to transfer payments to and from the agents, money is usually unavailable

		Deterministic	Randomized
SP	**Upper bound**	n/a	$\min\{4, 1+\mathcal{O}(1/k^{1/3})\}$
	Lower bound	∞	$1+\Omega(1/k^2)$
GSP	**Upper bound**	n/a	$\frac{n}{k}$
	Lower bound	∞	$\frac{n-1}{k}$

Table 1: Summary of our results for k-selection mechanisms, where n is the number of agents. SP stands for strategyproof, GSP for group strategyproof.

in Internet domains like the ones discussed above (social networks, search engines) due to security and accountability issues (see, e.g., the book chapter by Schummer and Vohra [26]). Our notion of a mechanism, sometimes referred to as a *social choice rule* in the social choice literature, therefore precludes payments by definition. Note that Procaccia and Tennenholtz [24], and also subsequent papers [20, 21, 1], deal with a completely different domain, namely facility location.

LeGrand et al. [19] study approximations in the context of approval voting, mainly from a complexity perspective. They consider the (less standard) minmax solution that selects alternatives in a way that minimizes the maximum Hamming distance to the agents' ballots (as binary vectors). LeGrand et al. show that the optimization problem is NP-hard, and provide a trivial 3-approximation algorithm: simply choose the subset that is closest to the ballot of an arbitrary agent. Furthermore, they observe that this algorithm is also SP when an agent's (dis)utility is its Hamming distance to the selected subset.

For $k = 1$, that is, if one agent must be selected, the game we deal with is a special case of so-called *selection games* [3], where the possible strategies are the outgoing edges. More generally, this setting is related to work in distributed computing on *leader election* (see, e.g., [2, 8, 11, 4]). This line of work does not deal with self-interested agents. Instead, there is a certain number of malicious agents trying to manipulate the selection process, and the goal is to guarantee the selection of a non-malicious agent, at least with a certain probability.

Finally, this paper is related to work on manipulation of reputation systems, which are often modeled as weighted directed graphs; a reputation function maps a given graph to reputation values for the agents (see, e.g., [6, 14]). Although our positive results can be extended to weighted graphs, when the target function is the sum of weights on incoming edges, this would hardly be a reasonable target function. Indeed, in this context the absence of a specific incoming edge (indicating lack of knowledge) is preferable to an edge with low weight (which indicates distrust); see Section 5 for further discussion.

Results and techniques. We give rather tight upper and lower bounds on the approximation ratio achievable by k-selection mechanisms in the setting described above; the properties of the mechanisms fall along two orthogonal dimensions: deterministic vs. randomized, and SP vs. GSP. A summary of our results is given in Table 1.

Our contribution begins in Section 3 with a study of deterministic k-selection mechanisms. It is quite easy to see that no deterministic SP 1-selection mechanism can yield a finite approximation ratio. Intuitively, this should not be true for large values of k. Indeed, in order to have a finite approximation ratio, a mechanism should very simply select a subset of agents with at least one incoming edge, if there is such a set. In the extreme case when $k = n - 1$, we must select all the agents save one, and the question is whether there exists an SP mechanism that never eliminates the unique agent with positive indegree. Our first result gives a surprising negative answer to this question, and in fact holds for every value of k.

Theorem 3.1. *Let $N = \{1, \ldots, n\}$, $n \geq 2$, and $k \in \{1, \ldots, n-1\}$. Then there is no deterministic SP k-selection mechanism that gives a finite approximation ratio.*

The proof of the theorem is compact but rather tricky. It involves two main arguments. We first restrict our attention to a subset of the graphs, namely to stars with all edges directed at a specific agent. An SP mechanism over such graphs can be represented using a function over the boolean $(n-1)$-cube, which must satisfy certain constraints. We then use a parity argument to show that the constraints lead to a contradiction.

In Section 4 we turn to randomized k-selection mechanisms. We design a randomized mechanism, *Random m-Partition (m-RP)*, parameterized by m, that works by randomly partitioning the set of agents into m subsets, and then selecting the (roughly) k/m agents with largest indegree from each subset, when only the incoming edges from the other subsets are taken into account. This rather simple technique is reminiscent of work on *random sampling* in the context of auctions for digital goods [13, 17, 12] and combinatorial auctions [10], although our problem is fundamentally different. We have the following theorem.

Theorem 4.1. *Let $N = \{1, \ldots, n\}$, $k \in \{1, \ldots, n-1\}$. For every value of m, m-RP is SP. Furthermore,*

1. *2-RP has an approximation ratio of four, and*
2. *$\left(\lceil k^{1/3} \rceil\right)$-RP has an approximation ratio of $1 + \mathcal{O}(1/k^{1/3})$.*

For a given number k of agents to be selected, we can in fact choose the best value of m when applying m-RP. Thus, there exists a mechanism that always yields an approximation ratio of at most four, and furthermore provides a ratio that approaches one as k grows. In addition, we prove a lower bound of $1 + \Omega(1/k^2)$ on the approximation ratio that can be achieved by any randomized SP k-selection mechanism; in particular, the lower bound is two for $k = 1$.

As our final result, we obtain a lower bound of $(n-1)/k$ for randomized GSP k-selection mechanisms. This result implies that when asking for group strategyproofness one essentially cannot do better than simply selecting k agents at random, which is obviously GSP and gives an approximation ratio of n/k.

2 The Model

Let $N = \{1, \ldots, n\}$ be a set of *agents*. For each $k = 1, \ldots, n$, let $\mathcal{S}_k = \mathcal{S}_k(n)$ be the collection of k-subsets of N, i.e., $\mathcal{S}_k = \{S \subseteq N : \ |S| = k\}$. We consider directed graphs $G = (N, E)$, that is, graphs with N as the set of vertices, and write $\mathcal{G} = \mathcal{G}(N)$ for the set of such graphs.

A *deterministic k-selection mechanism* is a function $f : \mathcal{G} \to \mathcal{S}_k$ that selects a subset of agents for each graph. When the subset $S \subseteq N$ is selected, agent $i \in N$ obtains utility $u_i(S) = 1$ if $i \in S$ and $u_i(S) = 0$ otherwise, i.e., agents only care about whether they are selected or not. We further discuss this utility model in Section 5.

A *randomized k-selection mechanism* is a function $f : \mathcal{G} \to \Delta(\mathcal{S}_k)$, where $\Delta(\mathcal{S}_k)$ is the set of probability distributions over $\mathcal{S}_k$. Given a distribution $\mu \in \Delta(\mathcal{S}_k)$, the utility of agent $i \in N$ is

$$u_i(\mu) = \mathbb{E}_{S \sim \mu}[u_i(S)] = \Pr_{S \sim \mu}[i \in S].$$

Deterministic mechanisms can be seen as a special case of a randomized ones, always selecting a set of agents with probability one.

We say that a k-selection mechanism is *strategyproof (SP)* if an agent cannot benefit from misreporting its edges. Formally, strategyproofness requires that for every $i \in N$ and every pair of graphs $G, G' \in \mathcal{G}$ that differ only in the outgoing edges of agent i, it holds that $u_i(G) = u_i(G')$.[2] This means that the probability of agent $i \in N$ being selected has to be independent of the outgoing edges reported by i. A discussion of this definition in the context of randomized mechanisms can be found in Section 5.

A k-selection mechanism is *group strategyproof (GSP)* if there is no coalition of agents that can all gain from jointly misreporting their outgoing edges. Formally, group strategyproofness requires that for every $S \subseteq N$ and every pair of graphs $G, G' \in \mathcal{G}$ that differ only in the outgoing edges of the agents in S, there exists $i \in S$ such that $u_i(G) \leq u_i(G')$. An alternative, stronger definition requires that some agent *strictly* lose as a result of the deviation. Crucially, our result with respect to group strategyproofness is an impossibility, hence using the weaker definition only strengthens the result.

Given a graph G, let $\deg(i) = \deg(i, G)$ be the indegree of agent i in G, i.e., the number of its incoming edges. We seek mechanisms that are SP or GSP, and in addition approximate the optimization target $\sum_{i \in S} \deg(i)$, that is, we wish to maximize the sum of indegrees of the selected agents. Formally, we say that a k-selection mechanism f has an approximation ratio of α if for every graph G,

$$\frac{\max_{S \in \mathcal{S}_k} \sum_{i \in S} \deg(i)}{\mathbb{E}_{S \sim f(G)}[\sum_{i \in S} \deg(i)]} \leq \alpha.$$

3 Deterministic Mechanisms

In this section we study deterministic k-selection mechanisms. Before stating our impossibility result, we discuss some special cases.

