Institute for Logic,
Language and Computation

Proceedings of the 1st International Workshop on Computational Social Choice (COMSOC-2006)

Ulle Endriss & Jérôme Lang (eds.)

Universiteit van Amsterdam

Programme Committee

Krzysztof Apt	CWI & Universiteit van Amsterdam
John Bartholdi	Georgia Institute of Technology
Vincent Conitzer	Duke University
Ulle Endriss (*co-chair*)	Universiteit van Amsterdam
Thibault Gajdos	CNRS & Université Paris-Panthéon-Sorbonne
Edith Hemaspaandra	Rochester Institute of Technology
Wiebe van der Hoek	University of Liverpool
Olivier Hudry	Ecole Nationale Supérieure des Télécommunications
Jérôme Lang (*co-chair*)	CNRS & Université Paul Sabatier
Christian List	London School of Economics
Nicolas Maudet	Université Paris-Dauphine
Eric Pacuit	Universiteit van Amsterdam
Marc Pauly	Stanford University
Hans Peters	Universiteit Maastricht
Jörg Rothe	Heinrich-Heine-Universität Düsseldorf

Additional Reviewers

Yann Chevaleyre	Lane Hemaspaandra
Arantza Estévez-Fernández	Christopher Homan

Sponsors

NWO: Netherlands Organisation for Scientific Research
ILLC: Institute for Logic, Language and Computation (Amsterdam)
BRICKS: Basic Research in Informatics for Creating the Knowledge Society
BNVKI: Belgium-Netherlands Association for Artificial Intelligence

Workshop Website

`http://www.illc.uva.nl/~ulle/COMSOC-2006/`

Preface

Computational Social Choice is a new discipline emerging at the interface of social choice theory and computer science. It is concerned with the application of computational techniques to the study of social choice mechanisms, and with the integration of social choice paradigms into computing.

You have in front of you the proceedings of the 1st International Workshop on Computational Social Choice (COMSOC-2006), hosted on 6-8 December 2006 by the Institute for Logic, Language and Computation (ILLC) at the University of Amsterdam. Our aim in organising COMSOC-2006 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 organise 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.

While the positive, and at times ecstatic, reactions by members of the community to the first Call for Papers and to invitations to join the PC or to give an invited talk suggested that we were on to something good, it still took everyone by surprise when we received a total of 48 paper submissions a few months later. This far exceeded all expectations and furthermore the quality of submissions has been truly excellent. Each paper was reviewed by at least two PC members, supported by a number of additional reviewers. We eventually accepted 38 papers out of the 48 submissions for presentation at the workshop. The revised versions of these papers, taking the comments of reviewers into account, are included in this volume. So are the abstracts of the talks to be given by our invited speakers: Steven Brams, Boi Faltings, Noam Nisan, Francesca Rossi, and Harrie de Swart. A quick glance through the table of contents confirms that Computational Social Choice is a broad and interdisciplinary field. Topics covered include, amongst others, complexity-theoretic studies of voting rules; computational barriers to strategic behaviour; resource allocation and fair division; negotiation in multiagent systems; preference elicitation; ranking systems; logics for social choice; computational issues in coalition formation; mechanism design; and the study of social choice phenomena by means of simulation.

The Call for Paper explicitly solicited submissions of both original papers and of papers describing recently published work, so some of the papers have recently appeared also in other publication venues. The copyright of the articles in this volume lies with the individual authors.

We would like to thank all authors for their interesting papers, the workshop participants for attending, and the PC members for their support and advice during the run-up to COMSOC-2006. Both our PC members and the additional reviewers all wrote high-quality review reports, and did so under a lot of time pressure, when the average workload turned out to be a lot more than first anticipated. We would also like to thank the many people who have helped us out with the local organisation of COMSOC-2006, in particular Ingrid van

Loon, Jessica Pogorzelski and Marjan Veldhuisen for their help with many small and not so small details, which included finding a suitable room for a, after all, not that small workshop in the middle of a busy semester.

Finally, we are grateful to the sponsors of COMSOC-2006 for their generous financial support. These are: the Netherlands Organisation for Scientific Research (NWO); the Institute for Logic, Language and Computation (ILLC); the BRICKS (Basic Research in Informatics for Creating the Knowledge Society) project; and the Belgium-Netherlands Association for Artificial Intelligence (BNVKI). We are looking forward to an exciting three days that promise to have long-lasting effects on the field.

Amsterdam & Toulouse U.E. & J.L.
November 2006

Invited Talks and Contributed Papers

Better Ways to Cut a Cake

Steven Brams

Procedures to divide a cake among n people with $n-1$ cuts (the minimum number) are analyzed and compared. For 2 persons, cut-and-choose, while envy-free and efficient, limits the cutter to exactly 50% if he or she is ignorant of the chooser's preferences, whereas the chooser can generally obtain more. By comparison, a new 2-person surplus procedure (SP), which induces the players to be truthful in order to maximize their minimum allocations, leads to a *proportionally equitable* division of the surplus—the part that remains after each player receives 50%–by giving each person exactly the same proportion of the surplus as he or she values it. For $n \geq 3$ persons, a new equitable procedure (EP) yields a *maximally equitable* division of a cake. This division gives all players the highest common value that they can achieve and induces truthfulness, but it may not be envy-free. The applicability of SP and EP to the fair division of a heterogeneous, divisible good, like land, is briefly discussed.

This is joint work with Michael A. Jones and Christian Klamler.

Steven Brams
Wilf Family Department of Politics
New York University
New York, NY 10003-9580, United States
Email: steven.brams@nyu.edu

Budget-Balance in Social Choice

Boi Faltings

Known general mechanisms for incentive-compatible social choice such as the Clarke tax generate budget surplus. While game theory stipulates that such surplus must be wasted, in practice it is usually given to an interested party, thus creating incentives for manipulation. The talk will discuss possibilities for achieving budget-balance in social choice mechanisms.

Boi Faltings
Artificial Intelligence Laboratory
School of Computer and Communication Sciences
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
Email: boi.faltings@epfl.ch

Approximation Mechanisms and Characterization of Implementable Social Choice Rules

Noam Nisan

The emerging field of Algorithmic Mechanism Design studies strategic implementations of social-choice functions that arise in computational settings–most importantly, various resource allocation rules. The clash between computational constraints and incentive constraints is at the heart of this field. This happens whenever one wishes to implement a computationally-hard social choice function (e.g. an allocation rule). In such cases, approximations or heuristics are computationally required, but it is not at all clear whether these can be strategically implemented.

This talk will demonstrate many of the issues involved by looking in depth at a representative problem: multi-unit auctions.

The talk will have the flavor of a survey and is based on my previous joint work with Amir Ronen, Ilya Segal, Ahuva Mu'alem, Ron Lavi, and Shahar Dobzinski.

Noam Nisan
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Givat Ram, Jerusalem 91904, Israel
Email: noam@cs.huji.ac.il

Incomparability and Uncertainty in Preference Aggregation

Francesca Rossi

We consider multi-agent settings where agents' preferences, which can be partially ordered, need to be aggregated. Moreover, such preferences may be incomplete. For example, agents may hide some of their preferences for privacy reasons, or we might be in the process of eliciting the agents' preferences. In the context of partially-ordered preferences, we study properties such as fairness and non-manipulability, and we show that suitable extensions of classical voting theory results continue to hold.

Moreover, we study the computational complexity of the problem of computing possible and necessary winners, that is, those candidates which can be or always are the most preferred among the agents. Possible and necessary winners are useful bounds to the exact set of winners, that can be known only when incompleteness will be resolved. For example, they help guiding preference elicitation in an efficient way. We show that computing possible and necessary winners is in general a difficult problem, and we identify sufficient conditions on the aggregation function that allow us to compute them in polynomial time.

We then consider the complexity of winner determination in a specific preference aggregation rule: sequential majority voting. Here, uncertainty can arise for two reasons: the choice of the agenda or incomplete preferences. We show that computing possible and necessary winners for this rule is easy. However, if we are interested only in balanced agendas, where the number of competitions for the candidates is as balanced as possible, then winner determination is difficult. This means that, by posing this restriction, this rule is difficult to manipulate.

This is joint work with Jérôme Lang, Maria Silvia Pini, K. Brent Venable, and Toby Walsh.

References

[1] Incompleteness and incomparability in preference aggregation, M.S. Pini, F. Rossi, K.B. Venable, T. Walsh, Proc. IJCAI 2007, Hyderabad, India, January 2007.

[2] Winner determination in sequential majority voting, J. Lang, M.S. Pini, F. Rossi, K.B. Venable, T. Walsh, Proc. IJCAI 2007, Hyderabad, India, January 2007.

[3] Strategic voting when aggregating partially ordered preferences, F. Rossi, M.S. Pini, K.B. Venable, T. Walsh, Proc. AAMAS 2006, Hakodate, Japan, May 2006.

[4] Aggregating partially ordered preferences: possibility and impossibility results, M.S. Pini, F. Rossi, K.B. Venable, T. Walsh, Proc. TARK X, Singapore, June 2005, ACM Digital Library.

Francesca Rossi
Department of Pure and Applied Mathematics
University of Padova
35121 Padova, Italy
Email: frossi@math.unipd.it

Social Software for Coalition Formation

Harrie de Swart

This paper concerns an interdisciplinary approach to coalition formation. We apply the MacBeth software, relational algebra, the RelView tool, graph theory, bargaining theory, social choice theory, and consensus reaching to a model of coalition formation.

A feasible government is a pair consisting of a coalition of parties and a policy supported by this coalition. A feasible government is stable if it is not dominated by any other feasible government. Each party evaluates each government with respect to certain criteria. MacBeth helps to quantify the importance of the criteria and the attractiveness and repulsiveness of governments to parties with respect to the given criteria. Feasibility, dominance, and stability are formulated in relation-algebraic terms. The RelView tool is used to compute the dominance relation and the set of all stable governments. In case there is no stable government, *i.e.*, in case the dominance relation is cyclic, we apply graph-theoretical techniques for breaking the cycles. If the solution is not unique, we select the final government by applying bargaining or appropriate social choice rules. We describe how a coalition may form a government by reaching consensus about a policy.

This is joint work Agnieszka Rusinowska, Rudolf Berghammer, Patrik Eklund, Jan-Willem van der Rijt, and Marc Roubens.

Harrie de Swart
Chair of Logic and Linguistic Analysis
Faculty of Philosophy
University of Tilburg
5000 LE Tilburg, The Netherlands
Email: `H.C.M.deSwart@uvt.nl`

Towards a Logic of Social Welfare[1]

Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge

Abstract

We present a formal logic of social welfare functions. The logical language is syntactically simple, but expressive enough to express interesting and complicated properties of social welfare functions involving, e.g., quantification over both preference relations and over individual alternatives, such as Arrow's thorem.

1 Introduction

In the recent years there has been a great deal of interest in the logical aspects of *societies*. For example, Alternating-time Temporal Logic (ATL) [1] and Coalition Logic (CL) [11] can be used to reason about the strategic abilities of individual agents and of coalitions. There is a close connection between these logics and game theory. A related field which, like game theory, also is concerned with social interaction, is social choice theory. A key issue in the latter field is the construction of *social welfare functions, (SWFs)*, mapping individual preferences into "social preferences". Many of the most well known results in social choice theory are impossibility results such as Arrow's theorem [3]: there is no SWF that meets all of a certain number of reasonable conditions. Formal logics related to social choice have focused mostly on the logical representation of preferences when the set of alternatives is large and on the computational properties of computing aggregated preferences for a given representation [7, 8, 9].

In this paper, we present a formal logic which makes it possible to explicitly represent and reason about individual preferences and social preferences. The main differences to the logics mentioned above are as follows. First, the logical language is interpreted directly by social welfare functions and thus that formulae can be read as properties of such functions; second, that preferences are represented in a more abstract way; and, third, that the expressive power is sufficient for interesting problems as discussed below.

Motivations for modeling social choice using logic are manyfold. In particular, logic enables formal *knowledge representation* and *reasoning*. For example, in multiagent systems [13], agents must be able to represent and reason about propositions involving other agents' preferences and preference aggregation. For social choice theory, logic can enable tools for, e.g., mechanically generating proofs, checking the soundness of proofs, mechanically generating possibly in-

[1] An almost identical version of this paper was presented at *the 7th conference on Logic and the Foundations of Game and Decision Theory (LOFT 06)*.

teresting theorems, checkin gproperties of particular social welfare functions, etc.

An example of a property of (some) social welfare functions is so-called *independence of irrelevant alternatives (IIA)*: given two preference profiles and two alternatives, if for each agent the two alternatives have the same order in the two preference profiles, then the two alternatives must have the same order in the two preference relations resulting from applying the SWF to the two preference profiles, respectively. From this example it seems that a formal language about SWFs should be able to express:

- Quantification on several levels: over alternatives; over preference profiles, i.e., over relations over alternatives (second-order quantification); and over agents.
- Properties of preference relations for different agents, and properties of several different preference relations for the same agent in the same formula.
- Comparison of different preference relations.
- The preference relation resulting from applying a SWF to other preference relations.

From these points it seems that such a language would be complex (in particular, they seem to rule out a "standard" propositional modal logic). However, perhaps surprisingly, the language we present in this paper is syntactically and semantically rather simple; and yet the language is, nevertheless, expressive enough to give an elegant and succinct expression of properties such as IIA.

In the next section, we introduce preference relations and social welfare functions. We formally define certain well known potential properties of SWFs, and give a statement of Arrow's theorem. In Section 3 we present the syntax and semantics of our logic, and discuss the complexity of the model checking problem. We show how the mentioned properties can be expressed in the logical language in Section 4. In particular, we show that we can express the statement of Arrow's theorem as a formula – as a result of the theorem, this formula is valid in our logic. In Section 5 we discuss some other valid properties of the logic, and briefly discuss how some of the properties can be expressed in the modal logic *arrow logic* (which originally is about arrows and not about Arrow!). We conclude in Section 6.

2 Social Welfare Functions

Social welfare functions (SWFs) are usually defined in terms of ordinal preference structures, rather than cardinal structures such as utility functions. An

SWF takes as input a preference relation, a binary relation over some set of alternatives, for each agent, and outputs another preference relation representing the aggregated preferences.

The most well known result about SWFs is Arrow's theorem [3]. Many variants of the theorem appears in the literature, differing in assumptions about the preference relations. In this paper, we take the assumption that all preference relations are linear orders, i.e., that neither agents nor the aggregated preference can be indifferent between distinct alternatives. This gives one of the simplest formulations of Arrow's theorem (Theorem 1 below). Cf., e.g., [4] for a discussion and more general formulations.

Formally, let A be a set of *alternatives*. We henceforth implicitly assume that there is always at least two alternatives. A *preference relation* (over A) is, here, a total (linear) order on A, i.e., a relation R over A which is antisymmetric (i.e., $(a,b) \in R$ and $(b,a) \in R$ implies that $a = b$), transitive (i.e., $(a,b) \in R$ and $(b,c) \in R$ implies that $(a,c) \in R$), and total (i.e., either $(a,b) \in R$ or $(b,a) \in R$ for every pair of alternatives a and b). We sometimes use the infix notation aRb for $(a,b) \in R$. The set of preference relations over alternatives A is denoted $L(A)$. Alternatively, we can view $L(A)$ as the set of all permutations of A. Thus, we shall sometimes use a permutation of A to denote a member of $L(A)$. For example, when $A = \{a,b,c\}$, we will sometimes use the expression acb to denote the relation $\{(a,c),(a,b),(c,b),(a,a),(b,b),(c,c)\}$. aRb means that b is preferred over a if a and b are different. R^s denotes the non-reflexive version of R, i.e., $R^s = R \setminus \{(a,a) : a \in A\}$. aR^sb means that b is preferred over a and that $a \neq b$.