Clearly, only one mechanism exists for $k = n$, that is, when all the agents must be selected, and this mechanism is optimal. More interestingly, it is easy to see that one cannot obtain a finite approximation ratio via a deterministic SP mechanism when $k = 1$. Indeed, let $n \geq 2$, let f be an SP deterministic mechanism, and consider a graph $G = (N, E)$ with $E = \{(1, 2), (2, 1)\}$, i.e., the only two edges are from agent 1 to agent 2 and vice versa. Without loss of generality we may assume that $f(G) = \{1\}$. Now, assume that agent 2 removes its outgoing edge; formally, we now consider the graph $G' = (N, E')$ with $E' = \{(1, 2)\}$. By strategyproofness, $f(G') = \{1\}$, but now agent 2 is the only agent with positive degree, hence the approximation ratio of f is infinite.

Note that in order to have a finite approximation ratio, our mechanism must satisfy the following property, which is also sufficient: if there is an edge in the graph, the mechanism must select a subset of agents with at least one incoming edge. The argument above shows that this property cannot be satisfied by any SP mechanism when $k = 1$, but intuitively it should be easy to satisfy when k is very large.

Consider, for example, the case where $k = n - 1$, that is, the mechanism must select all the agents save one. Can we design an SP mechanism with the extremely basic property that if there is only one agent with incoming edges, that agent would not be the only one *not* to be selected?

In the following theorem, we give a surprising negative answer to this question, even when we restrict our attention to graphs where each agent has at most one outgoing edge. Amusingly, a connection to the popular TV game show "Survivor" can be made. Consider a slight variation where each tribe member can vote for one other trusted member, but is also allowed not to cast a vote. One member must be eliminated at the tribal council, based on the votes. Since each member's first priority is not to be eliminated (i.e., to be selected), strategyproofness in our 0–1 utility model is in fact a necessary condition for strategyproofness in suitable, more refined utility models. The theorem then implies that

[2] By symmetry, this is equivalent to writing the last equality as an inequality.

a mechanism for choosing the eliminated member cannot be SP (even under 0–1 utilities) if it has the property that a member who is the only one that received votes cannot be eliminated. Put another way, lies are inherent in the game!

More generally, we show that for *any* value of k, strategyproofness and finite approximation ratio are mutually exclusive. A concise but nontrivial proof is given in the full version of this paper.

Theorem 3.1. *Let $N = \{1, \ldots, n\}$, $n \geq 2$, and $k \in \{1, \ldots, n-1\}$. Then there is no deterministic SP k-selection mechanism that gives a finite approximation ratio.*

It is interesting to note that if we change the problem formulation by allowing the selection of *at most* k agents for $k \geq 2$ then it is possible to design a curious deterministic SP mechanism with a finite approximation ratio that selects at most two agents. The reader is referred to Section 5 and to the full version of this paper for further discussion.

4 Randomized Mechanisms

In Section 3 we have established a total impossibility result with respect to deterministic SP k-selection mechanisms. In this section we ask to what extent this result can be circumvented using randomization.

4.1 SP Randomized Mechanisms

As we move to the randomized setting, it immediately becomes apparent that Theorem 3.1 no longer applies. Indeed, a randomized SP k-selection mechanism with a finite approximation ratio can be obtained by simply selecting k agents at random. However, this mechanism still yields a poor approximation ratio. Can we do better?

Consider first a simple deterministic mechanism that partitions the agents into two predetermined subsets S_1 and S_2. Next, the mechanism discards all edges between pairs of agents in the same subset. Finally, the mechanism chooses the top $k/2$ agents from each subset. In other words, the mechanism selects the $k/2$ agents with highest indegree from each subset, where the indegree is calculated only on the basis of incoming edges from the other subset. This mechanism is clearly SP. Indeed, consider some $i \in S_t$, $t \in \{1, 2\}$; its outgoing edges to agents inside its subset are disregarded, whereas its outgoing edges to agents in S_{3-t} can only influence which agents are selected from S_{3-t}. However, even without Theorem 3.1 it is easy to see that the mechanism does not yield a finite approximation ratio, since it might be the case that the only edges in the graph are between agents in the same subset.

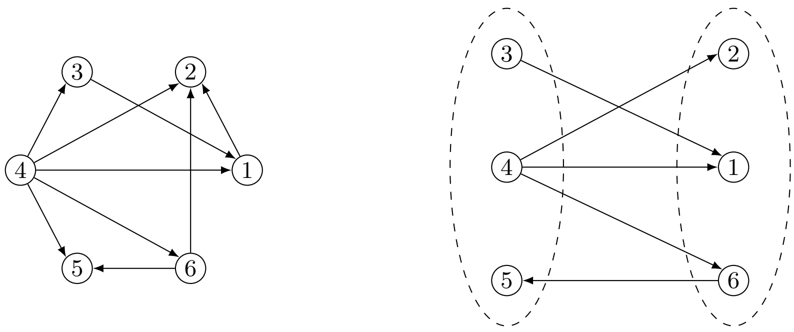
We leverage and refine the partition idea in order to design a randomized SP mechanism that yields a constant approximation ratio. More accurately, we define an infinite family of mechanisms, parameterized by a parameter $m \in \mathbb{N}$. Given m, the mechanism randomly partitions the set of agents into m subsets, and then selects (roughly) the top k/m agents from each subset, based only on the incoming edges from agents in other subsets. Below we give a more formal specification of the mechanism; an example can be found in Figure 1.

The Random m-Partition Mechanism (m-RP)

1. Assign each agent independently and uniformly at random to one of m subsets $S_1, \ldots, S_m$.

2. Let $T \subset \{1, \ldots, m\}$ be a random subset of size $k - m \cdot \lfloor k/m \rfloor$.

3. If $t \in T$, select the $\lceil k/m \rceil$ agents from S_t with highest indegrees based only on edges from $N \setminus S_t$. If $t \notin T$, select the $\lfloor k/m \rfloor$ agents from S_t with highest indegrees based



(a) The given graph (b) The partitioned graph

Figure 1: Example for the Random 2-Partition Mechanism, with $n = 6$ and $k = 2$. Figure 1(a) illustrates the given graph. The mechanism randomly partitions the agents into two subsets, shown in Figure 1(b), and disregards the edges inside each group. The mechanism then selects the best agent in each group based on the incoming edges from the other group; in the example, the selected subset is $\{1, 5\}$, with a sum of indegrees of four, whereas the optimal subset is $\{2, 5\}$, with a sum of indegrees of five.

only on edges from $N \setminus S_t$. Break ties lexicographically in both cases. If one of the subsets S_t is smaller than the number of agents to be selected from this subset, select the entire subset.

4. If only $k' < k$ agents were selected in Step 3, select $k - k'$ additional agents uniformly from the set of agents that were not previously selected.

Note that if $k = 1$ and $m = 2$ then we select one agent from one of the two subsets, based on the incoming edges from the other. In this case, step 2 is equivalent to a toss of a fair coin that determines from which of the two subsets we select an agent.

As in the deterministic case, given a partition of the agents into subsets $S_1, \ldots, S_m$, the choice of agents that are selected from S_t is independent of their outgoing edges. Furthermore, the partition is independent of the input. Therefore, m-RP is SP.[3] The following theorem explicitly states the approximation guarantees provided by m-RP; the technical and rather delicate proof of the theorem is relegated to the full version of this paper.

Theorem 4.1. *Let $N = \{1, \ldots, n\}$, $k \in \{1, \ldots, n-1\}$. For every value of m, m-RP is SP. Furthermore,*

1. *2-RP has an approximation ratio of four, and*
2. *$(\lceil k^{1/3} \rceil)$-RP has an approximation ratio of $1 + \mathcal{O}(1/k^{1/3})$.*

In fact, we can choose the best value of m for any given value of k when we apply m-RP. In other words, Theorem 4.1 implies that for every k there exists an SP mechanism with an approximation ratio of $\min\{4, 1 + \mathcal{O}(1/k^{1/3})\}$, that is, an approximation ratio that is bounded from above by four for any value of k, and approaches one as k grows.

It follows from the theorem that, for $k = 1$, 2-RP has an approximation ratio of four; for this case m-RP with $m > 2$ has a strictly worse ratio. It is interesting to note that the analysis is tight. Indeed, consider a graph $G = (N, E)$ with only one edge from agent 1 to agent n, that is, $E = \{(1, n)\}$. Assume without loss of generality that agent n is assigned to S_1. In order for agent n to be selected, two events must occur:

[3]The mechanism is even *universally SP*, see Section 5.

1. $T = \{1\}$, that is, the winner must be selected from S_1. This happens with probability $1/2$.

2. Either $1 \in S_2$, or $|S_1| = 1$. The probability that $1 \in S_2$ is $1/2$. The probability that $|S_1| = 1$, given that $n \in S_1$, is $1/2^{n-1}$. By the union bound, the probability of this event is at most $1/2 + 1/2^{n-1}$.

It is clear that n cannot be selected unless the first event occurs. If the second event does not occur, it follows that n has an indegree of zero based on the incoming edges from S_2, and there are other alternatives in S_1 (which also have an indegree of zero). Since tie-breaking is lexicographic, agent n would not be selected. As the two events are independent, the probability of both occurring is therefore at most $1/4 + 1/2^n$. We conclude that the approximation ratio of the mechanism cannot be smaller than

$$\frac{1}{\left(\frac{1}{4} + \frac{1}{2^n}\right) \cdot 1} = 4 - \mathcal{O}\left(\frac{1}{2^n}\right).$$

We next provide a very simple, though rather weak, lower bound for the approximation ratio yielded by randomized SP k-selection mechanisms. Let $k \in \{1, \ldots, n-1\}$, and let $f : \mathcal{G} \to \Delta(\mathcal{S}_k)$ be a randomized SP k-selection mechanism. Consider the graph $G = (N, E)$ where

$$E = \{(i, i+1) : \ i = 1, \ldots, k\} \cup \{(k+1, 1)\},$$

i.e., E is a directed cycle on the agents $1, \ldots, k+1$. Then there exists an agent $i \in \{1, \ldots, k+1\}$, without loss of generality agent 1, that is included in $f(G)$ with probability at most $k/(k+1)$. Now, consider the graph G' where $E' = E \setminus \{(1, 2)\}$, that is, agent 1 removes its outgoing edge to agent 2. By strategyproofness, agent 1 is included in $f(G')$ with probability at most $k/(k+1)$. Any subset $S \in \mathcal{S}_k$ such that $1 \notin S$ has at most $k-1$ incoming edges in G'. It follows that the expected number of incoming edges in $f(G')$ is at most

$$\frac{k}{k+1} \cdot k + \frac{1}{k+1} \cdot (k-1) = \frac{k^2 + k - 1}{k+1}.$$

Hence the approximation ratio of f cannot be smaller than

$$\frac{k}{\frac{k^2+k-1}{k+1}} = 1 + \frac{1}{k^2 + k - 1}. \tag{1}$$

We have therefore proved the following easy result.

Theorem 4.2. *Let $N = \{1, \ldots, n\}$, $n \geq 2$, $k \in \{1, \ldots, n-1\}$. Then there is no randomized SP k-selection mechanism with an approximation ratio smaller than $1 + \Omega(1/k^2)$.*

Not surprisingly, the lower bound given by Theorem 4.2 converges to one, albeit more quickly than the upper bound of Theorem 4.1. As usual, an especially interesting special case is when $k = 1$. Equation (1) gives an explicit lower bound of two for this case. On the other hand, Theorem 4.1 gives an upper bound of four. We conjecture that the correct value is two.

Conjecture 4.3. *There exists a randomized SP 1-selection mechanism with an approximation ratio of two.*

One deceptively promising avenue for proving the conjecture is designing an iterative version of the Random Partition Mechanism. Specifically, we start with an empty subset $S \subset N$, and at each step add to S an agent from $N \setminus S$ that has minimum indegree based on the incoming edges from S, breaking ties randomly (so, in the first step we would just add to S a random agent). The last agent that remains outside S is selected. This SP mechanism does remarkably well on some difficult instances, but fails spectacularly on a contrived counterexample. A detailed discussion of the mechanism and the illuminating counterexample is deferred to the full version of this paper.

4.2 GSP Randomized Mechanisms

In the beginning of Section 4.1 we identified a trivial randomized SP k-selection mechanism, namely the one that selects a subset of k agents at random. Of course this mechanism is even GSP, since the outcome is completely independent of the reported graph.

We claim that selecting a random k-subset gives an approximation ratio of n/k. Indeed, consider an optimal subset $K^* \subseteq N$ with $|K^*| = k$. Each agent $i \in K^*$ is included in the selected subset with probability k/n, and hence in expectation contributes a (k/n)-fraction of its indegree to the expected total indegree of the selected subset. By linearity of expectation, the expected total indegree of the selected subset is at least a (k/n)-fraction of the total indegree of K^*.

Theorem 4.1 implies that we can do much better if we just ask for strategyproofness. If one asks for *group* strategyproofness, on the other hand, just selecting a random subset turns out to be optimal up to a tiny gap. It it worth noting that the following result holds even if one is merely interested in coalitions of size at most two. The proof is given in the full version of this paper.

Theorem 4.4. *Let $N = \{1, \ldots, n\}$, $n \geq 2$, and let $k \in \{1, \ldots, n-1\}$. No randomized GSP k-selection mechanism can yield an approximation ratio smaller than $(n-1)/k$.*

5 Discussion

In this section we discuss the significance of our results and state some open problems.

Payments. If payments are allowed and the preferences of the agents are quasi-linear then truthful implementation of the optimal solution is straightforward: simply give one unit of payment to each agent that is not selected. This can be refined by only paying "pivotal" agents that are not selected, that is, agents that would have been selected had they lied. However, even under the latter scheme we may have to pay all the non-selected agents (e.g., when the graph is a clique). Moreover, a simple argument shows that there is no truthful payment scheme that does better.

The utility model. We have studied an "extreme" utility model, where an agent is only interested in the question of its own selection. The restriction of the preferences of the agents allows us to circumvent impossibility results that hold with respect to more general preferences, e.g., the Gibbard-Satterthwaite Theorem [15, 25] and its generalization to randomized rules [16].

A more practical assumption would be that an agent receives a utility of one if it is selected, plus a utility of $\beta \geq 0$ for each of its (outgoing) neighbors that is selected. In this case the social welfare (sum of utilities) of a set S of selected agents is k plus β times the total indegree of S. Hence, if $\beta > 0$, a set S maximizes social welfare if and only if it maximizes the total indegree. In particular, if $\beta > 0$ and payments are available, we can use the VCG mechanism [27, 7, 18] (see [22] for an overview) to maximize the total indegree in a truthful way.

It is easy to see that the lower bound of Theorem 3.1 for the 0–1 model also holds for the β–1 model if β is small. The latter is likely to be the case in many practical settings, such as those described in Section 1. Upper bounds identical to those of Theorem 4.1 hold for any value of β. In particular, m-RP remains strategyproof in the β–1 model, as the probability that an agent is selected increases in the number of votes it receives. Moreover, if β is small, a variation on the random partition mechanism achieves an approximation ratio close to one with respect to social welfare, even when $k = 1$. If $\beta \geq 1$ then simply selecting the optimal solution (and breaking ties lexicographically) is SP.

Robustness of the impossibility result. Theorem 3.1 provides a strong impossibility result for deterministic mechanisms. We have seen that this result is rather sensitive to the model, and no longer holds if one is allowed to select *at most* k agents rather than exactly k, or if each agent is forced to report at least one outgoing edge. That said, we note that these particular aspects of the model are crucial: in our motivating examples, and in approval voting in general, an agent may choose not to report any outgoing edges; in essentially all conceivable applications the set of agents to be selected is of fixed size.

Weights and an application to conference reviews. A seemingly natural generalization of our model can be obtained by allowing weighted edges. Interestingly, our main positive result, namely Theorem 4.1, also holds in this more general setting (subject to minor modifications to its formulation and proof). However, closer scrutiny reveals that it is our target function that is often meaningless in the weighted setting. Indeed, the absence of an edge between i and j would in this context imply that i has no information about j, whereas an edge with small weight would imply that i dislikes or distrusts j. Therefore, maximizing the sum of weights on incoming edges may not be desirable.