Let n be a number of *agents*; we write Σ for the set $\{1,\ldots,n\}$. A *preference profile* for Σ over alternatives A is a tuple $(R_1,\ldots,R_n) \in L(A)^n$.

A *social welfare function* (SWF) is a function

$$F : L(A)^n \to L(A)$$

mapping each preference profile to an aggregated preference relation. The class of all SWFs over alternatives A is denoted $\mathcal{F}(A)$.

Commonly discussed properties a SWF F can have include:

PO $\forall_{(R_1,\ldots,R_n)\in L(A)^n}\forall_{a\in A}\forall_{b\in A}((\forall_{i\in\Sigma}aR_i^sb) \Rightarrow aF(R_1,\ldots,R_n)^sb)$ (pareto optimality)

ND $\neg\exists_{i\in\Sigma}\forall_{(R_1,\ldots,R_n)\in L(A)^n}F(R_1,\ldots,R_n) = R_i$ (non-dictatorship)

IIA $\forall_{(R_1,\ldots,R_n)\in L(A)^n}\forall_{(S_1,\ldots,S_n)\in L(A)^n}\forall_{a\in A}\forall_{b\in A}((\forall_{i\in\Sigma}(aR_ib \Leftrightarrow aS_ib)) \Rightarrow (aF(R_1,\ldots,R_n)b \Leftrightarrow aF(S_1,\ldots,S_n)b))$ (independence of irrelevant alternatives)

Arrow's theorem says that the three properties above are inconsistent if there are more than two alternatives.

Theorem 1 (Arrow)**.** *If there are more than two alternatives, no SWF has all the properties **PO**, **ND** and **IIA**.*

We now introduce a formal language in which properties such the above can be expressed.

3 The Logic

We now present a logical language and its interpretation in SWFs. The language is syntactically simple, but the representation of preferences is unconventional and we will therefore discuss the main points before giving formal definitions.

An example of a formula is

$$\Diamond \boxdot (r_1 \leftrightarrow r) \tag{1}$$

A formula denotes a property of a SWF. The formula (1) says that there exist ($\Diamond$) preferences for the agents such that for all ($\boxdot$) pairs of alternatives, agent 1 (r_1) and the aggregated preferences (r) agree on the relative ranking of the two alternatives (i.e., on which of the two is better than the other).

While a formula is interpreted in a SWF, a subformula may be interpreted in additional structures depending on which quantifiers ($\Diamond$,$\Box$,$\diamondsuit$,$\boxdot$) the subformula is in the scope of. Here is a detailed description of the intended meaning of the parts of the formula (1):

r_1 : A statement about the combination of a SWF F, a preference profile $(R_1, \ldots, R_n)$ and a pair of alternatives (a, b). It says that according to the preference profile, agent 1 prefers b (the last element in the pair) over a (the first element in the pair).

r : A statement about the combination of a SWF F, a preference profile $(R_1, \ldots, R_n)$ and a pair of alternatives (a, b). It says that according to the preference relation resulting from applying the SWF to the preference profile, b is preferred over a.

$\boxdot(r_1 \leftrightarrow r)$: A statement about the combination of a SWF F and a preference profile $(R_1, \ldots, R_n)$. It says that for every pairs of alternatives, $(r_1 \leftrightarrow r)$ holds wrt. the SWF, preference profile, and pair of alternatives.

$\Diamond \boxdot (r_1 \leftrightarrow r)$: A statement about a SWF F. It says that there exists a preference profile such that for all pairs (a, b) of alternatives, b is preferred over a in the aggregation (by the SWF) of the preference profile if and only if agent 1 prefers b over a.

3.1 Syntax

The logical language is parameterised by the number of agents n, in addition to a stock of symbols $\Pi = \{r, s, \ldots\}$. A symbol $r \in \Pi$ will be used to refer to a preference profile $R \in L(A)^n$. In the example above, formula (1), we only used one symbol r, but as we shall see it is useful to be able to reason about several different preference profiles at the same time. Formally, we define three languages: $\mathcal{L}$ expresses properties of SWFs and is the language we are ultimately interested in. $\mathcal{L}$ is defined in terms of $\mathcal{L}_2$. $\mathcal{L}_2$ expresses properties of preference profiles (one for each member of Π) relative to a SWF, and is again

defined in terms of $\mathcal{L}_3$. $\mathcal{L}_3$ expresses properties of a pair $(a, b) \in A^2$ relative to a SWF and some preference profiles.

$\mathcal{L}$: $\phi ::= \Box\psi \mid \neg\phi \mid \phi_1 \wedge \phi_2$

$\mathcal{L}_2$: $\psi ::= \boxdot\gamma \mid \neg\psi \mid \psi_1 \wedge \psi_2$

$\mathcal{L}_3$: $\gamma ::= r_i \mid r \mid \neg\gamma \mid \gamma_1 \wedge \gamma_2$ where $i \in \Sigma$ and $r \in \Pi$

We use the duals: $\Diamond\psi \equiv \neg\Box\neg\psi$ and $\diamondsuit\gamma \equiv \neg \boxdot \neg\gamma$, in addition to the usual derived propositional connectives.

Note that we do not allow arbitrary nesting of the quantifiers.

3.2 Semantics

A *profile function*

$$\delta : \Pi \to L(A)^n$$

associates a preference profile $\delta(r) = (R_1, \ldots, R_n)$ with each symbol $r \in \Pi$. If $\delta(r) = (R_1, \ldots, R_n)$, we write $\delta_i(r)$ for R_i. The set of all profile functions over A and Π is denoted $\Delta(A, \Pi)$ (or just Δ). $\mathcal{L}$ is interpreted in an SWF $F \in \mathcal{F}(A)$ as follows:

$$\begin{array}{lcl} (A,F) \models \Box\psi & \Leftrightarrow & \forall_{\delta\in\Delta}(A,F,\delta) \models \psi \\ (A,F) \models \neg\phi & \Leftrightarrow & (A,F) \not\models \phi \\ (A,F) \models \phi_1 \wedge \phi_2 & \Leftrightarrow & (A,F) \models \phi_1 \text{ and } (A,F) \models \phi_2 \end{array}$$

$\mathcal{L}_2$ is interpreted in an SWF F and a profile function δ as follows:

$$\begin{array}{lcl} (A,F,\delta) \models \boxdot\gamma & \Leftrightarrow & (\forall_{(a,b)\in A\times A} a \neq b \Rightarrow (A,F,\delta,(a,b)) \models \gamma) \\ (A,F,\delta) \models \neg\psi & \Leftrightarrow & (A,F,\delta) \not\models \psi \\ (A,F,\delta) \models \psi_1 \wedge \psi_2 & \Leftrightarrow & (A,F,\delta) \models \psi_1 \text{ and } (A,F,\delta) \models \psi_2 \end{array}$$

$\mathcal{L}_3$ is interpreted in an SWF F, a profile function δ and a pair of distinct alternatives (a, b) as follows:

$$\begin{array}{lcl} (A,F,\delta,(a,b)) \models r_i & \Leftrightarrow & (a,b) \in \delta_i(r) \\ (A,F,\delta,(a,b)) \models r & \Leftrightarrow & (a,b) \in F(\delta(r)) \\ (A,F,\delta,(a,b)) \models \neg\gamma & \Leftrightarrow & (A,F,\delta,(a,b)) \not\models \gamma \\ (A,F,\delta,(a,b)) \models \gamma_1 \wedge \gamma_2 & \Leftrightarrow & (A,F,\delta,(a,b)) \models \gamma_1 \text{ and } (A,F,\delta,(a,b)) \models \gamma_2 \end{array}$$

Given a set of alternatives A, as formula is *valid on* A if $A, F \models \phi$ for all $F \in \mathcal{F}(A)$. A formula ϕ is *valid*, written $\models \phi$, if $A \models \phi$ for all A.

3.3 Model Checking

Most implemented systems for reasoning about cooperation are based on *model checking* [6, 2]. Roughly speaking, the model checking problem for a given logic is as follows: Given a formula ϕ of the logic, and a model/interpretation M for the logic, is it the case that $M \models \phi$? For our logic, we have three model checking problems, for the languages $\mathcal{L}$, $\mathcal{L}_2$, and $\mathcal{L}_3$ respectively. For example, the $\mathcal{L}$ model checking problem is as follows:

Given a set A of alternatives, a social welfare function $F \in \mathcal{F}(A)$, and a formula ϕ of $\mathcal{L}$, is it the case that $(A, F) \models \phi$?

The model checking problems for $\mathcal{L}_2$ and $\mathcal{L}_3$ may be derived similarly. The model checking problem for $\mathcal{L}$ can be understood as asking whether the property of social welfare functions expressed by the formula ϕ is true of the given social welfare function F. For example, given the formula PO discussed in the next section, checking whether $(A, F) \models PO$, is exactly the problem of checking whether F has the Pareto Optimality property.

The *complexity* of the model checking problem for $\mathcal{L}$ depends upon the representation chosen for the function F. The simplest representation will be an *extensive* one, where the function is enumerated as the set of all pairs of the form (i, o), where i is an input to F and $o = F(i)$ is the corresponding output. The obvious "catch" is that this representation of F must list the value of F for every input: and there will be exponentially many (in the number of alternatives) possible inputs. So, an alternative is to assume a *succinct* representation for F. We consider one such alternative, where F is represented as a polynomially bounded deterministic two-tape Turing machine. Roughly, this can be understood as representing F as a program computing the social welfare function which is guaranteed to terminate with an output in polynomial time. (Of course, it may be the case that there are F's which cannot be so represented.)

Now, it is easy to see that, assuming the extensive representation, the model checking problems for $\mathcal{L}$, $\mathcal{L}_2$, and $\mathcal{L}_3$ may be solved in deterministic polynomial time. However, since the inputs are exponentially large, this result is perhaps misleading. We can show the following.

Proposition 1. *For the succinct representation of SWFs, the model checking problem for $\mathcal{L}$ is NP-hard even for formulae of the form $\Box\psi$.*

Proof. We reduce SAT, the problem of determining whether a given formula ξ of propositional logic over variables $x_1, \ldots, x_k$ is satisfied by some assignment of truth/falsity to its Boolean variables $x_1, \ldots, x_k$ [10]. Given an instance $\xi(x_1, \ldots, x_k)$ of SAT, we create an instance of model checking for $\mathcal{L}$ as follows. First, we create just two alternatives, $A = \{a, b\}$; for each Boolean variable x_i we create an agent, and define an $\mathcal{L}_2$ variable r_i . We then define F so that it produces the ranking (a, b). Next, we define $\xi^{\#}$ to be the formula obtained from ξ by systematically replacing the variable x_i by r_i . We then define the formula ζ that is input to the $\mathcal{L}$ model checking problem to be:

$$\zeta = \Diamond\Diamond\xi^{\#}.$$

That the formula ζ is true given F and A as defined iff ξ is satisfiable is now straightforward. □

Notice that for the succinct representation, the model checking problems for $\mathcal{L}_2$ and $\mathcal{L}_3$ are easily seen to be solvable in deterministic polynomial time. The general model checking problem for $\mathcal{L}$ for succinct representations is also easily

seen to be in Δ_2^p (the class of problems solvable in polynomial time assuming an oracle for problems in NP).

4 Examples

The proofs of the following propositions are straightforward.

Pareto optimality can be expressed as follows:

$$PO = \Box \boxdot ((r_1 \wedge \cdots \wedge r_n) \rightarrow r) \tag{2}$$

Proposition 2. *Let $F \in \mathcal{F}(A)$. $(A, F) \models PO$ iff F has the property **PO**.*

Non-dictatorship can be expressed as follows:

$$ND = \bigwedge_{i \in \Sigma} \Diamond \circledcirc \neg (r \leftrightarrow r_i) \tag{3}$$

Proposition 3. *Let $F \in \mathcal{F}(A)$. $(A, F) \models ND$ iff F has the property **ND**.*

Independence of irrelevant alternatives can be expressed as follows:

$$IIA = \Box \boxdot ((r_1 \leftrightarrow s_1 \wedge \cdots \wedge r_n \leftrightarrow s_n) \rightarrow (r \leftrightarrow s)) \tag{4}$$

Proposition 4. *Let $F \in \mathcal{F}(A)$. $(A, F) \models IIA$ iff F has the property **IIA**.*

4.1 Cardinality of Alternatives

The properties expressed above are properties of social welfare functions. We turn to look now at which properties of the set of *alternatives* A we can express. Note that we cannot refer to particular alternatives directly in the logical language. Properties involving *cardinality* is often of interest, for example in Arrow's theorem. Let:

$$MT2 = \Diamond (\circledcirc (r_1 \wedge s_1) \wedge \circledcirc (r_1 \wedge \neg s_1))$$

Proposition 5. *Let $F \in \mathcal{F}(A)$. $|A| > 2$ iff $(A, F) \models MT2$.*

Proof. For the direction to the left, let $(A, F) \models MT2$. Thus, there is a δ such that there exists $(a^1, b^1), (a^2, b^2) \in A \times A$, where $a^1 \neq b^1$, and $a^2 \neq b^2$, such that (i) $(a^1, b^1) \in \delta_1(r)$, (ii) $(a^1, b^1) \in \delta_1(s)$, (iii) $(a^2, b^2) \in \delta_1(r)$ and (iv) $(a^2, b^2) \notin \delta_1(s)$. From (ii) and (iv) we get that $(a^1, b^1) \neq (a^2, b^2)$, and from that and (i) and (iii) it follows that $\delta_1(r)$ contains two different pairs each having two different elements. But that is not possible if $|A| = 2$, because if $A = \{a, b\}$ then $L(A) = \{ab, ba\} = \{ \{(a, b), (a, a), (b, b)\}, \{(b, a), (a, a), (b, b) \}$, so it cannot be that $\delta_1(r) \in L(A)$.

For the direction to the right, let $|A| > 2$; let a, b, c be three different elements of A. Let $\delta_1(r) = abc$ and $\delta_1(s) = acb$. Now, for any F, $(A, F, \delta, (a, b)) \models r_1 \wedge s_1$ and $(A, F, \delta, (b, c)) \models r_1 \wedge \neg s_1$. Thus, $(A, F) \models MT2$, for any F. □

Other interesting properties hold when the cardinality of the set of alternatives is finite and fixed:

Example 1. *Consider the case when $\Pi = \{r\}$, there are two agents, and three alternatives. Then the following holds (for every A with $|A| = 3$):*

$$A \models \Box(\diamond(r \wedge r_1 \wedge r_2) \wedge \diamond(r \wedge \neg r_1 \wedge r_2) \wedge \diamond(r \wedge r_1 \wedge \neg r_2) \rightarrow \boxdot(r \rightarrow (r_1 \vee r_2))$$

This validity says that, for any SWF and any preferences, if there exist pairs of alternatives on which (i) both agents agree with the SWF, (ii) only agent 1 agrees with the SWF and (iii) only agent 2 agrees with the SWF, then for every pair at least one of the agents must agree with the SWF.

Here is a justification. There are eight "descriptors" of the form $r_1 \wedge r_2 \wedge r$, $\neg r_1 \wedge r_2 \wedge r$, etc., i.e. conjunctions of literals completely describing preferences over a pair. But, given a SWF F and a profile function δ, a $\mathcal{L}_3$ formula on the form $\diamond d$ where d is a descriptor holds for exactly six of the eight descriptors. To see this, observe that with three alternatives, there are only six distinct pairs, and two different descriptors cannot be true in the same pair. Furthermore, these six descriptors consists of three pairs of complementary descriptors, where the complement of a descriptor is obtained by changing the sign of each literal: if d is true in a pair (a, b), then the complement of d is true in the pair (b, a). So $\diamond d$ can be true in a given SWF and profile function for only three different non-complimentary descriptors d at the same time. In the example formula above, the three descriptors in the antecedent of the implications are non-complimentary, and the fourth descriptor in the consequent is non-complimentary to these three as well, so the latter cannot be true at the same time as all the three former.