That said, in very specific situations maximizing the sum of weights on incoming edges makes perfect sense; one prominent example is conference reviews. In this context the reviewers assign scores to papers while often submitting a paper of their own, and a subset of papers must be selected. This setting is special since it is usually the case that each paper is reviewed by three reviewers, i.e., each agent has exactly three incoming weighted edges, hence maximizing the sum of scores is the same as maximizing the average score. We conclude that m-RP can be employed to build a truthful conference program!

Universal strategyproofness vs. strategyproofness in expectation. In the context of randomized mechanisms, two flavors of strategyproofness are usually considered. A mechanism is *universally SP* if for every fixed outcome of the random choices made by the mechanism an agent cannot gain by lying, that is, the mechanism is a distribution over SP mechanisms. A mechanism is *SP in expectation* if an agent cannot increase its expected utility by lying. In this paper we have used the latter definition, which clearly is the weaker of the two. On the one hand, this strengthens the randomized SP lower bound of Theorem 4.2. On the other hand, notice that the randomized mechanisms of Section 4 are in fact universally SP. Indeed, for every fixed partition, selecting agents from one subset based on incoming edges from other subsets is SP. Hence, Theorem 4.1 is even stronger than originally stated.

Open problems. Our most enigmatic open problem is the gap for randomized SP 1-selection mechanisms: Theorem 4.1 gives an upper bound of four, while Theorem 4.2 gives a lower bound of two. We conjecture that there exists a randomized SP 1-selection mechanism that gives a 2-approximation.

In addition, a potentially interesting variation of our problem can be obtained by changing the target function. One attractive option is to maximize the minimum indegree in the selected subset. Clearly, our total impossibility for deterministic SP mechanisms (Theorem 3.1) carries over to this new target function. However, it is unclear what can be achieved using randomized SP mechanisms.

6 Acknowledgments

We thank Moshe Babaioff, Liad Blumrosen, Michal Feldman, Gil Kalai, David Parkes, Yoav Shoham, and Aviv Zohar for valuable discussions.

References

[1] N. Alon, M. Feldman, A. D. Procaccia, and M. Tennenholtz. Strategyproof approximation of the minimax on networks. *Mathematics of Operations Research*. Forthcoming.

[2] N. Alon and M. Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM Journal on Computing*, 22:403–417, 1993.

[3] A. Altman and M. Tennenholtz. Strategyproof deterministic lotteries under broadcast communication. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1549–1552, 2008.

[4] S. Antonakopoulos. Fast leader-election protocols with bounded cheaters' edge. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 187–196, 2006.

[5] S. J. Brams and P. C. Fishburn. *Approval Voting*. Springer, 2nd edition, 2007.

[6] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proceedings of the 3rd Workshop on the Economics of Peer-to-Peer Systems (P2PECON)*, pages 128–132, 2005.

[7] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[8] J. Cooper and N. Linial. Fast perfect-information leader-election protocols with linear immunity. *Combinatorica*, 15:319–332, 1995.

[9] O. Dekel, F. Fischer, and A. D. Procaccia. Incentive compatible regression learning. *Journal of Computer and System Sciences*. Forthcoming. Earlier version appeared in the Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).

[10] S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 644–652, 2006.

[11] U. Feige. Noncryptographic selection protocols. In *Proceedings of the 40th Symposium on Foundations of Computer Science (FOCS)*, pages 142–152, 1999.

[12] U. Feige, A. Flaxman, J. D. Hartline, and R. Kleinberg. On the competitive ratio of the random sampling auction. In *Proceedings of the 1st International Workshop on Internet and Network Economics (WINE)*, pages 878–886, 2005.

[13] A. Fiat, A. V. Goldberg, J. D. Hartline, and A. R. Karlin. Competitive generalized auctions. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 72–81, 2002.

[14] E. Friedman, P. Resnick, and R. Sami. Manipulation-resistant reputation systems. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 27. Cambridge University Press, 2007.

[15] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[16] A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45:665–681, 1977.

[17] A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright. Competitive auctions. *Games and Economic Behavior*, 55:242–269, 2006.

[18] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[19] R. LeGrand, E. Markakis, and A. Mehta. Some results on approximating the minimax solution in approval voting. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1193–1195, 2007.

[20] P. Lu, X. Sun, Y. Wang, and Z. A. Zhu. Asymptotically optimal strategy-proof mechanisms for two-facility games. In *Proceedings of the 11th ACM Conference on Electronic Commerce (ACM-EC)*, pages 315–324, 2010.

[21] P. Lu, Y. Wang, and Y. Zhou. Tighter bounds for facility games. In *Proceedings of the 5th International Workshop on Internet and Network Economics (WINE)*, pages 137–148, 2009.

[22] N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9. Cambridge University Press, 2007.

[23] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1–2):166–196, 2001.

[24] A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. In *Proceedings of the 10th ACM Conference on Electronic Commerce (ACM-EC)*, pages 177–186, 2009.

[25] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[26] J. Schummer and R. V. Vohra. Mechanism design without money. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 10. Cambridge University Press, 2007.

[27] W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

Noga Alon
Microsoft Israel R&D Center
13 Shenkar Street, Herzeliya 46725, Israel
and Schools of Mathematics and Computer Science
Tel Aviv University, Tel Aviv, 69978, Israel
Email: `nogaa@tau.ac.il`

Felix Fischer and Ariel D. Procaccia
Harvard School of Engineering and Applied Sciences
33 Oxford Street, Cambridge, MA 02138, USA
Email: `{fischerf,arielpro}@seas.harvard.edu`

Moshe Tennenholtz
Microsoft Israel R&D Center
13 Shenkar Street, Herzeliya 46725, Israel
and Technion, IIT
Haifa 32000, Israel.
Email: `moshet@microsoft.com`

Complexity Consideration on the Existence of Strategy-proof Social Choice Functions

Koji Takamiya

Abstract

Social choice theorists have long recognized that in models of private goods economies, strategy-proofness is sometimes incompatible with individual rationality plus Pareto efficiency, and that it is usually more or less "difficult" to prove this incompatibility. In this paper we examine this "difficulty" from the viewpoint of computational complexity. We set up a simple model of private goods exchange where agents bring in and trade indivisible objects under consumption constraints. We consider the computational problem of deciding whether for a given specification of the economy, there exists a social choice function which is strategy-proof, individually rational and Pareto efficient. We prove that (i) this is an $\mathcal{NP}$-hard problem, and point out, however, that (ii) the problem becomes computationally trivial if we drop one of these three properties of the social choice function.

1 Introduction

In the traditional literature of social choice, it has been a central issue to investigate the *existence* of social choice procedures which satisfy various desirable properties from the viewpoints of incentive, efficiency, equity and so on. Among many themes in this realm, the existence of *strategy-proof* social choice functions has attracted significant attention for many years. It has been long recognized that strategy-proofness often conflicts with other desirable properties. And not only that there are conflicts but also it is often *difficult* to establish that there *is* indeed a conflict, i.e. to prove that strategy-proofness is incompatible with some other desirable properties. For example, the celebrated Gibbard-Satterthwaite theorem (Gibbard 1973, Satterthwaite 1975) depicts the conflict between strategy-proof and *non-dictatorship*, a very weak requirement of equity. And this conflict was difficult to establish: The Gibbard-Satterthwaite theorem had been conjectured many years before it was proved. It took a long time for the theorem to be proved.

More recent studies on strategy-proof functions in models of *private goods economies* have revealed the conflict between strategy-proofness and *individual rationality plus Pareto efficiency*, which is also often difficult to establish. For example, in 1972, Hurwicz proved that for any classical pure exchange economy with *two persons* and *two goods*, if the preference domain includes a sufficiently wide set of classical preferences, then there does not exist a social choice function which is strategy-proof, individually rational and Pareto efficient (Hurwicz, 1972). He conjectured that the same result holds true for those economies with three or more agents and goods. However, this problem had remained unsolved for about thirty years until Serizawa's work appeared (Serizawa, 2002), which proved that Hurwicz's result is generalized to the case of any finite numbers of agents and goods.