4.2 Arrow's Theorem

We now have everything we need to express Arrow's statement as a formula. It follows from his theorem that the formula is valid.

Theorem 2.

$$\models MT2 \rightarrow \neg(PO \wedge ND \wedge IIA)$$

Proof. Let A be a set of alternatives, $F \in \mathcal{F}(A)$, and $(A, F) \models MT2$. By Proposition 5, A has more than two alternatives. By Arrow's theorem, F cannot have all the properties **PO**, **ND** and **IIA**. By Propositions 2, 3 and 4, $(A, F) \models \neg PO \vee \neg ND \vee \neg IIA$. □

5 Logical Properties

We here take a closer look at additional universal properties of SWFs expressible in the logic: which $\mathcal{L}$ formulae are valid?

First – trivially – we have that

$$\models \phi \qquad \phi \text{ instance of prop. tautology} \qquad (Prop_1)$$
$$\models \Box\psi \qquad \psi \text{ instance of prop. tautology} \qquad (Prop_2)$$
$$\models \Box\,\Box\,\gamma \qquad \gamma \text{ instance of prop. tautology} \qquad (Prop_3)$$

It is also easy to see that we have the K axiom, on both "level" $\mathcal{L}$ and $\mathcal{L}_2$:

$$\models \Box(\psi_1 \to \psi_2) \to (\Box\psi_1 \to \Box\psi_2) \qquad (K_1)$$
$$\models \Box\,(\boxdot(\psi_1 \to \psi_2) \to (\boxdot\psi_1 \to \boxdot\psi_2)) \qquad (K_2)$$

However, the remainding principle of normal modal logics (cf., e.g., [5]), *uniform substitution*, does *not* hold for our logic. A counter example is the fact that the following is valid:

$$\Box\Diamond r \tag{5}$$

– no matter what preferences the agents have, the SWF will always rank some alternative over another – while this is not valid:

$$\Box\Diamond(r \wedge r_1) \tag{6}$$

– the SWF will not necessarily rank any two alternatives in the same order as agent 1.

The formuale in (5) and (6) have the same pattern of quantifiers ($\Box\Diamond$), and a natural question is then for which γ the formula $\Box\Diamond\gamma$ is valid. Theorem 3 below partly answers that question (both claims above about validity and non-validity of (5) and (6), respectivelly, thus follow from that theorem). First some definitions and an intermediate result.

We shall sometimes treat $\mathcal{L}_3$ as the language of propositional logic, with atomic propositions

$$Atoms(\Pi, \Sigma) = \{r_i, r : r \in \Pi, i \in \Sigma\}$$

(or just *Atoms* when Π and Σ are clear from context). A propositional valuation will simply be represented as a subset V of *Atoms*. We reuse the $\models$ symbol (no confusion can occur), and write $V \models \gamma$ when V is a valuation satisfying (in the classical truth-functional sense) a formula $\gamma \in \mathcal{L}_3$, as well as $\models \gamma$ when $V \models \gamma$ for all $V \subseteq Atoms$. We use $Lit(\Pi, \Sigma)$ (or just *Lit*) to denote the set of literals: $Lit(\Pi, \Sigma) = Atoms(\Pi, \Sigma) \cup \{\neg q : q \in Atoms(\Pi, \Sigma)\}$. When $\gamma \in \mathcal{L}_3$, we use $\overline{\gamma}$ to denote the result of negating every occurrence of an atom in γ.[2] Formally: $\overline{q} = \neg q$ when $q \in Atoms$; $\overline{\neg\gamma} = \neg\overline{\gamma}$; $\overline{\gamma_1 \wedge \gamma_2} = \overline{\gamma_1} \wedge \overline{\gamma_2}$.

The proof of the following Lemma is straightforward.

Lemma 1. *For any A, F, δ, any pair $a, b \in A$, $a \neq b$, and any $\mathcal{L}_3$ formula γ:*

$$(A, F, \delta, (a, b)) \models \overline{\gamma} \Leftrightarrow (A, F, \delta, (b, a)) \models \gamma$$

[2]The "overline" notation is sometimes used to denote negation, note that our use is different.

Theorem 3. *For any $k \geq 1$, and any $\gamma_1, \ldots, \gamma_k \in \mathcal{L}_3$:*

$$\models \Box(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k) \Leftrightarrow \models \gamma_1 \vee \overline{\gamma_1} \vee \cdots \vee \gamma_k \vee \overline{\gamma_k}$$

Proof. Let $\gamma_1, \ldots, \gamma_k \in \mathcal{L}_3$.

For the direction to the left, let A be a set of alternatives, F an SWF, and $\delta \in \Delta$. Note that $\overline{\gamma_1 \vee \cdots \vee \gamma_k} = \overline{\gamma_1} \vee \cdots \vee \overline{\gamma_k}$. Let $a, b \in A$, $a \neq b$. $(A, F, \delta, (a, b))$ can be seen as a valuation (over *Atoms*), so by the right hand side, $(A, F, \delta, (a, b)) \models (\gamma_1 \vee \cdots \vee \gamma_k) \vee (\overline{\gamma_1 \vee \cdots \vee \gamma_k})$, so either $(A, F, \delta, (a, b)) \models \gamma_1 \vee \cdots \vee \gamma_k$ or $(A, F, \delta, (a, b)) \models \overline{\gamma_1 \vee \cdots \vee \gamma_k}$ (or both). By Lemma 1, either $(A, F, \delta, (a, b)) \models \gamma_1 \vee \cdots \vee \gamma_k$ or $(A, F, \delta, (b, a)) \models \gamma_1 \vee \cdots \vee \gamma_k$ (or both). Thus, there is a j such that either $(A, F, \delta, (a, b)) \models \gamma_j$ or $(A, F, \delta, (b, a)) \models \gamma_j$. It follows that $(A, F, \delta) \models \Diamond\gamma_j$, and thus that $(A, F, \delta) \models \Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k$. Since A, F, δ were arbitrary, we have that $\models \Box(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k)$.

For the direction to the right, we show the contrapositive. Assume that there is a propositional valuation V such that $V \not\models \gamma_1 \vee \overline{\gamma_1} \vee \cdots \vee \gamma_k \vee \overline{\gamma_k}$. Then $V \models \neg(\gamma_1 \vee \cdots \vee \gamma_k)$ and $V \models \neg(\overline{\gamma_1} \vee \cdots \vee \overline{\gamma_k})$. The latter is equivalent to $V \models \overline{\neg(\gamma_1 \vee \cdots \vee \gamma_k)}$. Now, let $A = \{a, b\}$ $(a \neq b)$, and let F and δ be defined as follows:

$$\delta_i(r) = \begin{cases} ab & r_i \in V \\ ba & \text{otherwise} \end{cases} \qquad F(\delta(r)) = \begin{cases} ab & r \in V \\ ba & \text{otherwise} \end{cases}$$

It can easily be seen, by induction over the formula, that V and (a, b) agrees on every $\mathcal{L}_3$ formula, i.e., that for every $\gamma \in \mathcal{L}_3$

$$V \models \gamma \Leftrightarrow (A, F, \delta, (a, b)) \models \gamma \tag{7}$$

Thus, we have that $(A, F, \delta, (a, b)) \models \neg(\gamma_1 \vee \cdots \vee \gamma_k)$. But since $V \models \overline{\neg(\gamma_1 \vee \cdots \vee \gamma_k)}$, we also get $(A, F, \delta, (a, b)) \models \overline{\neg(\gamma_1 \vee \cdots \vee \gamma_k)}$ from (7), and thus that $(A, F, \delta, (b, a)) \models \neg(\gamma_1 \vee \cdots \vee \gamma_k)$ from Lemma 1. Since (a, b) and (b, a) are the only pairs of distinct elements from A, we have that $(A, F, \delta) \models \boxdot\neg(\gamma_1 \vee \cdots \vee \gamma_k)$. From K_2 and $Prop_2$ and $Prop_3$ we get that $(A, F, \delta) \models \boxdot\neg\gamma_1 \wedge \cdots \wedge \boxdot\neg\gamma_k$. This is, again by propositional reasoning, the same as $(A, F, \delta) \models \neg(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k)$. Thus, we have established that $\not\models \Box(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k)$. □

Some applications showing both directions of Theorem 3:

$\models \Box\Diamond q$ **for any** $q \in Lit$**:** Both the individual agents and the SWF will always rank some alternative above another and, conversely, some alternative below some other. (5) above is an instance. Justification: if $q \in Lit$, then $\overline{q} = \neg q$, so $\models q \vee \overline{q}$ holds.

$\not\models \Box\Diamond(q_1 \wedge q_2)$ **when** $q_1 \neq q_2 \in Lit$**:** we are not guaranteed that there is a pair of alternatives ranked in the same order by two agents and/or the SWF. (6) above is an instance. Justification: if $q_1 \neq q_2 \in Lit$, then $\overline{q_1 \wedge q_2} = \neg q_1 \wedge \neg q_2$. But it is not the case that $(q_1 \wedge q_2) \vee (\neg q_1 \wedge \neg q_2)$ is a propositional tautology.

$\models \Box(\boxdot(r_1 \vee r_2) \rightarrow \diamondsuit(r_1 \wedge \neg r_2))$**:** if, given preferences of agents and a SWF, for any two alternatives it is always the case that either agent 1 or agent 2 prefers the second alternative over the first, then there must exist a pair of alternatives for which the two agents disagree. Justification: the formula in question is equivalent to $\Box(\diamondsuit\gamma_1 \vee \diamondsuit\gamma_2)$, where $\gamma_1 = \neg r_1 \wedge \neg r_2$ and $\gamma_2 = r_1 \wedge \neg r_2$. $\overline{\gamma_1} = \neg\neg r_1 \wedge \neg\neg r_2$ and $\overline{\gamma_2} = \neg r_1 \wedge \neg\neg r_2$, so $\gamma_1 \vee \gamma_2 \vee \overline{\gamma_1} \vee \overline{\gamma_2}$ is a propositional tautology.

The following theorem characterises all valid formulae of the form $\Box\boxdot\gamma$: γ is a propositional tautology. The proof is straightforward.

Theorem 4.

$$\models \Box\boxdot\gamma \Leftrightarrow \models \gamma$$

Properties involving other combinations of quantifiers include:

$\models \diamondsuit\diamondsuit(r_1 \wedge r_2)$**:** There exist preference relations such that agents 1 and 2 agree on some pair of alternatives.

$\not\models \diamondsuit\diamondsuit(r_1 \wedge r)$**:** There does not necessarily exist preference relations such that agent 1 and the SWF agree on some pair of alternatives.

$\models \diamondsuit\boxdot(r_1 \leftrightarrow r_2)$**:** There exist preference relations such that agents 1 and 2 always agree.

$\not\models \diamondsuit\boxdot(r_1 \leftrightarrow r)$**:** There does not necessarily exist preference relations such that agent 1 and the SWF always agree.

5.1 Arrow Logic for Arrow's logic

The modal logic *arrow logic* is designed to reason about any object that can be graphically represented as an arrow [12]. Arrows typically represent a transition triggered by the execution of an action or a computer program, or even the dynamic meaning of a discourse, which explains the popularity of arrow logic among computer scientists, philosophers, and linguists. However, arrows can also be thought of as representing a *preference*, which justifies using arrow logic for our study as well. In this section, we only describe how the language and semantics of arrow logic can be used to represent properties of language $\mathcal{L}_3$: all definitions and notation used in this section are taken from [12].

An *arrow frame* is a tuple $\mathcal{F} = \langle W, \mathcal{R}\rangle$ where W, the universe of $\mathcal{F}$, is a set of *arrows*. Sometimes, it is convenient to think about an arrow a as having as start a_0 and end a_1. Moreover, $\mathcal{R}$ is a set of relations on W, which we will discuss shortly. Given a set of atomic propositions P denoting basic properties, in line with standard modal logic, we can then base a model $\mathcal{M} = \langle\mathcal{F}, V\rangle$ on a frame $\mathcal{F}$ by adding a valuation function $V : P \rightarrow 2^W$, with the meaning that $V(p)$ collects those arrows that satisfy property p. For our purposes, we will take $P = Atoms$, representing the agents' preferences r_i and the collective preference

r, where $\mathcal{M}, a \models r_i$ is meant to mean that according to agent i, alternative a_1 is preferred over a_0 . And similarly $\mathcal{M}, a \models r$ denotes that the welfare function has decided upon judging a_1 better than a_0.

In "basic" arrow logic, there are three relations in $\mathcal{R}$. We follow the notation of [12] and denote them by $C \subseteq W \times W \times W$, and $R \subseteq W \times W$ and $I \subseteq W$, respectively. For three arrows a, b and c, when $Cabc$, we say that a is the composition of b and c. Putting it a bit more formal: $Cabc$ iff $a_0 = b_0$, $b_1 = c_0$ & $c_1 = a_1$. The relation R holds between a and b if b is the inverse of a: Rab iff $a_0 = b_1$ & $b_0 = a_1$. Finally, Ia denotes that a is a reflexive arrow: Ia iff $a_0 = a_1$.

Naturally, in the language for basic arrow logic, we have an operator for each of these relations:

$$\varphi := p \mid \delta \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \circ \varphi \mid \otimes\varphi$$

We now immediately give the truth definition of a formula in an arrow:

$$\begin{array}{lll}
\mathcal{M}, a \models p & \text{iff} & a \in V(p) \\
\mathcal{M}, a \models \delta & \text{iff} & Ia \\
\mathcal{M}, a \models \neg\varphi & \text{iff} & not\mathcal{M}, a \models \varphi \\
\mathcal{M}, a \models \varphi \vee \psi & \text{iff} & \mathcal{M}, a \models \varphi \text{ or } \mathcal{M}, a \models \psi \\
\mathcal{M}, a \models \varphi \circ \psi & \text{iff} & \text{for some } b, c\ (Cabc \& \mathcal{M}, b \models \varphi \& \mathcal{M}, c \models \psi) \\
\mathcal{M}, a \models \otimes\varphi & \text{iff} & \text{for some } b\ (Rab \& \mathcal{M}, b \models \varphi)
\end{array}$$

Recall that $P = \{r_1, r_2, \ldots, r_n, r, \ldots\}$, and that $\mathcal{M}, a \models p$ means that according to p, alternative a_1 is better than a_0, where p either refers to one of the agents, or to the aglomorated result.

Properties of Preferences It appears that most properties we used for preferences have an straightforward translation in arrow logic. We list the following:

1. *transitivity*. This property is expressed by $(p \circ p) \rightarrow p$
2. *asymmetry*. This is $p \rightarrow \otimes\neg p$
3. *linearity*. This becomes $p \vee \otimes p$.
4. *irreflexivity*. This is $\neg\delta$
5. *pareto optimality*. $(\bigwedge r_i \leq nr_i) \rightarrow r$
6. *at most* $n + 1$ *alternatives*. This is $\neg \underbrace{(\top \circ (\top \circ (\cdots \circ \top \cdots)))}_{n \times \top}$

Arrow logics are ususally proven complete wrt. an *algebra*. This would mean, in our context, that it might be possible to use algebras as the underlying structures to represent individual and collective preferences. Then, δ is used to take us from one algebra to another, and F determines the collective preference, in each of the algebras.

6 Conclusions

We have presented a logic of social welfare functions, which is syntactically simple but which can express interesting and complicated properties, involving quantification on several levels, such as Arrow's theorem.

In Section 5 we discussed in depth several properties of the logic. These seem to be a good starting point for a complete axiomatisation of the logic, which remains to be found. Also of importance is to investigate the complexity of the satisfiability problem. Further possibilities for future work include the expression of additional results from social choice theory in general, and in particular relaxing the assumptions about linear orders for the preference relations and the expression of more general variants of Arrow's theorem.

It is interesting to observe that the logic can also be easily used to reason about *judgment aggregation*, i.e., about *judgment aggregation rules* which aggregate consistent sets of propositional formulae, each representing the judgments of an individual agent, into a single consistent set of formulae representing the collective judgments. We are currently working on this interpretation, which we feel can help shed light on the relationship between preference aggregation and judgment aggregation by allowing us to compare the logical principles of each.

The relationship between our logic and arrow logic could also be investigated further.

Acknowledgements The research reported in this paper was carried out when the first author was visiting the Department of Computer Science, University of Liverpool. The first author's work was funded by grant 166525/V30 from the Norwegian Research Council.

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 100–109, Florida, October 1997.

[2] R. Alur, T. A. Henzinger, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Taşiran. Mocha: Modularity in model checking. In *CAV 1998: Tenth International Conference on Computer-aided Verification, (LNCS Volume 1427)*, pages 521–525. Springer-Verlag, 1998.

[3] K. J. Arrow. *Social Choice and Individual Values*. Wiley, 1951.

[4] K. J. Arrow, Amartya K. Sen, and Kotaro Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1. North-Holland, 2002.

[5] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge University Press, 2001.

[6] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.

[7] Celine Lafage and Jérôme Lang. Logical representation of preferences for group decision making. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *Proceedings of the Conference on Principles of Knowledge Representation and Reasoning (KR-00)*, pages 457–470, S.F., April 11–15 2000. Morgan Kaufman Publishers.

[8] Jérôme Lang. From preference representation to combinatorial vote. In Dieter Fensel, Fausto Giunchiglia, Deborah L. McGuinness, and Mary-Anne Williams, editors, *Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002*, pages 277–290. Morgan Kaufmann, 2002.

[9] Jérôme Lang. Logical preference representation and combinatorial vote. *Ann. Math. Artif. Intell*, 42(1-3):37–71, 2004.

[10] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[11] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[12] Y. Venema. A crash course in arrow logic. In M. Marx, M. Masuch, and L. Pólos, editors, *Arrow Logic and Multi-Modal Logic*, pages 3–34. CSLI Publications, Stanford, 1996.

Thomas Ågotnes
Department of Computer Engineering, Bergen University College
P.O.Box 7030, N-5020 Bergen, Norway
Email: tag@hib.no

Wiebe van der Hoek
Department of Computer Science, University of Liverpool
Ashton Building, Ashton Street, Liverpool L69 3BX, UK
Email: wiebe@csc.liv.ac.uk

Michael Wooldridge
Department of Computer Science, University of Liverpool
Ashton Building, Ashton Street, Liverpool L69 3BX, UK
Email: mjw@csc.liv.ac.uk

A Generic Approach to Coalition Formation

Krzysztof R. Apt and Andreas Witzel

Abstract

We propose an abstract approach to coalition formation by focusing on partial preference relations between partitions of a grand coalition. Coalition formation is modelled by means of simple merge and split rules that transform partitions. We identify conditions under which every iteration of these rules yields a unique partition. The main conceptual tool is the notion of a stable partition. The results naturally apply to coalitional TU-games and to some classes of hedonic games.

1 Introduction

1.1 Background

Coalition formation has been a research topic of continuing interest in the area of coalitional games. It has been analyzed from several points of view, starting with [2], where the static situation of cooperative games in the presence of a given coalition structure (i.e., a partition) was considered. Early research on the subject is discussed in [10].

More recently, the problem of formation of stable coalition structures was considered in [15] in the presence of externalities and in [13] in the presence of binding agreements. In both papers two-stage games are analyzed. In the first stage coalitions form and in the second stage the players engage in a non-cooperative game given the emerged coalition structure. In this context the question of stability of the coalition structure is then analyzed.

Much research on stable coalition structures focused on hedonic games. These are games in which the payoff of a player depends exclusively on the members of the coalition he belongs to. In other words, a payoff of a player is a preference relation on the sets of players that include him. [5] considered four forms of stability in such games: core, Nash, individual and contractually individual stability. Each alternative captures the idea that no player, respectively, no group of players has an incentive to change the existing coalition structure. The problem of existence of (core, Nash, individually and contractually individually) stable coalitions was considered in this and other references, for example [14] and [6]. A potentially infinitely long coalition formation process in the context of hedonic games was studied in [3]. This leads to another notion of stability analogous to subgame perfect equilibrium.

Recently, [4] compared various notions of stability and equilibria in network formation games. These are games in which the players may be involved in a network relationship that, as a graph, may evolve. Other interaction structures

which players can form were considered in [8], in which formation of hierarchies was studied, and [11] in which only bilateral agreements that follow a specific protocol were allowed. Various aspects of coalition formation are also discussed in the recent collection of articles [9].

In [1] we introduced the concept of a stable partition for coalitional TU-games and investigated whether and how so defined stable partitions can be reached from any initial partition by means of simple transformations. The underlying concept of 'quality' of a partition was defined there by means of social welfare, which is simply the summed value of the partition.

Finally, the computer science perspective is illustrated by [7] in which an approach to coalition formation based on Bayesian reinforcement was considered and tested empirically.

1.2 Approach

In this paper we generalize the approach of [1] and investigate the idea of coalition formation in an abstract setting. To this end we introduce an abstract preference relation $\rhd$ between partitions of any subset of players. We then model coalition formation by means of simple transformations of partitions of the grand coalition through merges and splits that yield a 'local' improvement w.r.t. the $\rhd$ preference relation.

We then turn to the question of identifying conditions to ensure that arbitrary sequences of merges and splits yield the same outcome. We provide an answer to this question by imposing natural conditions on the $\rhd$ preference relation (namely transitivity and monotonicity) and by considering a parametrized concept of a stable partition.

The introduced notion of a stable partition focuses only on the way a group of players is partitioned. Intuitively, a partition P of the grand coalition is stable w.r.t. a class of partioned groups iff no such group gains advantage (modelled by an improvement w.r.t. $\rhd$) by changing the way it is partitioned by P to its own partition.

This way we obtain a generic presentation that allows us to study the idea of coalition formation by focusing only on an abstract concept of the 'quality' of a partition. In particular this analysis does not take into account any allocations to individual players. Also, in our results no specific coalitional game is assumed.

In the setting of coalitional TU-games we obtain results for concrete preference relations induced by specific orders, some of which are discussed in [12], viz. the utilitarian, Nash, egalitarian and leximin orders. We also discuss applications to hedonic games.

In our future work we plan to incorporate into this analysis the concept of a network structure. In this context a *network* is an undirected graph on the set of players that makes explicit the direct links between players. In the presence of a network only coalitions formed by connected players are allowed.

The paper is organized as follows. In the next section we set the stage by

introducing an abstract comparison relation between partitions of a group of players and the corresponding merge and split rules that act on such partitions. Then in Section 3 we discuss a number of natural comparison relations on partitions within the context of coalitional TU-games. Next, in Section 4, we introduce and study a parametrized concept of a stable partition and in Section 5 relate it to the merge and split rules. Finally, in Section 6 we explain how to apply the obtained results to the coalitional TU-games and some classes of hedonic games.

2 Comparing and transforming collections

Let $N = \{1, 2, \ldots, n\}$ be a fixed set of players called the *grand coalition*. Non-empty subsets of N are called *coalitions*. A *collection* (in the grand coalition N) is any family $C := \{C_1, \ldots, C_l\}$ of mutually disjoint coalitions, and l is called its *size*. If additionally $\bigcup_{j=1}^{l} C_j = N$, the collection C is called a *partition* of N. For $C = \{C_1, \ldots, C_k\}$, we define $\bigcup C := \bigcup_{i=1}^{k} C_i$.

In this article we are interested in comparing collections. In what follows we only compare collections A and B that are partitions of the same set, i.e., such that $\bigcup A = \bigcup B$. Intuitively, assuming a comparison relation $\rhd$, $A \rhd B$ means that the way A partitions K, where $K = \bigcup A = \bigcup B$, is preferable to the way B partitions K.

In specific examples we shall deal both with reflexive and non-reflexive transitive relations. So, to keep the presentation uniform we only assume that the relation $\rhd$ is transitive, i.e. for all collections A, B, C with $\bigcup A = \bigcup B = \bigcup C$,

$$A \rhd B \rhd C \text{ imply } A \rhd C, \tag{tr}$$

and that $\rhd$ is monotonic in the following two senses: for all collections A, B, C, D with $\bigcup A = \bigcup B$, $\bigcup C = \bigcup D$, and $\bigcup A \cap \bigcup C = \emptyset$,

$$A \rhd B \text{ and } C \rhd D \text{ imply } A \cup C \rhd B \cup D, \tag{m1}$$

and for all collections A, B, C with $\bigcup A = \bigcup B$ and $\bigcup A \cap \bigcup C = \emptyset$,

$$A \rhd B \text{ implies } A \cup C \rhd B \cup C. \tag{m2}$$

Of course, if $\rhd$ is reflexive (m2) follows from (m1).

The role of monotonicity will become clear in Section 4. If $\rhd$ is reflexive, we may denote it by $\succeq$ and if $\rhd$ is irreflexive, we may denote it by $\succ$.

Definition 2.1. By a *comparison relation* we mean a relation on collections that satisfies the conditions (tr), (m1) and (m2). □

In what follows we study coalition formation by focusing on the following two rules that allow us to transform partitions of the grand coalition:

merge: $\{T_1, \ldots, T_k\} \cup P \to \{\bigcup_{j=1}^{k} T_j\} \cup P$, where $\{\bigcup_{j=1}^{k} T_j\} \rhd \{T_1, \ldots, T_k\}$

split: $\{\bigcup_{j=1}^{k} T_j\} \cup P \to \{T_1, \ldots, T_k\} \cup P$, where $\{T_1, \ldots, T_k\} \rhd \{\bigcup_{j=1}^{k} T_j\}$

Note that both rules use the $\rhd$ comparison relation 'locally', by focusing on the coalitions that take part and result from the merge resp. split. In this paper we are interested in finding conditions that guarantee that arbitrary sequences of these two rules yield the same outcome. So, once these conditions hold, a specific *preferred* partition exists such that any initial partition can be transformed into it by applying the merge and split rules in an arbitrary order.

To start with, the following observation isolates the condition that guarantees the termination of the iterations of these two rules.

Note 2.2. *Suppose that $\rhd$ is an irreflexive comparison relation. Then every iteration of the merge and split rules terminates.*

Proof. Every iteration of these two rules produces by (m2) a sequence of partitions $P_1, P_2, \ldots$ with $P_{i+1} \rhd P_i$ for all $i \geq 1$. But the number of different partitions is finite. So by transitivity and irreflexivity of $\rhd$ such a sequence has to be finite. □

The analysis of the conditions guaranteeing the unique outcome of the iterations is now deferred to Section 5.

3 TU-games

To properly motivate the subsequent considerations and to clarify the status of the monotonicity conditions we now introduce some natural comparison relations on collections for coalitional TU-games. Recall that a *coalitional TU-game* is a pair (v, N), where $N = \{1, \ldots, n\}$ and v is a function from the powerset of N to the set of non-negative reals.[1] In what follows we assume that $v(\emptyset) = 0$.

For a coalitional TU-game (v, N) the comparison relations on collections are induced in a canonic way from the corresponding relations on the multisets of reals, by stipulating that for the collections A and B

$$A \rhd B \text{ iff } v(A) \rhd v(B),$$

where for a collection $A := \{A_1, \ldots, A_m\}$, $v(A) := \dot{\{}v(A_1), \ldots, v(A_m)\dot{\}}$, denoting the multisets using dotted braces.

To take into account payoffs to individual players we need to use the concept of a *value function* ϕ that given a coalition A assigns to each player $i \in A$ a real $\phi^A(i)$ such that $\sum_{i \in A} \phi^A(i) = v(A)$. Then for a collection $A := \{A_1, \ldots, A_m\}$ we put $v(A) := \dot{\{}\phi^{A_j}(i) \mid i \in A_j, j \in \{1, \ldots, m\}\dot{\}}$.

So first we introduce the appropriate relations on the multisets of non-negative reals. The corresponding definition of monotonicity for such a relation

[1] The assumption that the values of v are non-negative is non-standard and is needed only to accomodate for the Nash order, defined below.

$\rhd$ is that for all multisets a, b, c, d of reals

$$a \rhd b \text{ and } c \rhd d \text{ imply } a \dot\cup c \rhd b \dot\cup d$$

and

$$a \rhd b \text{ implies } a \dot\cup c \rhd b \dot\cup c,$$

where $\dot\cup$ denotes the multiset union.

Given two sequences $(a_1, \ldots, a_m)$ and $(b_1, \ldots, b_n)$ of real numbers we define the (extended) *lexicographic order* on them by putting

$$(a_1, \ldots, a_m) >_{lex} (b_1, \ldots, b_n)$$

iff

$$\exists i \leq min(m, n)\, (a_i > b_i \wedge \forall j < i\, a_j = b_j)$$

or

$$\forall i \leq min(m, n)\, a_i = b_i \wedge m > n.$$

Note that in this order we compare sequences of possibly different length. We have for example $(1, 1, 1, 0) >_{lex} (1, 1, 0)$ and $(1, 1, 0) >_{lex} (1, 1)$. It is straightforward to check that it is a linear order.

We assume below that $a = \dot\{a_1, \ldots, a_m\dot\}$ and $b = \dot\{b_1, \ldots, b_n\dot\}$ and that a^* is a sequence of the elements of a in decreasing order, and define

- the *utilitarian* order:
  $a \succ_{ut} b$ iff $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$,
- the *Nash* order:
  $a \succ_{Nash} b$ iff $\Pi_{i=1}^m a_i > \Pi_{j=1}^n b_j$,
- the *elitist* order:
  $a \succ_{el} b$ iff $max(a) > max(b)$,
- the *egalitarian* order:
  $a \succ_{eg} b$ iff $min(a) > min(b)$,
- the *leximin* order:
  $a \succ_{lex} b$ iff $a^* >_{lex} b^*$.

In [12] these orders were considered for the sequences of the same length. The intuition behind the Nash order is that when the sum $\sum_{i=1}^m a_i$ is fixed, the product $\Pi_{i=1}^m a_i$ is largest when all a_is are equal. So in a sense the Nash order favours an equal distribution.

For the first four relations, the corresponding reflexive counterparts are obtained by replacing $>$ by $\geq$. In turn, $\succeq_{lex}$, the reflexive version of $\succ_{lex}$, is obtained by additionally including all pairs of equal multisets. Note that all these preorders are in fact linear (i.e., total) preorders.

Note 3.1. *The above relations are all monotonic both in sense* (m1) *and* (m2).

Proof. The only relations for which the claim is not immediate are $\succ_{lex}$ and $\succeq_{lex}$. We will only prove (m1) for $\succ_{lex}$; the remaining proofs are analogous.