Another example is from the theory of matching models such as of the *marriage problem* (Gale and Shapley, 1962) and the *housing market* (Shapley and Scarf, 1972). It had been known from the early 1980's that (i) for the marriage problem with the full strict preference domain[1], no *core stable rule* (i.e. a social choice function which chooses a core stable

[1] The *full strict preference domain* is the domain where each agent can have any strict ranking over the agent's own assignments (i.e. no consumption externalities). This domain is usually assumed in social choice analysis of matching problems.

matching for each preference profile) is strategy-proof, and that (ii) in contrast, for the housing market also with the full strict preference domain, there exists the unique core stable rule and this rule is strategy-proof (Roth, 1982a; Roth 1982b). (Note that any core stable rule is both individually rational and Pareto efficient.) Clearly these two results exhibit a sharp contrast. However, for a long time, it had not been fully understood where this sharp contrast came from. It was 1999 when Sönmez provided an answer to this question: He set up a general model of indivisible objects allocation, which covers both the marriage problem and the housing market, and proved the following: Provided that the preference domain is the full strict preference domain, if a social choice function is strategy-proof, individually rational and Pareto efficient, then it must be that for each preference profile, the core (i.e. the set of core stable allocations) is a singleton unless it is empty, and that this function chooses the core stable allocation whenever available. In the marriage problem the core is neither a singleton nor empty for some preference profile. Thus Sönmez's result implies the nonexistence of strategy-proof functions which are individaully raiotional and Pareto efficient in the marriage problem. Later, Takamiya (2003) showed a conditional converse of Sönmez's result: Provided that the preference domain is the full strict preference domain, if the core is a singleton for each preference profile, then the unique core stable rule is strategy-proof. Evidently the strategy-proofness of the core stable rule in the housing market immediately follows from this result. These two results of Sönmez and Takamiya have provided some understanding in the existence problem of strategy-proof functions by relating it to the singletonness of the core. However, to this date, it has not been fully investigated under what conditions the singletonness of the core is obtained in the general setting formulated by Sönmez. This seems to be a hard combinatorial problem.[2]

To date, social choice theorists know from experience (partially described as above) that in models of private goods economies, it is usually more or less difficult to decide whether there exists a social choice function which satisfies these three properties altogether. This is in contrast to that it is also known that it is usually easy to obtain strategy-proof functions which are individually rational *or* Pareto efficient *separately.* For example, in most models of private goods economies, it is trivial to have a strategy-proof function which is Pareto efficient only: A *dictatorial* function, in which some fixed agent always receives all the goods in the economy, is both strategy-proof and Pareto efficient.

The purpose of the present research is to examine the idea that *in private goods economies, it is difficult to determine whether there exists a social choice function which is strategy-proof, individually rational and Pareto efficient.* Our approach is metaphorical in the sense that we do not directly analyze those problems which social choice theorists have attacked or do not go into the ingenuity of their proofs. Rather, for our analytical purpose, we set up a simple and artificial problem and analyze its difficulty of a specific kind: To embody the concept of "difficulty" we employ the concept of *time complexity* in the theory of *computational complexity.*

Concretely, our analysis is as follows: We give a simple model of private goods economies where agents bring in and trade indivisible objects. There each agent is faced with a consumption constraint. This model is a special case of the general model of indivisible objects allocation formulated in Sönmez's above-mentioned paper. We consider the computational problem of deciding whether for a given specification of the economy (i.e. a given instance of the problem), there exists a social choice function which is strategy-proof, individually rational and Pareto efficient. First, for our main theorem, we prove that this is an $\mathcal{NP}$-hard problem. Here $\mathcal{NP}$-hardness captures the idea of "difficulty" in deciding the existence of

[2]This problem has been partially solved: In the setting of the *coalition formation problem*, a special case of Sönmez's general model, Pápai (2004) has provided a necessary and sufficient condition for the core to be a singleton for each preference profile in the full strict preference domain. However, to our knowledge, the computational complexity of checking this condition has not been investigated.

such functions. Second, we point out, however, that this problem becomes computationally *trivial* if we drop one of these three properties of the social choice function. That is, for any two properties out of these three properties, for *any* instance of the problem there exists at least one social choice function which satisfies these two properties. Thus the answer to the decision problem is always "yes".

It is important to note that the nature of our research is different from that of most lines of research in computational social choice. Usually in computational social choice theory, computational ideas are used to formulate and analyze various realistic constraints put on the prosecution of social choice procedures or the behavior of agents who act in the social choice process, which arise from the limited availability of material and mental resources. However, here computational ideas are employed to express the idea of the difficulty which (traditional) social choice researchers face with when they look for desirable strategy-proof functions. In this sense, our paper is still research in computational social choice but more precisely is "research about the traditional social choice research from the viewpoint of computation".

2 Preliminaries

2.1 Economic Model

Let us define the economic model that we examine. We consider modeling reallocation of multiple indivisible objects. An **allocation problem** is a list $\mathcal{E} = (N, \Omega, \{\Theta^i\}_{i \in N}, u, (w, q), x_0)$. Here N is the set of **agents**, and Ω is the set of **(indivisible) objects**. N and Ω are both assumed to be nonempty finite sets.

An **allocation** is a set-valued function $x : N \rightarrow\rightarrow \Omega$ which is "partitional," i.e. (i) $i \neq j \Longrightarrow x(i) \cap x(j) = \emptyset$ and (ii) $\bigcup_N x(i) = \Omega$. Let us denote the set of allocations by $\mathcal{X}$.

$u := \{u^i\}_{i \in N}$ are **utility functions**, and Θ^i is the **type space** of agent i. Each element of Θ^i specifies the preference of agent i in the following way: A **value function** $v : (\Theta^1 \cup \cdots \cup \Theta^n) \times \Omega \to \mathbb{Z}$ is defined. We assume utility functions are all *additive* with respect to the values given by the value function. That is, the utility function of agent i, $u^i : \mathcal{X} \times \Theta^i \to \mathbb{Z}$ is defined so as to satisfy the following: for all $x \in \mathcal{X}$ and $\theta^i \in \Theta^i$,

$$u^i(x, \theta^i) = \sum_{\omega \in x(i)} v(\theta^i, \omega). \tag{1}$$

Note that values $v(\theta^i, \cdot)$ could be negative and so are utility levels.

(w, q) is a **feasibility constraint**, which consists of **weights** w and **capacities** q. Here w is a function $w : N \times \Omega \to \mathbb{Z}_+$, and q is a function $q : N \to \mathbb{Z}_+$. Here $\mathbb{Z}_+ := \{0, 1, 2, \cdots\}$. For agent $i \in N$, object $\omega \in \Omega$ has a weight $w(i, \omega)$, and i can consume a bundle of objects unless the sum of the weights of these objects exceeds i's capacity $q(i)$. Thus it is defined that an allocation $x \in \mathcal{X}$ is **feasible** to agent $i \in N$ if

$$\sum_{\omega \in x(i)} w(i, \omega) \leq q(i). \tag{2}$$

An allocation $x \in \mathcal{X}$ is called **feasible** if it is feasible to all the agents. Let us denote the set of feasible allocations by $\mathcal{X}^f$. In the following we refer to feasible allocations simply as allocations.

Finally, x_0 denotes the **initial endowments**. We assume $x_0 \in \mathcal{X}^f$.

Let $x \in \mathcal{X}^f$ and $\theta \in \Theta$. Then x is **individually rational** at θ if for any $i \in N$,

$$u^i(x, \theta^i) \geq u^i(x_0, \theta^i). \tag{3}$$

And x is **Pareto efficient** at θ if for any $y \in \mathcal{X}^f$

$$[\forall i \in N,\ u^i(y, \theta^i) \geq u^i(x, \theta^i)] \Longrightarrow [\forall i \in N,\ u^i(y, \theta^i) = u^i(x, \theta^i)]. \tag{4}$$

2.2 Relevance of the model

Our model is a special case of the general allocation model studied by Sönmez (1999) (which we have mentioned in Sec.1). To Sönmez's model, we have added specific structures on admissible preferences and feasible allocations, namely, *additivity* of utilities, *weights* of objects, *capacities* of agents. These structures admit concise representations of feasible allocations and preferences. Without such structures, inputs of the problem can be overly redundant, which apparently reduces the complexity of the problem.