Let arbitrary multisets of non-negative reals a, b, c, d be given. We define, with e denoting any sequence or multiset of non-negative reals,

$$\begin{aligned} len(e) &:= \text{the number of elements in } e, \\ \mu &:= (a \dot\cup b \dot\cup c \dot\cup d)^* \text{ with all duplicates removed}, \\ \nu(x, e) &:= \text{the number of occurrences of } x \text{ in } e, \\ \beta &:= 1 + \max_{k=1}^{len(\mu)} \{\nu(\mu_k, a \dot\cup b \dot\cup c \dot\cup d)\}, \\ \#(e) &:= \sum_{k=1}^{len(\mu)} \nu(\mu_k, e) \cdot \beta^{-k}. \end{aligned}$$

So μ is the sequence of all distinct reals used in $a \dot\cup b \dot\cup c \dot\cup d$, arranged in a decreasing order. The function $\#(\cdot)$ injectively maps a multiset e to a real number y in such a way that in the floating point representation of y with base β, the kth digit after the point equals the number of occurrences of the kth biggest number μ_k in e. The base β is chosen in such a way that even if e is the union of some of the given multisets, the number $\nu(x, e)$ of occurrences of x in e never exceeds $\beta - 1$. Therefore, the following sequence of implications holds:

$$\begin{aligned} a^* >_{lex} b^* \text{ and } c^* >_{lex} d^* &\Rightarrow \#(a) > \#(b) \text{ and } \#(c) > \#(d) \\ &\Rightarrow \#(a) + \#(c) > \#(b) + \#(d) \\ &\Rightarrow \#(a \dot\cup c) > \#(b \dot\cup d) \\ &\Rightarrow (a \dot\cup c)^* >_{lex} (b \dot\cup d)^* \end{aligned}$$

□

As a natural example of a transitive relation that is not monotonic consider $\succeq_{av}$ defined by

$$a \succeq_{av} b \text{ iff } (\textstyle\sum_{i=1}^{m} a_i)/m \geq (\sum_{j=1}^{n} b_j)/n.$$

Note that for

$$a := \dot\{3\dot\}, b := \dot\{2, 2, 2, 2\dot\}, c := \dot\{1, 1, 1, 1\dot\}, d := \dot\{0\dot\}$$

we have both $a \succeq_{av} b$ and $c \succeq_{av} d$ but not $a \dot\cup c \succeq_{av} b \dot\cup d$ since $\dot\{3, 1, 1, 1, 1\dot\} \succeq_{av} \dot\{2, 2, 2, 2, 0\dot\}$ does not hold.

4 Stable partitions

We now return to our study of collections. One way to identify conditions guaranteeing the unique outcome of the iterations of the merge and split rules is through focusing on the properties of such a unique outcome. This brings us to a concept of a stable partition.

We follow here the approach of [1], although now no notion of a game is present. The introduced notion is parametrized by means of a *defection function* $\mathbb{D}$ that assigns to each partition some partitioned subsets of the grand coalition. Intuitively, given a partition P the family $\mathbb{D}(P)$ consists of all the collections $C := \{C_1, \ldots, C_l\}$ whose players can leave the partition P by forming a new, separate, group of players $\cup_{j=1}^{l} C_j$ divided according to the collection C. Two most natural defection functions are $\mathbb{D}_p$, which allows formation of all partitions of the grand coalition, and $\mathbb{D}_c$, which allows formation of all collections in the grand coalition.

Next, given a collection C and a partition $P := \{P_1, \ldots, P_k\}$ we define

$$C[P] := \{P_1 \cap \bigcup C, \ldots, P_k \cap \bigcup C\} \setminus \{\emptyset\}$$

and call $C[P]$ the *collection C in the frame of P*. (By removing the empty set we ensure that $C[P]$ is a collection.) To clarify this concept consider Figure 1. We depict in it a collection C, a partition P and C in the frame of P (together with P). Here C consists of three coalitions, while C in the frame of P consists of five coalitions.

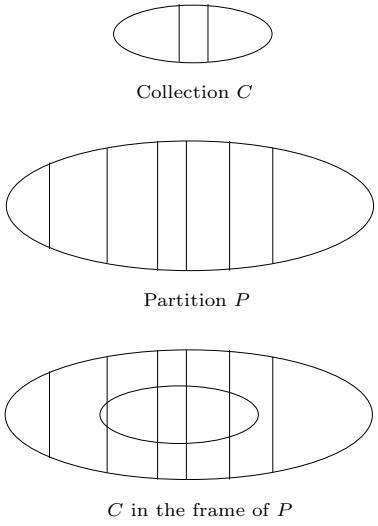


Figure 1: A collection C in the frame of a partition P

Intuitively, given a subset S of N and a partition $C := \{C_1, \ldots, C_l\}$ of S, the collection C offers the players from S the 'benefits' resulting from the partition

of S by C. However, if a partition P of N is 'in force', then the players from S enjoy instead the benefits resulting from the partition of S by $C[P]$, i.e., C in the frame of P.

To get familiar with the $C[P]$ notation note that

- if C is a singleton, say $C = \{T\}$, then $\{T\}[P] = \{P_1 \cap T, \ldots, P_k \cap T\} \setminus \{\emptyset\}$, where $P = \{P_1, \ldots, P_k\}$,
- if C is a partition of N, then $C[P] = P$,
- if $C \subseteq P$, that is C consists of some coalitions of P, then $C[P] = C$.

In general the following simple observation holds.

Note 4.1. *For a collection C and a partition P, $C[P] = C$ iff each element of C is a subset of a different element of P.* □

This brings us to the following notion.

Definition 4.2. Assume a defection function $\mathbb{D}$ and a comparison relation $\rhd$. We call a partition P $\mathbb{D}$-stable if $C[P] \rhd C$ for all $C \in \mathbb{D}(P)$ such that $C[P] \neq C$.

The last qualification, that is $C[P] \neq C$, requires some explanation. First note that if C is a partition of N, then $C[P] \neq C$ is equivalent to the statement $P \neq C$, since then $C[P] = P$. So in the case of the $\mathbb{D}_p$ defection function we have the following simpler definition.

Theorem 4.3. *A partition P is $\mathbb{D}_p$-stable iff for all partitions $P' \neq P$, $P \rhd P'$ holds.* □

Corollary 4.4. *Suppose that $\rhd$ is an irreflexive linear comparison relation. Then a $\mathbb{D}_p$-stable partition exists.* □

Next, if we deal with a reflexive comparison relation $\succeq$, then the qualification $C[P] \neq C$ can be dropped, as then $C[P] = C$ implies $C[P] \succeq C$. However, if we deal with an irreflexive comparison relation $\succ$, then this qualification is of course necessary. So using it we can deal with the irreflexive and reflexive case in a uniform way.

Intuitively, the condition $C[P] \neq C$ indicates that the players only care about the way they are partitioned. Indeed, if $C[P] = C$, then the partitions of $\bigcup C$ by means of P and by means of C coincide and are viewed as equally satisfactory for the players in $\bigcup C$. By disregarding the situations in which $C[P] = C$ we therefore adopt a limited viewpoint of cooperation according to which the players in C do not care about the presence of the players from outside of $\bigcup C$ in their coalitions.

The definition of $\mathbb{D}$-stability calls for checks involving (almost) all collections from $\mathbb{D}$(P). In the case of the $\mathbb{D}_c$ defection function, we can considerably simplify these checks as the following characterization results shows. Given a partition $P := \{P_1, \ldots, P_k\}$ we call here a coalition T *P-compatible* if for some $i \in \{1, \ldots, k\}$ we have $T \subseteq P_i$ and *P-incompatible* otherwise.

Theorem 4.5. *A partition $P = \{P_1, \ldots, P_k\}$ of N is $\mathbb{D}_c$-stable iff the following two conditions are satisfied:*

(i) for each $i \in \{1, \ldots, k\}$ and each pair of disjoint coalitions A and B such that $A \cup B \subseteq P_i$

$$\{A \cup B\} \rhd \{A, B\}, \tag{1}$$

(ii) for each P-incompatible coalition $T \subseteq N$

$$\{T\}[P] \rhd \{T\}. \tag{2}$$

Proof. ($\Rightarrow$) Immediate.

($\Leftarrow$) Transitivity (tr), monotonicity (m2) and (1) imply by induction that for each $i \in \{1, \ldots, k\}$ and each collection $C = \{C_1, \ldots, C_l\}$ with $l > 1$ and $\bigcup C \subseteq P_i$,

$$\left\{\bigcup C\right\} \rhd C. \tag{3}$$

Let now C be an arbitrary collection in N such that $C[P] \neq C$. We prove that $C[P] \rhd C$. Define

$$D^i := \{T \in C \mid T \subseteq P_i\},$$
$$E := C \setminus \textstyle\bigcup_{i=1}^{k} D^i,$$
$$E^i := \{P_i \cap T \mid T \in E\} \setminus \{\emptyset\}.$$

Note that D^i is the set of P-compatible elements of C contained in P_i, E is the set of P-incompatible elements of C and E^i consists of the non-empty intersections of P-incompatible elements of C with P_i.

Suppose now that $\bigcup_{i=1}^{k} E^i \neq \emptyset$. Then $E \neq \emptyset$ and consequently

$$\bigcup_{i=1}^{k} E^i = \bigcup_{i=1}^{k} (\{P_i \cap T \mid T \in E\} \setminus \{\emptyset\}) = \bigcup_{T \in E} \{T\}[P] \overset{(\mathrm{m1}),(2)}{\rhd} E. \tag{4}$$

Consider now the following property:

$$|D^i \cup E^i| > 1. \tag{5}$$

Fix $i \in \{1, \ldots, k\}$. If (5) holds, then

$$\left\{P_i \cap \bigcup C\right\} = \left\{\bigcup (D^i \cup E^i)\right\} \overset{(3)}{\rhd} D^i \cup E^i$$

and otherwise

$$\left\{P_i \cap \bigcup C\right\} = \left\{D^i \cup E^i\right\}.$$

Recall now that

$$C[P] = \bigcup_{i=1}^{k} \left\{P_i \cap \bigcup C\right\} \setminus \{\emptyset\}.$$

We distinguish two cases.

Case 1. (5) holds for some $i \in \{1, \ldots, k\}$.

Then by (m1) and (m2)

$$C[P] \rhd \bigcup_{i=1}^{k}(D^i \cup E^i) = (C \setminus E) \cup \bigcup_{i=1}^{k} E^i.$$

If $\bigcup_{i=1}^{k} E^i = \emptyset$, then also $E = \emptyset$ and we get $C[P] \rhd C$. Otherwise by (4), (tr) and (m2)

$$C[P] \rhd (C \setminus E) \cup E = C.$$

Case 2. (5) does not hold for any $i \in \{1, \ldots, k\}$.

Then

$$C[P] = \bigcup_{i=1}^{k}(D^i \cup E^i) = (C \setminus E) \cup \bigcup_{i=1}^{k} E^i.$$

Moreover, because $C[P] \neq C$, by Note 4.1 a P-incompatible element in C exists. So $\bigcup_{i=1}^{k} E^i \neq \emptyset$ and by (4) and (m2) we get as before

$$C[P] \rhd (C \setminus E) \cup E = C.$$

□

In [1] this theorem was proved for the coalitional TU-games and both the irreflexive and the reflexive utilitarian orders. The above result isolates the relevant conditions that the comparison relation, here $\rhd$, needs to satisfy.

In contrast to the case of the $\mathbb{D}_p$-stable partitions, as shown in [1], a $\mathbb{D}_c$-stable partition does not need to exist, even if $\rhd$ is irreflexive. In that paper a natural class of TU-games is defined for which $\mathbb{D}_c$-stable partitions are guaranteed to exist. In Section 6 we introduce a natural class of hedonic games for which $\mathbb{D}_c$-stable partitions exist.

5 Stable partitions and merge/split rules

We now resume our investigation of the conditions under which every iteration of the merge and split rules yields the same outcome. With this in mind we establish the following results concerned with the $\mathbb{D}_c$ defection function.

Note 5.1. *If $\rhd$ is an irreflexive comparison relation, then every $\mathbb{D}_c$-stable partition P is closed under the applications of the merge and split rules.*

Proof. To prove the closure under merge rule assume that for some $\{T_1, \ldots, T_k\} \subseteq P$ we have $\{\bigcup_{j=1}^{k} T_j\} \rhd \{T_1, \ldots, T_k\}$. $\mathbb{D}_c$-stability of P with $C := \{\bigcup_{j=1}^{k} T_j\}$ yields

$$\{T_1, \ldots, T_k\} = \{\bigcup_{j=1}^{k} T_j\}[P] \rhd \{\bigcup_{j=1}^{k} T_j\},$$

which is a contradiction by virtue of the transitivity and irreflexivity of $\rhd$.

The closure under the split rule is shown analogously. □

Lemma 5.2. *Assume that $\rhd$ is an irreflexive comparison relation and P is $\mathbb{D}_c$-stable. Let P' be closed under applications of merge and split rules. Then $P' = P$.*

Proof. Suppose $P = \{P_1, \ldots, P_k\}$, $P' = \{T_1, \ldots, T_m\}$. Assume $P \neq P'$. Then there is $i_0 \in \{1, \ldots, k\}$ such that for all $j \in \{1, \ldots, m\}$ we have $P_{i_0} \neq T_j$. Let $T_{j_1}, \ldots, T_{j_l}$ be the minimum cover of P_{i_0}. In the following case distinction we use Theorem 4.5.

Case 1. $P_{i_0} = \bigcup_{h=1}^{l} T_{j_h}$.

Then $\{T_{j_1}, \ldots, T_{j_l}\}$ is a proper partition of P_{i_0}. But (1) (through its generalization to (3)) yields $P_{i_0} \rhd \{T_{j_1}, \ldots, T_{j_l}\}$, thus the merge rule is applicable to P'.

Case 2. $P_{i_0} \subsetneq \bigcup_{h=1}^{l} T_{j_h}$.

Then for some j_h we have $\emptyset \neq P_{i_0} \cap T_{j_h} \subsetneq T_{j_h}$, so T_{j_h} is P-incompatible. By (2) we have $\{T_{j_h}\}[P] \rhd \{T_{j_h}\}$, thus the split rule is applicable to P'. □

This allows us to conclude the following result that answers our initial question and clarifies the importance of the $\mathbb{D}_c$-stable partitions.

Theorem 5.3. *Suppose that $\rhd$ is an irreflexive comparison relation and P is a $\mathbb{D}_c$-stable partition. Then*

(i) P is the outcome of every iteration of the merge and split rules.

(ii) P is a unique $\mathbb{D}_p$-stable partition.

(iii) P is a unique $\mathbb{D}_c$-stable partition.

Proof. (*i*) By Note 2.2 every iteration of the merge and split rules terminates, so the claim follows by Lemma 5.2.

(*ii*) Since P is $\mathbb{D}_c$-stable, it is in particular $\mathbb{D}_p$-stable. Uniqueness follows from the transitivity and irreflexivity of $\rhd$ by virtue of Theorem 4.3.

(*iii*) Suppose that P' is a $\mathbb{D}_c$-stable partition. By Note 5.1 P' is closed under the applications of the merge and split rules, so by Lemma 5.2 $P' = P$. □

This generalizes the considerations of [1], where this result was established for the coalitional TU-games and the irreflexive utilitarian order. It was also shown there that there exist coalitional TU-games in which all iterations of the merge and split rules have a unique outcome which is not a $\mathbb{D}_c$-stable partition.

6 Hedonic games

Note that the results of the last two sections do not involve any notion of a game. Only by choosing the monotonic comparison relations introduced in Section 3 we obtain specific results that deal with coalitional TU-games.

These considerations also readily apply to NTU-games. However, one needs to be careful since the resulting notion of a stable coalition can be in some situations counterintuitive. To clarify the limitation of this approach we now focus on the hedonic games (see, e.g., [5]) that form a specific class of NTU-games. Recall that a *hedonic game* $(N, \succeq_1, \ldots, \succeq_n)$ consists of a set of players $N = \{1, \ldots, n\}$ and a sequence of linear preorders $\succeq_1, \ldots, \succeq_n$, where each $\succeq_i$ is the preference of player i over the subsets of N containing i. In what follows we shall not need the assumption that the $\succeq_i$ relations are linear.

Again, we let $\succ_i$ denote the associated irreflexive relation. Given a partition A of N and player i we denote by $A(i)$ the element of A to which i belongs and call it the set of *friends of* i *in* A. Given a hedonic game $(N, \succeq_1, \ldots, \succeq_n)$ a natural preference relation on the collections is given by:

$$A \succeq B \text{ iff } \neg\exists C \in B \, \forall i \in C . C \succ_i A(i), \tag{6}$$

where $\bigcup A = \bigcup B$.

It states that A is preferred over B unless B contains a coalition C such that each player in C strictly prefers C to his coalition in A. Clearly $\succeq$ is monotonic. The notion of $\mathbb{D}_p$-stability then coincides with the notion of core stability in [5].

However, the resulting notion of a $\mathbb{D}_c$-stable partition can contradict the intuition. To see this consider the following example.

Example 6.1. Suppose $N = \{1, 2, 3, 4\}$. Consider a hedonic game in which

$$\{2\} \succ_2 \{2, 3\} \succ_2 \{1, 2\}$$

and

$$\{3\} \succ_3 \{2, 3\} \succ_3 \{3, 4\}.$$

Now take $P = \{\{1, 2\}, \{3, 4\}\}$ and $C = \{\{2, 3\}\}$. Then $C[P] = \{\{2\}, \{3\}\}$. So both players 2 and 3 strictly prefer their coalition in $C[P]$ to the one in C and consequently P is 'stable' w.r.t. collection C. In fact, it is straightforward to extend the above ordering in such a way that P is $\mathbb{D}_c$-stable.

However, both players 2 and 3 favour the coalition $\{2, 3\}$ higher than their coalition within P, so intuitively P should not be stable. □

The difficulty in the above example arises from the fact that in players' preferences smaller coalitions can be preferred over the larger ones. Natural hedonic games in which this is not the case can be derived from arbitrary partitions of the set of players. Given a partition $P := \{P_1, \ldots, P_k\}$ of N we assume that each player

- prefers a larger set of friends over a smaller one,

- only 'cares' about the sets of his friends in P.

We formalize this order by putting for all sets of players that include i

$$S \succeq_i T \text{ iff } S \cap P(i) \supseteq T \cap P(i).$$

With this definition, all partitions which result from arbitrary (including no) applications of the merge rule to P are $\mathbb{D}_c$-stable w.r.t. the reflexive comparison relation $\succeq$ defined in (6).

Next, we provide an example of a hedonic game in which a $\mathbb{D}_c$-stable partition w.r.t. to a natural irreflexive comparison relation $\succ$ exists. To this end given a partition $P := \{P_1, \ldots, P_k\}$ of N we now assume that each player

- prefers a larger set of his friends in P over a smaller one,
- 'dislikes' coalitions that include a player who is not his friend in P.

We formalize this by putting for all sets of players that include i

$$S \succeq_i T \text{ iff } S \cup T \subseteq P(i) \text{ and } S \supseteq T,$$

and by extending this order to the coalitions that include player i and also a player from outside of $P(i)$ by assuming that they are the minimal elements in $\succeq_i$. So $S \succ_i T$ iff either $S \cup T \subseteq P(i)$ and $S \supset T$ or $S \subseteq P(i)$ and $\neg T \subseteq P(i)$.

We then define an irreflexive comparison relation on collections by

$$A \succ B \text{ iff for } i \in \{1, \ldots, n\} \; A(i) \succeq_i B(i) \text{ with at least one } \succeq_i \text{ being strict.}$$

It is straightforward to check that for this comparison relation the partition $\{P_1, \ldots, P_k\}$ satisfies the conditions (1) and (2) of Theorem 4.5. So by virtue of this theorem $\{P_1, \ldots, P_k\}$ is $\mathbb{D}_c$-stable. Further, by virtue of Theorem 5.3, $\{P_1, \ldots, P_k\}$ can be reached from any initial partition through an arbitrary sequence of the applications of the split and merge rules.

Acknowledgement

We thank Tadeusz Radzik for helpful comments.

References

[1] K. R. Apt and T. Radzik. Stable partitions in coalitional games, 2006. Available from `http://arxiv.org/abs/cs.GT/0605132`.

[2] R.J. Aumann and J.H. Drèze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3:217–237, 1974.

[3] F. Bloch and E. Diamantoudi. Noncooperative formation in coalitions in hedonic games, 2005. Working paper.

[4] F. Bloch and M. Jackson. Definitions of equilibrium in network formation, 2005. Working paper.

[5] A. Bogomolnaia and M. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[6] N. Burani and W.S. Zwicker. Coalition formation games with separable preferences. *Mathematical Social Sciences*, 45(1):27–52, 2003.

[7] G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-04)*, pages 1090–1097, 2004.

[8] G. Demange. On group stability in hierarchies and networks. *Journal of Political Economy*, 112(4):754–778, 2004.

[9] G. Demange and M. Wooders, editors. *Group Formation in Economics*. Cambridge University Press, 2006.

[10] J. Greenberg. Coalition structures. In R.J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 2 of *Handbook of Game Theory with Economic Applications*, chapter 37, pages 1305–1337. Elsevier, 1994.

[11] I. Macho-Stadler, D. Prez-Castrillo, and N. Porteiro. Sequential formation of coalitions through bilateral agreements, 2005. Working paper.

[12] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1998.

[13] D. Ray and R. Vohra. Equilibrium binding agreements. *Journal of Economic Theory*, (73):30–78, 1997.

[14] T. Sönmez, S. Banerjee, and H. Konishi. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.

[15] S.S. Yi. Stable coalition structures with externalities. *Games and Economic Behavior*, 20:201–237, 1997.

Krzysztof R. Apt
CWI
1098 SJ Amsterdam, The Netherlands
Email: K.R.Apt@cwi.nl

Andreas Witzel
ILLC, University of Amsterdam
1018 TV Amsterdam, The Netherlands
Email: awitzel@illc.uva.nl

Welfarism and the assessment of social decision rules

Claus Beisbart and Stephan Hartmann

Abstract

The choice of a social decision rule for a federal assembly affects the welfare distribution within the federation. But which decision rules can be recommended on welfarist grounds? In this paper, we focus on two welfarist desiderata, viz. (i) maximizing the expected utility of the whole federation and (ii) equalizing the expected utilities of people from different states in the federation. We consider the European Union as an example, set up a probabilistic model of decision making and explore how different decision rules fare with regard to the desiderata. We start with a default model, where the interests, and therefore the votes of the different states are not correlated. This default model is then abandoned in favor of models with correlations. We perform computer simulations and find that decision rules with a low acceptance threshold do generally better in terms of desideratum (i), whereas the rules presented in the Accession Treaty and in the (still unratified) Constitution of the European Union tend to do better in terms of desideratum (ii). The ranking obtained regarding desideratum (i) is fairly stable across different correlation patterns.

1 Introduction

For a long time, social choice theory has been dominated by axiomatic approaches in the tradition of Arrow ([1]) and Sen ([9]). There works typically start with a few axioms that put intuitively reasonable constraints on the social welfare function, for instance. Unfortunately, it turns out that these constraints cannot be fulfilled at the same time. Impossibility results of this kind are very exciting. But they are of no help, if we are to decide between different social decision rules.

Consider the European Union as an example. Many decisions are taken by the European Council of Ministers (Council, for short). It works in the following way: Each state of the European Union sends a representative to the Council. The European Commission drafts a proposal, and the representatives cast their votes on behalf of the states. The votes are aggregated, and a decision is taken according to some decision rule. But which rule is most appropriate? Impossibility results do not answer this question.

In this paper, we will take a different line of thought. We will start with simple principles that spell out what makes a decision rule pro tanto better than another one. We will then evaluate decision rules according to these principles. As we will see, this requires us to set up a different framework (see [4]); and

we will need to use new mathematical techniques and computational methods such as computer simulations.

We choose a *welfarist* framework to evaluate alternative decision rules. It is based on the following simple idea. The outcomes of a decision affect the welfares of the people in the federation. A particular outcome may benefit some people, it may harm other people, and it may make no difference to yet others. Now different decision rules lead to different outcomes. As a consequence, different decision rules result in different welfare distributions.

But which decision rule is best? To address this question, the welfare distribution that results from the adoption of a certain decision rule has to be evaluated, and we propose to evaluate it according to the following two welfarist principles:

Utilitarianism Decision rule D_1 is pro tanto better than decision rule D_2, if the expected utility is larger under D_1 than under D_2 (cp. [4]).

Egalitarianism Decision rule D_1 is pro tanto better than decision rule D_2, if there is more equality in the distribution of the expected utilities across the federation under D_1 than under D_2.[1]

We consider the European Union as an example. Over the last years, there has been a lot of controversy about the question which decision rule to adopt for the Council of Ministers (see, for example, [7]). Various decision rules have been suggested and a large number of arguments has been put forward for each of them.

In previous work, we examined these proposals from a welfarist perspective [2] and assumed that the interests of the different states are uncorrelated. But this is too strong an idealization, as similar states have similar interests and therefore tend to cast the same votes. The new members of the EU are a case in point. They have similar problems and have to meet similar challenges; so a proposal that benefits, say, Poland will typically also benefit Slovenia; and proposals that harm Poland, will also harm Slovenia. There might also be negative correlations. For instance, a proposal which is good for the large states might be bad for the small states, and vice versa. This presence of correlations in the interests of states (and their corresponding voting behavior) raises the question if the decision rules that do best for uncorrelated interests will also do best if interstate correlations are taken into account. This is the question we will address in this paper.

The remainder of this paper is organized as follows: Section 2 introduces our framework and lays out some of the relevant mathematics. The following section 3 shows some of our results for vanishing correlations. Section 4 explains how correlations can be modeled in our framework. We introduce four different correlation patterns and run computer simulations. The results of these simulations are presented and discussed in section 5. The papers ends, in section 6, with some more general reflections.

[1] To make this principle more precise, an equality measure will be specified below.

2 Welfarism and social decision rules

The basic idea of the welfarist approach to social decision making is that, whether a proposal is accepted or rejected, makes a difference to the welfare of the people in the federation. Here is an illustration. Let us assume that it is proposed to construct a freeway in Portugal. If the proposal is rejected, nothing changes. No one can profit from the freeway, and no one has to pay for it. If the proposal is accepted, then the people of Portugal will, on average, gain utility (as they get, for example, faster to work every day) while the people in, say, Austria will, on average, loose utility as they have to contribute to its costs without having a chance to use it that often.

Whether a proposal is accepted or not depends on the decision rule. Weighted decision rules assign different weights to different states. Consider the freeway example again and assume that a weighted rule is adopted. Clearly, if the weights of Portugal are much larger than the weights of Austria, then the freeway proposal might get accepted with the result that the Portugese can sleep longer in the morning and the Austrians are left with the bill. If, on the other hand, the weights of Austria are much larger than the weights of Portugal, then the situation might be the other way round. In the end, the challenge is to find a decision rule that leads to a good welfare distribution according to our principles.

But there is a challenge ahead: We do not know the proposals beforehand. To account for this uncertainty, we set up a probabilistic framework.

Let us now formalize these ideas. We consider a federation of m states with a total number of N people. States are numbered from 1 to m and labeled by lowercase letters (e.g. i, j). The i^{th} state has N_i inhabitants. Of course, $\sum_i N_i = N$.

We model the proposals as exogeneous. A single proposal is represented by a utility vector $\mathbf{v} = (v_1, \ldots, v_m)$. Here v_i is the average utility that people from state i will receive, if the proposal is accepted.[2] v_i is positive, if there is an average gain in utility for people from state i, and it is negative, if there is an average loss in utility for people from state i. The status quo is normalized to 0 and a rejected proposal leads to a zero average utility transfer. Since we do not know the proposals in advance, the utilities v_i are values of random variables V_i $(i = 1, \ldots, m)$.

The vote of state i (or its representative) is described by another random variable Λ_i with values λ_i. $\lambda_i = -1$ means that state i votes against the proposal, and $\lambda_i = +1$ means state i votes for the proposal. $(\lambda_1, \ldots, \lambda_m)$ is a *voting profile*.

[2] *Average* utilities should not be confused with *expected* utilities which we will discus below. Average utilities are means over people, expected utilities are means over different proposals that follow a particular probability distribution. Note also that we start with a rather coarse-grained description of decision making. A more fine-grained view would begin with the utilities of the individual people in the federation. Accordingly, we will only consider inequality at a coarse-grained level, i.e. on the level of states, and not of individual people.

How do states vote, if a certain proposal $\mathbf{v}$ is on the table? We assume that each state examines the average utility that the proposal will confer to its own people. If the average utility is positive, it will vote in favour of the proposal. If the average utility is negative, it will vote against the proposal. In mathematical terms, the vote of state i is then given by $\lambda_i = \mathrm{sign}(v_i)$.[3]

A *decision rule* can be represented as a function D from voting profiles $(\lambda_1, \ldots, \lambda_m)$ to $\{0, 1\}$. It takes the value 1, if the proposal is accepted, and the value 0, if the proposal is rejected.

Suppose now, that a decision rule has been adopted and that a particular proposal $\mathbf{v}$ is on the table. How will the decision affect the average utilities for the different states? Call u_i the average utility that people from state i will receive from a decision on $\mathbf{v}$. According to our assumptions, we have:

$$u_i = v_i \times D\left(\lambda_1(v_1), \ldots, \lambda_m(v_m)\right) . \tag{1}$$

Since the v_is are values of random variables, so are the u_is. We denote the corresponding random variables by U_i for $i = 1, \ldots, m$.

The expectation values of these random variables, $E[U_i]$, are the key quantities in our welfarist framework. Once we know them, we can calculate other quantities that are required by our two principles. Utilitarianism requires the average expected utility of a person in the EU which is given by

$$E[U] = \frac{1}{N} \sum_i N_i \, E[U_i] \, . \tag{2}$$

Egalitarianism requires an equality measure. To keep things simple, we measure the spread in the distribution of the $E[U_i]$s. Let us call this measure I. If I is small, then the equality in the federation is high. If I is large, then the equality is low. [4]

Let us now calculate the expected utility $E[U_i]$ for state i. To do so, we need the joint probability distribution $p(\mathbf{v})$ over the proposed utilities. According to Eq. (1), we have

$$E[U_i] = \int \mathrm{d}\mathbf{v} \, p(\mathbf{v}) \, v_i \, D\left(\lambda_1(v_1), \ldots, \lambda_m(v_m)\right) , \tag{3}$$

where the integral over $\mathrm{d}\mathbf{v}$ is m-dimensional. Note that the decision rule D is a function of the voting profiles which are, in turn, a function of the v_is.

For further analytical calculations, Eq. (3) can be rendered more manageable. To do so, we hold a voting profile $(\lambda_1, \ldots, \lambda_m)$ fixed. The probability that voting profile $(\lambda_1, \ldots, \lambda_m)$ occurs is $p(\lambda_1, \ldots, \lambda_m)$. It is given by

$$p(\lambda_1, \ldots, \lambda_m) = \int \mathrm{d}\mathbf{v} \, \theta(\lambda_1 v_1) \ldots \theta(\lambda_m v_m) \, . \tag{4}$$

[3] We need not consider the case of $v_i = 0$ here, as it has zero measure under any reasonable probability distribution.

[4] We assume here that each person in state i receives the average utility $E[U_i]$ and calculate the standard deviation of the expected utilities of single people. Note that this is nothing but a first quick-and-dirty estimate of the inequality in the federation. There are other measures, such as the Gini coefficient, that might be more appropriate.