We admit that as a model of private goods economies, our model is unusual and perhaps artificial in assuming weights and capacities. However, it is still relevant as a modeling of economic problems. For example, our model includes the housing market (Shapley and Scarf, 1974), an important economic model, as a special case. Further, in some cases, we may interpret weights as *personalized prices* of objects and capacities as *budgets* that agents face.

2.3 Properties of social choice functions

Let an allocation problem be given. Let us denote $\Theta := \Theta^1 \times \Theta^2 \times \cdots \times \Theta^n$. Any element θ of Θ is called a **type profile**. A **social choice function** is a function $f : \Theta \to \mathcal{X}^f$. We consider the following three properties of social choice functions.

- **Strategy-proofness.** Let $i \in N$ and $\theta \in \Theta$. Then we say that i **manipulates** f at θ if for some $\tilde{\theta}^i \in \Theta^i$,

  $$u^i(f(\theta^{-i}, \tilde{\theta}^i), \theta^i) > u^i(f(\theta^{-i}, \theta^i), \theta^i), \tag{5}$$

  f is called **strategy-proof** if for any $i \in N$, i cannot manipulate f at any $\theta \in \Theta$.

- **Individual rationality.** Let us call f **individually rational** if for any $\theta \in \Theta$, x is individually rational at θ.

- **Pareto efficiency.** Let us call f **Pareto efficient** if for any $\theta \in \Theta$, x is Pareto efficient at θ.

3 Results

3.1 Main theorem

We consider the decision problem in the following. Let a positive integer $\bar{n}$ be given.

NAME: $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(\bar{n})$
INSTANCE: An allocation problem $\mathcal{E} = (N, \Omega, \{\Theta^i\}_{i \in N}, u, (w, q), x_0)$ with $|N| = \bar{n}$.
QUESTION: Does there exist a social choice function for $\mathcal{E}$ which is strategy-proof, individually rational and Pareto efficient?

Our main result is that $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(\bar{n})$ is $\mathcal{NP}$*-hard.* Note that in our formulation of the computational problem above, the number of agents is fixed, i.e. $|N| = \bar{n}$. Without this restriction on the number of agents, it is not much surprising even if the problem would be computationally hard because the space of type profiles grows exponentially as the number of agents gets larger.

Theorem 1 *The problem* $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(\bar{n})$ *is* $\mathcal{NP}$*-hard if* $\bar{n} \geq 4$.

The construction made in our proof of Theorem 1 requires four agents at least ($\bar{n} \geq 4$). We do not know whether $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(\bar{n})$ is $\mathcal{NP}$-hard if $\bar{n} = 2$ or 3.

3.2 Interpretation of the main theorem

(1) To understand the subtlety of Theorem 1, we should be aware of the following fact.

Theorem 2 *Let an allocation problem* $\mathcal{E} = (N, \Omega, \{\Theta^i\}_{i \in N}, u, (w, q), x_0)$ *with an arbitrary size of* $|N|$ *be given. And let us pick up any two of the three properties, strategy-proofness, individual rationality and Pareto efficiency. Then there exists a social choice function which satisfies these two properties.*

Theorem 2 says that if we drop one of the three properties of social choice functions which are listed in the problem $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(\bar{n})$, then the computational problem becomes *trivial*: The answer is "yes" for *any instance*. This fact tells us that what makes the computational problem hard is neither each single requirement of strategy-proofness, individual rationality or Pareto efficiency, or even each *pair* of these three properties, but rather is the *combination* of these *three* properties altogether.

(2) It is important to notice that what is at issue here is the computational problem deciding the *existence* of social choice functions which satisfy some properties. One should carefully distinguish this problem from the computational problem of deciding whether a given social choice function satisfies those properties. In fact, the latter problem can be computationally very hard without combining these three properties. For example, if we are given an allocation problem and a type profile is fixed, then the problem of deciding whether a given allocation is not Pareto efficient is $\mathcal{NP}$-complete. To state more precisely, the following theorem holds. Let us define the following problem: Let a natural number $\bar{n}$ be given.

NAME: NOTPARETO($\bar{n}$)
INSTANCE: An allocation problem $\mathcal{E} = (N, \Omega, \{\Theta^i\}_{i \in N}, u, (w, q), x_0)$ with $|N| = \bar{n}$ and $\Theta = \{\theta\}$; and an allocation $x \in \mathcal{X}^f$.
QUESTION: Is x not Pareto efficient?

Theorem 3 *The problem* NOTPARETO($\bar{n}$) *is* $\mathcal{NP}$*-complete if* $\bar{n} \geq 2$.

From the above theorem, it directly follows that the problem of deciding whether a given social choice function is not Pareto efficient is also $\mathcal{NP}$-complete. On the contrary, it is computationally trivial to decide whether a Pareto efficient social choice function *exists* because such a function always exists.

Our Theorem 3 follows from Theorem 1 in the paper of de Keijzer, Bouveret, Klos and Zhang (2009), which studies computational problems arising from an allocation model of indivisible objects with additive utilities.[3] However, in Sec.3.3, we will give our own proof of Theorem 3, which utilizes a construction used in our proof of Theorem 1.

3.3 Proofs

For the preparation of proving Theorem 1, let us consider the following allocation problem $\mathcal{E}_1$.

- $N = \{1, 2, 3\}$.

[3]We are thankful to an anonymous referee for notifying us of the work of de Keijzer *et al.*

- $\Omega = \{c_1, c_2, c_3\}$.
- $x_0^1 = \{c_1\}$; $\quad x_0^2 = \{c_2, c_3\}$; $\quad x_0^3 = \emptyset$.
- $\Theta^1 = \{\theta_1^1, \theta_2^1, \theta_3^1, \theta_4^1, \theta_5^1, \theta_6^1\}$; $\quad \Theta^2 = \{\theta_1^2, \theta_2^2, \theta_3^2, \theta_4^2, \theta_5^2, \theta_6^2\}$; $\quad \Theta^3 = \{\theta^3\}$.
- The following table depicts the values of $v(\theta_j^1, c_k)$, $v(\theta_j^2, c_k)$ and $v(\theta^3, c_k)$.

Table 1.

	θ_1^1	θ_2^1	θ_3^1	θ_4^1	θ_5^1	θ_6^1	θ_1^2	θ_2^2	θ_3^2	θ_4^2	θ_5^2	θ_6^2	θ^3
$v(\cdot, c_1)$	1	1	2	2	3	3	3	3	2	2	1	1	0
$v(\cdot, c_2)$	2	3	1	3	1	2	2	1	3	1	3	2	0
$v(\cdot, c_3)$	3	2	3	1	2	1	1	2	1	3	2	3	0

- The following table depicts the values of $w(i, c_k)$ and $q(i)$.

Table 2.

	1	2	3
$w(\cdot, c_1)$	1	1	0
$w(\cdot, c_2)$	1	1	0
$w(\cdot, c_3)$	1	1	0
$q(\cdot)$	1	2	0

Lemma 1 *For the allocation problem $\mathcal{E}_1$, there does not exist any social choice function that is strategy-proof, individually rational and Pareto efficient.*

Proof. Suppose that f is strategy-proof, individually rational and Pareto efficient. Let us denote allocations $x \in \mathcal{X}^f$ by 3-tuples, i.e. $x = (x(1), x(2), x(3))$. Let

$$x_1 = (\{c_1\}, \{c_2, c_3\}, \emptyset), \quad x_2 = (\{c_2\}, \{c_3, c_1\}, \emptyset), \quad x_3 = (\{c_3\}, \{c_1, c_2\}, \emptyset). \tag{6}$$

Since f is Pareto efficient, for all $\theta \in \Theta$, $f(\theta) \in \{x_1,\ x_2,\ x_3\}$. Given the above, clearly, for each of agents 1 and 2, the agent's possible preferences can be regarded as the set of strict rankings over $\{x_1,\ x_2,\ x_3\}$. Further, we can ignore the existence of agent 3. Therefore, f is regarded as a social choice function with three alternatives and two agents whose admissible preferences are exactly the set of strict rankings of the three alternatives. Then since f is strategy-proof and Pareto efficient, by the Gibbard-Satterthwaite theorem (Gibbard 1973, Satterthwaite 1975) f is dictatorial, i.e. there exists some $i \in \{1, 2\}$ such that for any $\theta \in \Theta$, $f(\theta)$ equals the allocation that maximizes agent i's utility at θ^i. Clearly f violates individual rationality. Thus we reach the desired conclusion. $\square$

Proof of Theorem 1. Clearly it suffices to prove only for the case where $\bar{n} = 4$ because one can increase the number of agents by adding dummy agents who does not have any initial assignments and is not able to receive any objects for the capacity constraint. The proof is done by reduction from the following problem PARTITION ([SP 12] in Gary and Johnson (1979)).