Similarly, we calculate the expected utility of state i if the voting profile is $(\lambda_1, \ldots, \lambda_m)$:

$$\overline{v}_i^{\lambda_1,\ldots,\lambda_m} = \int \mathrm{d}\mathbf{v}\, v_i \theta(\lambda_1 v_1) \ldots \theta(\lambda_m v_m) / p(\lambda_1, \ldots, \lambda_m)\,. \tag{5}$$

With $p(\lambda_1, \ldots, \lambda_m)$ and $\overline{v}_i^{\lambda_1,\ldots,\lambda_m}$ we can now calculate the expected utility $E[U_i]$ of state i:

$$E[U_i] = \sum_{\lambda_1,\ldots,\lambda_m} \overline{v}_i^{\lambda_1,\ldots,\lambda_m} \times p(\lambda_1, \ldots, \lambda_m)\, D(\lambda_1, \ldots, \lambda_m)\,. \tag{6}$$

To simplify things a bit more, we assume that the marginals for the different states, i.e.

$$p_i(v_i) = \int \mathrm{d}v_1 \cdots \int \mathrm{d}v_{i-1} \int \mathrm{d}v_{i+1} \cdots \int \mathrm{d}v_m\, p(\mathbf{v})\,, \tag{7}$$

are identical. This means that, on the level of the proposals, there is no bias towards one or the other state. We furthermore assume that the marginals are normally distributed with a mean μ and a standard deviation σ. All utilities are scaled such that $\sigma = 1$.[5]

3 Independent utilities from proposals

In order to explore the welfarist framework, we start with a simple *default model* in which the V_is are independent. We will later relax this assumption. In the default model, the joint probability distribution $p(\mathbf{v})$ factorizes:

$$p(\mathbf{v}) = p_i(v_i) \cdots p_m(v_m)\,. \tag{8}$$

This means that the utilities from proposals are uncorrelated for the various states. If one knows that a proposal puts benefits on the Fins, one cannot infer anything about the benefits or harms for people from other states. In order to refer to Eq. (8) more quickly, we will somehow loosely say that the states are independent. Note, however, that, even under Eq. (8), the random variables U_i are *not* independent, but correlated. The reason is that the decision takes all v_is into account.

Under the assumption of Eq. (8), the sum in Eq. (6) can be worked out analytically or directly calculatd by a computer program. For details, see [2].

To apply our methodology to the decision making in the European Union, we consider five decision rules that were discussed in the context of the constitutional reform of the EU.[6] These decision rules can be organized into two

[5] If the utilities are independent in the same state, we would expect, according to the central limit theorem, that the standard deviations for the different states are proportional to $1/\sqrt{N_i}$. However, [3] present a model with correlations within the same state that justifies our choice of identical standard deviations.

[6] For a complementary approach in terms of expected utility see [4].

groups. In the first group are three *theoretical rules* that assign a weight w_i proportional to N_i^α with $0 \leq \alpha \leq 1$ ([5]) to each state i (see [6], Chapter 2). The weights are normalized to 1, i.e. $\sum_i w_i = 1$ and a proposal is accepted if the combined weights of the states which vote for the proposal exceeds a threshold of .5. We consider the following theoretical rules.

(SME) Simple majority with equal weights ($\alpha = 0$).

(P50) Simple majority with square root weights ($\alpha = .5$).[7]

(SME) Simple majority with proportional weights ($\alpha = 1$).

In the second group are two *political rules*, which are more complex than the theoretical rules. Here each state is assigned several weights, which are aggregated seperately. A proposal is accepted, if the aggregates exceed their respective thresholds (for details see [2], Section 2).

(Acc) This rule, which is formulated in the Accession Treaty and which builds on the Nice Treaty, is presently in force. It identifies three classes of weights, one with $\alpha = 0$ (threshold 50%), one with $\alpha = 1$ (62%), and one with an unsystematic weights (72%).

(Con) This rule is part of the Constitution that is presently in the process of ratification. It identifies two classes of weights, one with $\alpha = 0$ (threshold 58%) and one with $\alpha = 1$ (65%).

Let us now briefly consider results for the default model (for details, see [2]). In Figure 1 we show the expected utility of an average person in the EU (left panel) and our measure of inequality (right panel). The larger the spread, the more inequality we find in the federation. Our characteristics are shown as a function of μ, the mean over the utilities from proposals.

Let us first consider expected utility. For μ significantly smaller than 0, proposals are typically bad. They are therefore mostly rejected, and the utilities of the people in the federation do not change. A closer inspection of the curves shows that the political rules do slightly better for a range of negative μ-values.

For μ significantly larger than 0, the proposals are typically very good. Therefore, most of them are accepted under any decision rule. As the utilities are now conferred to the people, $E[U]$ will be positive. For $\mu > 1$, the curves for the different rules almost coincide.

The most interesting range is the one around $\mu = 0$. This is also the most realistic range of parameters, as we argue in sec. 5 of [2]. In this range the decision rules yield significantly different results. The general trend is that the theoretical rules do better. At $\mu = 0$, SMP is the best rule, followed by P50 and SME.

Let us now turn to equality. As the right panel of Figure 1 shows, SMP does very badly in terms of equality for $\mu \approx 0$. It is followed by P50 and the political rules. SME exactly equalizes the expected utilities for any value of μ.

[7] This rule is named after Penrose, who invented it. See [8].

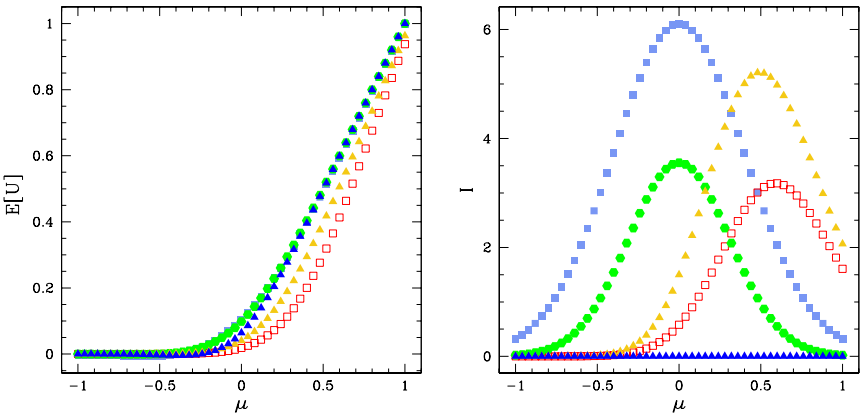


Figure 1: The expected utility (left panel) and the measure of inequality (right panel) as a function of μ for the alternative decision rules. Different point styles designate different rules. Filled light blue squares: SMP; filled green circles: P50 (square root weights); Filled dark blue triangles: SME. Red open squares: Acc. Filled orange triangles: Constitution.

So far we have ranked a few decision rules. But there is still the question, whether we have found the best rules on our desiderata. For the default model, there are a few analytic results in this respect. [4] specify the best decision rule in terms of expected utility – expected utility is maximized under proportional weights and a threshold that depends on μ. An alternative proof for this result is given by [3]. [3] also provide analytical arguments regarding the egalitarian desideratum. They are based upon a relation to Banzhaf voting power (see [6]).

4 Modeling correlations

So far, our results assume that the utilities from proposals are uncorrelated for the different states. But this assumption is not realistic, as we have argued above. Thus the question arises whether the results we obtained for the default model are stable if correlations are taken into account. To address this question, we concentrate on the case of $\mu = 0$.

To model correlations between the states, we assume that $p(\mathbf{v})$ is a multivariate normal. It is fully determined by its covariance matrix. The entries in this matrix are $c_{ij} = E[V_i V_j] - E[V_i]E[V_j]$, where one has to take the expectation value over the probability distribution p in order to calculate $E[\cdot]$. c_{ii} is the variance for the utility from proposals for state i. We assume that it is set to 1 for $i = 1, \ldots, m$ as before.

As there is a lot of freedom to specify the entries c_{ij} and as we are interested in typical behavior that arises from correlations amongst the states, we

CP	type	neg. cross corr.	$\alpha = 0$	$\alpha = .5$	$\alpha = 1$
1	small/large	no	.84/.16	.64/.36	.43/.57
2	South/North	no	.48/.52	.49/.51	.48/.52
3	small/large	yes	.84/.16	.64/.36	.43/.57
4	South/North	yes	.48/.52	.49/.51	.48/.52

Table 1: The parameters used in patterns CP1 to CP4. The numbers in the α-columns are the aggregated weights of the states in each group.

define four *correlation patterns* (CP1 – CP4). Each correlation pattern has one parameter (ϱ) which measures the strength of the correlations. In the covariance matrix, every off-diagonal entry is scaled by ϱ. $\varrho = 0$ means vanishing correlations.

Each correlation pattern groups the states of the EU into two groups of similar (population) size. Patterns CP1 and CP3 consider larger vs. smaller states, and patterns CP2 and CP3 southern vs. northern states (see Table 1 for details).

CP1–2 States i, j from the same group are correlated with strength $c_{ij} = \varrho$. States i, j from different groups are uncorrelated ($c_{ij} = 0$).

CP3–4 States i, j from the same group are correlated with strength $c_{ij} = \varrho$. States i, j from different groups are negatively correlated with $c_{ij} = -\varrho$ ($\varrho > 0$) reflecting the "zero-sum" character of (at least) some of the decision making progresses in the EU: The gains of one states equal the losses of another state.

While the case of zero correlations could be dealt with analytically, the case of non-zero correlations requires the use of computer simulations. They are done as follows. We evaluate the integral Eq. (3) in a Monte Carlo way. As many Monte Carlo integrations, our simulations allow for a dynamical interpretation in terms of an intuitive picture. The picture is as follows: We randomly draw utilities v_i according to our multivariate normal. We determine the votes of the states and check whether the proposal is accepted or rejected. If it is accepted, the respective utilities are distributed to the states, if not, nothing changes. We repeat this $N_{sim} = 10^6$ times. In practice, the procedure converges quickly. In order to get fast random numbers following a multivariate normal, we make a coordinate transformation so that the correlation matrix becomes diagonal.

5 Results

Let us now turn to the outcomes of our simulations which are depicted in Figs. 2 to Fig. 5. The figures exhibit a rich structure and we will restrict ourselves to a discussion of the main results.

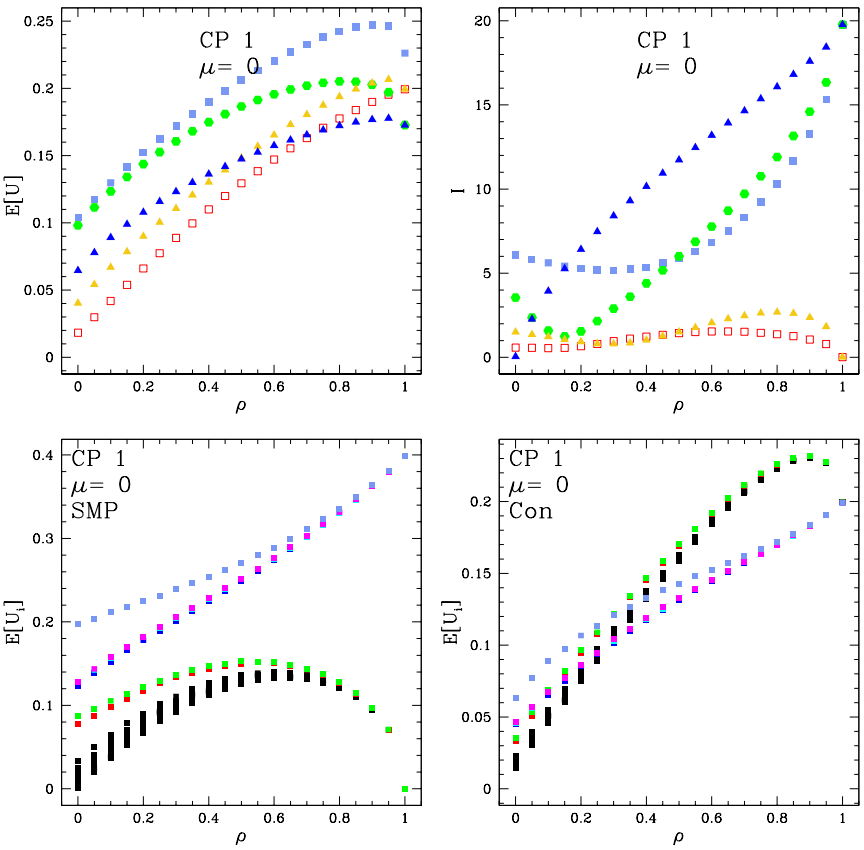


Figure 2: Different characteristics as a function of ϱ for CP1. Left top panel: expected utility $E[U]$. Right top panel: variance $\sigma_w(\{U_i\})$. Point styles in the top panels as in Fig. 1. In the bottom panels we consider one rule and show the expected utilities $E[U_i]$ for every state i. Left bottom panel: SMP. Right bottom panel: Constitution. The point styles are different here: Poland (red), Spain (green), Italy (dark blue), U.K (cyan), France (magenta), Germany (light blue), all other states (black).

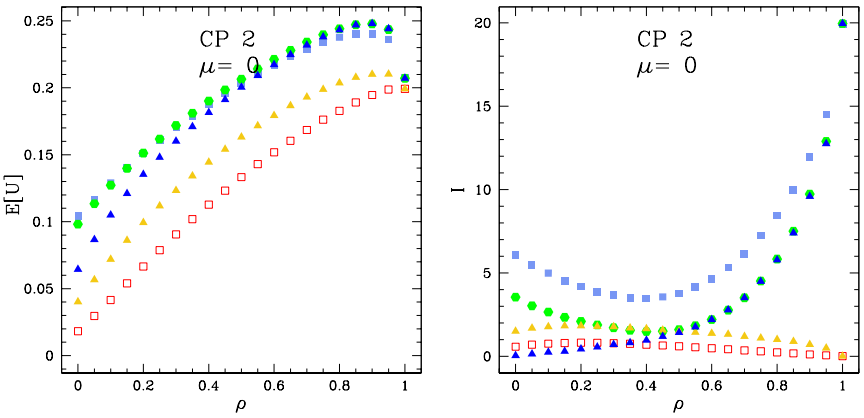


Figure 3: Results for CP2. Point styles in the top panels are as in Fig. 1; point styles in the bottom panels are as in the bottom panels of Fig. 2.

We reserve one figure for each correlation pattern. We show the expected utility $E[U]$ and the standard deviation of $E[U_i]$s as a function of the correlation strength ϱ for our rules (Fig. 2 contains two more panels, which we will consider presently). The leftmost point ($\varrho = 0$) corresponds to the point $\mu = 0$ in Fig. 1. Note that the ranking changes whenever two lines intersect.

The most important question is: Does the ranking of the various decision rules that we obtained for the default model (Fig. 1) change if correlations are taken into account? As Figs. 2 to 5 show, this ranking is fairly stable, as far as the expected utility is concerned. Regarding inequality, there is one significant change: SME, which minimizes inequality under the default model, is worse than the political rules for all correlation patterns and a large range of correlation strengths ϱ. Apart from this, the political rules are better in terms of equality than SMP and P50 both under the default model and if correlations are turned on.

Let us now look at the expected utility of the whole federation, $E[U]$, in more detail and explain some of its features. Whereas, under CP1 and CP2, the expected utilities tend to increase with increasing correlation strength, they decrease under CP3 and CP4. The reason is as follows: The most significant contribution to $E[U]$ comes from proposals from which people from many states benefit. Under CP1 and CP2, there are only positive correlations. The stronger these correlations are, the more likely proposals will benefit people from many states in the federation. Thus, $E[U]$ increases as a function of the correlation strength. This holds quite independently of the respective decision rule. Note, however, that, around $\varrho \approx .9$, things get more complicated, and particularly SME is outrun by the political rules.