NAME: PARTITION
INSTANCE: A finite set $A = \{a_1, a_2, \ldots, a_p\}$ and a function $s : A \to \mathbb{N}$.
QUESTION: Does there exist a partition $\{A_1, A_2\}$ of A such that $\sum_{a \in A_1} s(a) =$

$\sum_{a \in A_2} s(a)$.

Let an instance of PARTITION (A, s) be given. Then we give a polynomial-time transformation of this instance into an instance of $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(4)$ in the following, and we will show that the answer to this instance of PARTITION is "yes" if and only if the the answer to this instance of $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(4)$ is "yes".

Let us consider the following instance $\mathcal{E}_2$ of $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(4)$. We denote $\sum_{a \in A} s(a)$ by $s(A)$ in the sequel.

- $N = \{1, 2, 3, 4\}$.
- $\Omega = A \cup \{b, c_1, c_2, c_3\}$.
- $x_0^1 = \{c_1\}$; $\quad x_0^2 = \{c_2, c_3\}$; $\quad x_0^3 = A$; $\quad x_0^4 = \{b\}$.
- $\Theta^1 = \{\theta_1^1, \theta_2^1, \theta_3^1, \theta_4^1, \theta_5^1, \theta_6^1\}$; $\quad \Theta^2 = \{\theta_1^2, \theta_2^2, \theta_3^2, \theta_4^2, \theta_5^2, \theta_6^2\}$; $\quad \Theta^3 = \{\theta^3\}$; $\quad \Theta^4 = \{\theta^4\}$.
- The following table depicts the values of $v(\cdot, \cdot)$.

Table 3.

	θ_1^1	θ_2^1	θ_3^1	θ_4^1	θ_5^1	θ_6^1	θ_1^2	θ_2^2	θ_3^2	θ_4^2	θ_5^2	θ_6^2	θ^3	θ^4
$v(\cdot, c_1)$	1	1	2	2	3	3	3	3	2	2	1	1	0	0
$v(\cdot, c_2)$	2	3	1	3	1	2	2	1	3	1	3	2	0	0
$v(\cdot, c_3)$	3	2	3	1	2	1	1	2	1	3	2	3	0	0
$v(\cdot, a_1)$	0	0	0	0	0	0	0	0	0	0	0	0	0	$2s(a_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$v(\cdot, a_i)$	0	0	0	0	0	0	0	0	0	0	0	0	0	$2s(a_i)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$v(\cdot, a_p)$	0	0	0	0	0	0	0	0	0	0	0	0	0	$2s(a_p)$
$v(\cdot, b)$	4	4	4	4	4	4	0	0	0	0	0	0	0	$s(A) - 1$

- The following table depicts the values of $w(\cdot, \cdot)$ and $q(\cdot)$.

Table 4.

	1	2	3	4
$w(\cdot, c_1)$	1	1	0	$s(A) + 1$
$w(\cdot, c_2)$	1	1	0	$s(A) + 1$
$w(\cdot, c_3)$	1	1	0	$s(A) + 1$
$w(\cdot, a_1)$	2	3	0	$2s(a_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w(\cdot, a_i)$	2	3	0	$2s(a_i)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w(\cdot, a_p)$	2	3	0	$2s(a_p)$
$w(\cdot, b)$	1	3	1	$s(A)$
$q(\cdot)$	1	2	0	$s(A)$

Note the two key facts of this construction: (a) Agent 3 brings the objects $A = \{a_1, a_2, \ldots, a_p\}$ into the economy as the initial allocation, and these objects are valuable only to agent 4. (b) If agent 4 leaves this economy with his initial assignment and agent 3's initial assignment is deleted, then the resulting economy is identical with $\mathcal{E}_1$ constructed for Lemma 1.

(i) First, we show that if the answer to the PARTITION instance (A, s) is "yes", then that to the $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(4)$ instance constructed above is "yes". Now suppose that the answer to the PARTITION instance is "yes". In the following we show that for any type profile, there is only one allocation which is both individually rational and Pareto efficient, and that the social choice function which chooses this allocation for any type profile (thus individually rational and Pareto efficient) is strategy-proof.

Let us fix a type profile. Given the date above, it is clear that agent 4 is to be better off by releasing the object b and instead collecting some objects A' out of A if and only if A' satisfies $\sum_{a \in A'} 2s(a) = s(A)$. And that the answer to the PARTITION instance (A, s) is "yes" means that such A' exists. This reallocation (agent 4 releases b and obtains A') is Pareto improvement because the objects A are valuable only to agent 4 and chucking out the object b never hurts the other agents' utility. And this fills up agent 4's capacity. Further, Pareto efficiency forces the object b to go to agent 1, and this fills up agent 1's capacity. By Pareto efficiency, agent 2 receives the two his most preferred objects out of $\{c_1, c_2, c_3\}$ depending on agent 2's type, and this fills up agent 2's capacity. Finally agent 3 receives the remaining object from $\{c_1, c_2, c_3\}$ and A. Obviously this allocation is individually rational. This is the unique allocation which is individually rational and Pareto efficient.

Let us consider the social choice function which chooses the unique individually rational and Pareto efficient allocation for each type profile. Agents 1 and 4 receives the same assignment for any type profile, and only the assignments of agents 2 and 3 vary. Agent 2 obtains his most preferred assignment. And agent 3's utility level is constant whatever this agent receives. Therefore, there is no situation where some agent can manipulate the outcome, that is, this function is strategy-proof.

(ii) Second, we show that if the answer to the PARTITION instance (A, s) is "no", then that to the $\mathrm{SP} + \mathrm{IR} + \mathrm{PE}(4)$ instance is "no". Now suppose that the answer to the PARTITION instance is "no". In this case, it is not possible for agent 4 to improve his utility level by receiving some objects from A in return for giving up the object b. Thus individual rationality forces agent 4 to keep the object b that fills up agent 4's capacity. Now for the feasibility constraint, any object in A cannot go to either agents 1 or 2 so agent 3 has to keep all the objects of A. Therefore, by the fact (b) indicated above, now the situation is identical with the economy $\mathcal{E}_1$. Then by applying Lemma 1, we conclude that there does not exist any strategy-proof social choice function which is individually rational and Pareto efficient. □

Proof of Theorem 2. (i) There exists a social choice function which is *individually rational* and *Pareto efficient*. Because it is clear that for every type profile, there exists at least one allocation which is both Pareto efficient and individually rational.

(ii) There exists a social choice function which is *strategy-proof* and *individually rational.* An example is the *constant* function, which always chooses the initial endowments x_0.

(iii) There exists a social choice function which is *strategy-proof* and *Pareto efficient.* A social choice function based on *serial dictatorship* (Satterthwaite and Sonnenschein, 1981) satisfies both properties. In the following we define this class of functions and prove that any function in this class satisfies these two properties: Let π be a bijection from $\{1, 2, \ldots, |N|\}$ to N. For each $\theta \in \Theta$, the sets $C^\pi(\theta, i)$ $(i = 0, 1, 2, \ldots, |N|)$ is defined inductively as follows:

$$C^\pi(\theta, 0) = \mathcal{X}^f, \tag{7}$$

$$\text{for } i = 1, 2, \ldots, |N|, \quad C^\pi(\theta, i) = \arg \max_{x \in C^\pi(\theta,\ i-1)} u^{\pi(i)}(x, \theta^{\pi(i)}). \tag{8}$$

Note that if $i < j$, then $C^\pi(\theta, j) \subset C^\pi(\theta, i)$. A social choice function f is a **serial dictatorship based on** π if for all $\theta \in \Theta$, $f(\theta) \in C^\pi(\theta, |N|)$.

First, we show that for any bijection $\pi : \{1, 2, \ldots, |N|\} \to N$, any serial dictatorship f on π is Pareto efficient: Let $x \in f(\theta)$ and $y \in \mathcal{X}^f$. Suppose $\forall i \in N,\ u^i(y, \theta^i) \geq u^i(x, \theta^i)$. Then, first of all, we have $y \in C^\pi(\theta, 1)$ because otherwise it would be $u^{\pi(1)}(y, \theta^{\pi(1)}) < u^{\pi(1)}(x, \theta^{\pi(1)})$, a contradiction. Next, we note that for any $i \in \{2, 3, \ldots, |N|\}$ if $y \in C^\pi(\theta, i-1)$, then $y \in C^\pi(\theta, i)$ because otherwise it would be also $u^{\pi(i)}(y, \theta^{\pi(i)}) < u^{\pi(i)}(x, \theta^{\pi(i)})$. Consequently, by induction, we have $y \in C^\pi(\theta, |N|)$, which implies $\forall i \in N,\ u^i(y, \theta^i) = u^i(x, \theta^i)$. Thus we conclude that x is Pareto efficient.

Second, it is easy to see that these f are also strategy-proof. Because of the way serial dictatorship is defined, any agent $\pi(i)$ receives one of his best assignments among $C^\pi(\theta, i-1)$. However, $C^\pi(\theta, i-1)$ is fully determined by $(\theta^{\pi(1)}, \ldots, \theta^{\pi(i-1)})$ so agent $\pi(i)$'s reporting of his type does not affect this set. Thus $\pi(i)$ cannot be better off by misreporting his type. $\square$

Proof of Theorem 3. First, we show that NOTPARETO(2) (so NOTPARETO($\bar{n}$) with $\bar{n} \geq 2$) is $\mathcal{NP}$-*hard* by reduction from PARTITION. Let an instance (A, s) of PARTITION be given. Let us give a polynomial-time transformation of this instance into an instance of NOTPARETO(2) as follows. The following construction is based on the same idea as the gadget consisting of agents 3 and 4 in the proof of Theorem 1 above.

- $N = \{1, 2\}$.
- $\Omega = A \cup \{b\}$.
- $x_0^1 = A; \quad x_0^2 = \{b\}$
- The following table depicts the values of v.

Table 5.

	θ^1	θ^2
$v(\cdot, a_1)$	0	$2s(a_1)$
$\vdots$	$\vdots$	$\vdots$
$v(\cdot, a_i)$	0	$2s(a_i)$
$\vdots$	$\vdots$	$\vdots$
$v(\cdot, a_p)$	0	$2s(a_p)$
$v(\cdot, b)$	0	$s(A) - 1$

- The following table depicts the values of w, q.

Table 6.

	1	2
$w(\cdot, a_1)$	0	$2s(a_1)$
$\vdots$	$\vdots$	$\vdots$
$w(\cdot, a_i)$	0	$2s(a_i)$
$\vdots$	$\vdots$	$\vdots$
$w(\cdot, a_p)$	0	$2s(a_p)$
$w(\cdot, b)$	0	$s(A)$
$q(\cdot)$	0	$s(A)$

- The allocation x equals x_0.

Now the allocation x is not Pareto efficient if and only if there exists a subset A' of A such that $\sum_{a \in A'} 2s(a) = s(A)$. (Because if such A' exists, agent 2 can be better off without hurting agent 1's utility by releasing the object b and instead collecting A'.) And that the answer to the PARTITION instance (A, s) is "yes" if and only if such A' exists. This establishes the $\mathcal{NP}$-hardness of NOTPARETO.

Second, it is easy to see NOTPARETO($\bar{n}$) $\in \mathcal{NP}$. If the answer to a NOTPARETO($\bar{n}$) instance is "yes" i.e. the considered allocation x is *not* Pareto efficient, then there exists some other allocation y which Pareto dominates x. Now y is a *certificate* and it can be checked in polynomial time whether y Pareto dominates x. □

Acknowledgments

This research is supported by a Grant-in-Aid for Scientific Research of the Ministry of Education, Culture, Sports, Science and Technology of Japan, No.21730155. I am grateful to Ryuichiro Ishikawa, Nobuyuki Hanaki, Toyotaka Sakai, Tomoichi Shinotsuka, Takuma Wakayama and Naoki Watanabe for various comments, and in particular to Takashi Ui for his invaluable critical comments and to Akira Tanaka for many conversations on various topics in computer science and economics. I also thank participants of the seminars at Tsukuba University and Yokohama National University. And I am very thankful to two anonymous referees of COMSOC 2010 for their comments which have improved this paper. All errors are my own responsibility.

References

[1] de Keijzer B, Bouveret S, Klos T and Zhang Y (2009): On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences, *Proceedings of the First International Conference on Algorithmic Decision Theory*: 98–110.

[2] Hurwicz L (1972): On informationally decentralized systems, In *Decision and Organization.* G. B. McGuire and R. Radner (Eds.) North Holland. Amsterdam.

[3] Gibbard A (1973): Manipulation of voting schemes: A general result, *Econometrica* 41: 587–601.

[4] Gale D and Shapley L (1962): College admissions and the stability of marriage, *American Mathematical Monthly* 69: 9–15.

[5] Garey MR and Johnson DS (1979): *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman. New York.

[6] Pápai S (2004): Unique stability in simple coalition formation games, *Games and Economic Behavior* 48: 337–54.

[7] Roth AE (1982a) The economics of matching: Stability and incentives, *Mathematics of Operations Research* 7: 617–28.

[8] Roth AE (1982b): Incentive compatibility in a market with indivisible goods, *Economics Letters* 9: 127–32.

[9] Satterthwaite MA (1975): Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions, *Journal of Economic Theory* 10: 187–217.

[10] Satterthwaite MA and Sonnenschein H (1981): Strategy-proof allocation mechanisms at differentiable points, *Review of Economic Studies* 48: 587–97.

[11] Serizawa S (2002): Inefficiency of strategy-proof rules for pure exchange economies, *Journal of Economic Theory* 106: 219–241.

[12] Shapley L and Scarf H (1974): On cores and indivisibility, *Journal of Mathematical Economics* 1: 23–37.

[13] Sönmez T (1999): Strategy-proofness and essentially single-valued cores, *Econometrica* 67: 677–89.

[14] Takamiya K (2003): On strategy-proofness and essentially single-valued cores: A converse result, *Social Choice and Welfare* 20: 77–83.

Koji Takamiya
Faculty of Economics
Niigata University
8050 2-no-cho Ikarashi Nishi-ku Niigata 950-2181 JAPAN
Email: takamiya@econ.niigat-u.ac.jp

An Oddity That Was Rejected

When Alternatives Vote over Voters

Marky D. Kondor VII

Abstract

We drastically depart from the standard theory of voting by considering settings where the alternatives rank the voters.

1 Introduction

In traditional voting theory, each voter ranks all the alternatives. Recently, some settings have been considered where the voters and the alternatives coincide. This leaves a much-needed gap that we fill in this paper: what if, instead of the voters ranking the alternatives, the alternatives rank the voters? This has numerous applications, including addressing a common post-election sentiment of many alternatives, "Let's see how *you* like being ranked."

In this paper, we aim to establish the fundamentals of this ambitious novel research agenda of *inverted social choice*. Surprisingly, we find that many traditional results in social choice theory have remarkably similar analogues in the setting of inverted social choice. In fact, we have so far not been able to find any result for which this is not the case. We suspect that there may be a deeper reason for this, and are currently in the process of applying for a multi-million dollar cross-institution grant to investigate this fascinating phenomenon more thoroughly, and hope to establish a new workshop, COSMOC, which will be held in odd years. We expect that the number of submissions to COSMOC will be roughly equal to that of COMSOC. We will issue a call for reviewers shortly; each reviewer will be evaluated by at least three papers.

Acknowledgments

Professors Biggard and Bow have benefited from this work's comments on them; all remaining errors that they make are, of course, their own.

Marky D. Kondor VII
Department of Scientific Computers
University of International Waters
Email: marky@sc.uiw.ude

Author Index

d|u|p

Drawing: Irene Rothe
Textlayout: Vincent Conitzer, Jörg Rothe
Coverdesign: Isabelle Valerie Mehlhorn, Bernd Prümm
Print: Printsupport 4U, *Focus on Books* (NL- Meppel)