Under CP3 and CP4, on the contrary, there are more negative correlations than positive correlations. So typically, if people from one group of states re-

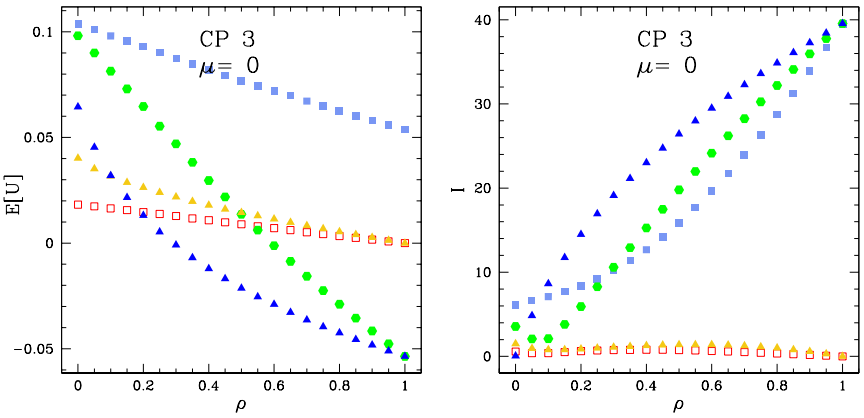


Figure 4: Results for CP3. Point styles as in Fig. 1.

ceive benefits, people from the other group have to pay. Accordingly, proposals from which people from many states take profits, become less likely, and $E[U]$ decreases, as ϱ increases.

The curves for CP3 are very peculiar. SME and P50, which do reasonably well under the default model, are outrun by the political rules for $\varrho \gtrsim .2$ and $\varrho \gtrsim .5$, respectively. At $\varrho = 1$, Acc and Con result in zero expected utility for the federation, whereas SME and P50 produce a negative expected utility. Here is an explanation for this behavior. Under CP3, the large states have the majority of people. However, as Table 1 shows, under SME and P50, the small states hold more weights than the threshold requires. At sufficiently large values of ϱ, the small states are very likely to vote in the same way. Thus, if a proposal is accepted, it will very likely benefit most of the small states. However, because of the anticorrelations in CP3, such a proposal tends to be harmful to people from the larger states. And since there are more people from larger states than from smaller states, $E[U]$ will drop below zero.

The political rules, on the other hand, have higher thresholds of acceptance. A proposal is only accepted, if both large and small states vote for it. As ϱ increases, under CP3, proposals will less likely put benefits on both people from large and from small states. Accordingly, large and small states are less likely to cast the same vote "yes". As a result, proposals are less likely to be accepted, and $E[U]$ approaches 0.

The lesson is, clearly, as follows: If there are two groups that have anticorrelated interests, it is very bad in terms of expected utility to give the smaller group more weights than the threshold requires.

Let's now look at our measure of inequality I in more detail (right panels). Overally, the curves look very similar: As ϱ increases, the measure of inequality for the theoretical rules increases. At $\varrho = 1$, a maximum value of I is reached. The political rules change a bit in terms of I and approach $I = 0$ at $\varrho = 1$.

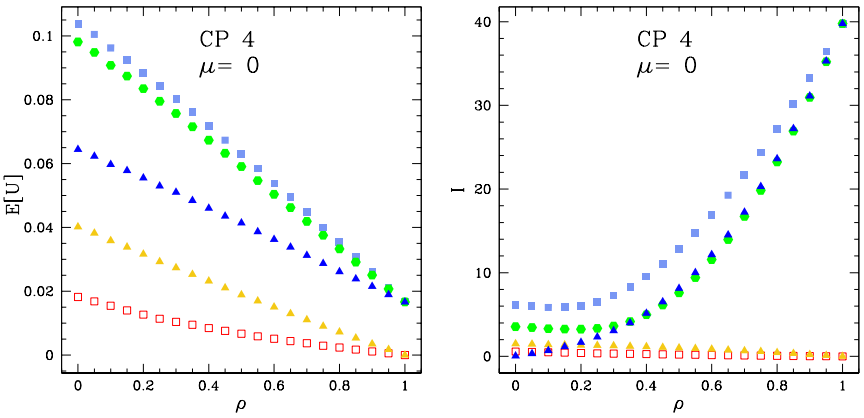


Figure 5: Results for CP4. Point styles as in Fig. 2.

The explanation can be obtained from the bottom panels of Fig. 2, where the $E[U_i]$s are shown as functions of ϱ. We observe that two groups are formed in the following sense: As ϱ increases, the $E[U_i]$-values of states from the same group get closer. However, whereas, under Con, the expected utilities for the groups converge in the limit of $\varrho = 1$, they diverge under SMP. This produces a finite variance. The explanation should come as no surprise, given what we have said before. Under the political rules, states from both groups are needed for acceptance. As a consequence, it will not make a big difference in terms of $E[U_i]$, to which group a state belongs. At $\varrho = 1$, the only proposals that have non-zero probability and that yield acceptance, put the same utility to people from every state. So there is zero variance. Under the theoretical rules, on the other hand, the states from one group will hold more weights than the threshold requires. Accordingly, it makes a big difference to $E[U_i]$, whether state i is a member of this group or not, and the variance approaches a finite value.

Note, that the ranking of the theoretical rules is different for the different correlation patterns, as far as I is concerned. For CP1 and CP3, SME is worst for a large range of ϱ-values, whereas SMP does worst for almost all values of ϱ.

We also obtained results for finite μ-values. Overally, our results do not change much, as we move to finite μ-values of the order of .2 (other values of μ are not realistic).

Again there is the question, whether we have found the best decision rules on our desiderata. From the proof of Theorem 1 in [4] one can construct the rule that maximizes expected utility, even if there are non-zero correlations. Unfortunately, this decision rule is very complicated in general and not suitable for practical purposes. So we think it more appropriate to start with some subset of simple decision rules and to look for the best of them, as we did. But for this it is certainly useful to scan the range of α-values more systematically.

We leave that for future work. Regarding equality, we are not aware of analytic arguments on the best decision rule.

6 Conclusions

The welfarist framework presented in this paper complements the axiomatic approach that has been dominating social choice theory for the last fifty years. We start from sensible desiderata that specify when one rule is better than another one and rank alternative decision rules with respect to these desiderata. Our approach allows us to naturally include "empirical" constraints (such as correlations in the interests of states). Our work profits from the rapid progress in computer science; this helps us to simulate proposals and votes that follow complicated probability distributions. The computational methods we adopt have been used in other disciplines, and we hope to have convinced the reader that they have much to offer to social choice theory as well.

In this paper, we found two main results. *First*, regarding expected utility, we obtain a fairly stable ranking of the decision rules, where SMP does best and the political rules do worst. This is suggested by our simulations of four different correlation patterns with varying correlation strength. We take the stability of the ranking to be good news for the welfarist framework – if the the ranking of the rules ware too sensitive to the correlation pattern and the correlation strength, our account would be useless for policy recommendations. *Second*, the two welfarist principles that we studied in this paper, utilitarianism and egalitarianism, pull in different directions. Whereas political rules with high acceptance thresholds tend to do better in maximizing the expected utility of the federation, theoretical rules are superior in achieving equality. As both principles cannot be satisfied at the same time (at least by the rules studied in this paper), one has to strike a compromise. For vanishing correlations, the rule SME seems to be a reasonable candidate: It yields no inequality at all and is at least better than the political rules in terms of expected utility. Unfortunately, this result does not hold anymore for finite correlations, where SME may produce inequalities that are much larger than the inequalities under political rules.

Another way to compromise between utilitarianism and egalitarianism is to introduce relative weights for these principles. We leave this for future research. We also plan to find realistic correlation models that adequately reflect the correlations of votes found in empirical data.

References

[1] K. J. Arrow. *Social Choice and Individual Values* (2nd edition). Wiley, 1963.

[2] C. Beisbart, L. Bovens and S. Hartmann. A utilitarian assessment of alternative decision rules in the Council of Ministers. *European Union Politics*, 6(4): 395–419 (2005).

[3] C. Beisbart and L. Bovens. Welfarist evaluations of decision rules for boards of Representatives. Working paper, 2006

[4] S. Barberà and M. O. Jackson. On the weights of nations. *Journal of Political Economy*, 114(2): 317–339 (2006).

[5] L. Bovens and S. Hartmann. Welfare, voting and the constitution of a federal assembly. Forthcoming in: M.C. Galavotti, R. Scazzieri and P. Suppes (eds.), *Reasoning, Rationality and Probability*. CSLI Publications, 2006.

[6] D. S. Felsenthal and M. Machover. *The Measurement of Voting Power: Theory and Practice, Problems and Paradoxes*. Edward Elgar, 1998.

[7] D. S. Felsenthal and M. Machover. Enlargement of the EU and weighted voting in its Council of Ministers. URL: http://eprints.lse.ac.uk/archive/00000407/01/euenbook.pdf (2000)

[8] L. S. Penrose. The elementary statistics of majority voting, *Journal of the Royal Statistical Society* 109: 53–57 (1946).

[9] A. Sen. *Collective Choice and Social Welfare* (2nd edition) Holden-Day, 1984.

Claus Beisbart
Institute for Philosophy, Faculty 14
University of Dortmund
D–44221 Dortmund, Germany Email: `Claus.Beisbart@udo.edu`

Stephan Hartmann
Department of Philosophy, Logic and Scientific Method
London School of Economics and Political Science
Houghton Street
London WC2A 2AE, UK Email: `S.Hartmann@lse.ac.uk`

Finding leximin-optimal solutions using constraint programming: new algorithms and their application to combinatorial auctions

Sylvain Bouveret and Michel Lemaître

Abstract

We study the problem of computing a leximin-optimal solution of a constraint network. This problem is highly motivated by fairness and efficiency requirements in many real-world applications implying human agents. We compare several generic algorithms which solve this problem in a constraint programming framework. The second one is entirely original, and the other ones are partially based on existing works adapted to fit with this problem. These algorithms are tested on combinatorial auctions instances.

1 Introduction

Many advances have been done in recent years in modeling and solving combinatorial problems with constraint programming (CP). These advances concern, among others, the ability of this framework to deal with human reasoning schemes, such as, for example, the expression of preferences with soft constraints. However, one aspect of importance has only received little attention in the constraints community to date: the way to handle fairness requirements in multiagent combinatorial problems.

The seek for fairness stands as a subjective but strong requirement in a wide set of real-world problems implying human agents. It is particularly relevant in crew or worker timetabling and rostering problems, or the optimization of long and short-term planning for firemen and emergency services. Fairness is also ubiquitous in multiagent resource allocation problems, like, among others, bandwidth allocation among network users, fair share of airspace and airport resources among several airlines or Earth observing satellite scheduling and sharing problems [11].

In spite of the wide range of problems concerned by fairness issues, it often lacks a theoretical and generic approach. In many Constraint Programming and Operational Research works, fairness is only enforced by specific heuristic local choices guiding the search towards supposed equitable solutions. However, a few works may be cited for their approach of this fairness requirement. [11] make use of an Earth observation satellite scheduling and sharing problem to investigate three ways of handling fairness among agents in the context of constraint satisfaction. More recently [18] proposed a new constraint based on statistics, which enforces the relative balance of a given set of variables, and can

possibly be used to ensure a kind of equity among a set of agents. Equity is also studied in Operational Research, with for example [17], who investigate a way of solving linear programs by aggregating multiple criteria using an Ordered Weighted Average Operator (OWA) [22]. Depending on the weights used in the OWA, this kind of aggregators can provide equitable compromises.

Microeconomy and Social Choice theory provide an important literature on fairness in collective decision making. From this theoretical background we borrow the idea of representing the agents preferences by *utility* levels, and we adopt the *leximin* preorder on utility profiles for conveying the fairness and efficiency requirements.

Apart from the fact that it conveys and formalizes the concept of equity in multiagent contexts, the leximin preorder is also a subject of interest in other contexts, such as fuzzy CSP [6], and symmetry-breaking in constraint satisfaction problems [7].

This contribution is organized as follows. Section 2 gives a minimal background in social choice theory and justifies the interest of the leximin preorder as a fairness criterion. Section 3 defines the search for leximin-optimality in a constraint programming framework. The main contribution of this paper is Section 4, which presents three algorithms for computing leximin-optimal solutions, the first one being entirely original, and the other ones adapted from existing works. The proposed algorithms have been implemented and tested within a constraint programming system. Section 5 presents an experimental comparison of these algorithms[1].

2 Background on social choice theory

We first introduce some notations. Calligraphic letters (*e.g.* $\mathcal{X}$) will stand for sets. Vectors will be written with an arrow (*e.g.* $\overrightarrow{x}$), or between brackets (*e.g.* $\langle x_1, \ldots, x_n \rangle$). $f(\overrightarrow{x})$ will be used as a shortcut for $\langle f(x_1), \ldots, f(x_n) \rangle$. Vector $\overrightarrow{x}^{\uparrow}$ will stand for the vector composed by each element of $\overrightarrow{x}$ rearranged in increasing order. We will write $x_i^{\uparrow}$ for the i^{th} component of vector $\overrightarrow{x}^{\uparrow}$. Finally, the interval of integers between k and l will be written $[\![k, l]\!]$.

2.1 Collective decision making and welfarism

Let $\mathcal{N}$ be a set of n agents, and $\mathcal{S}$ be a set of admissible alternatives concerning all of them, among which a benevolent arbitrator has to choose one. The most classical model describing this situation is *welfarism* (see *e.g.* [9, 15]): the choice of the arbitrator is made on the basis of the utility levels enjoyed by the individual agents and on those levels only. Each agent $i \in \mathcal{N}$ has an individual utility function u_i that maps each admissible alternative $s \in \mathcal{S}$ to a numerical

[1] A similar paper is going to appear in the proceedings of IJCAI'07 with the section 5 based on a different application.

index $u_i(s)$. We make here the classical assumption that the individual utilities are comparable between the agents[2]. Therefore each alternative s can be attached to a single *utility profile* $\langle u_1(s), \ldots, u_n(s)\rangle$. According to welfarism, comparing two alternatives is performed by comparing their respective utility profiles.

A standard way to compare individual utility profiles is to aggregate each of them into a *collective utility* index, standing for the collective welfare of the agents community. If g is a well-chosen aggregation function, we thus have a collective utility function uc that maps each alternative s to a collective utility level $uc(s) = g(u_1(s), \ldots, u_n(s))$. An optimal alternative is one of those maximizing the collective utility.

2.2 The leximin order as a fairness and efficiency criterion

The main difficulty of equitable decision problems is that we have to reconcile the contradictory wishes of the agents. Since generally no solution fully satisfies everyone, the aggregation function g must lead to fair and Pareto-efficient[3] compromises.

The problem of choosing the right aggregation function g is far beyond the scope of this paper. We only describe the two classical ones corresponding to two opposite points of view on social welfare[4]: classical utilitarianism and egalitarianism. The rule advocated by the defenders of classical utilitarianism is that the best decision is the one that maximizes the sum of individual utilities (thus corresponding to $g = +$). However this kind of aggregation function can lead to huge differences of utility levels among the agents, thus ruling out this aggregator in the context of equitable decisions. From the egalitarian point of view, the best decision is the one that maximizes the happiness of the least satisfied agent (thus corresponding to $g = \min$). Whereas this kind of aggregation function is particularly well-suited for problems in which fairness is essential, it has a major drawback, due to the idempotency of the min operator, and known as "drowning effect" in the community of fuzzy CSP (see *e.g.*[4]). Indeed, it leaves many alternatives indistinguishable, such as for example the ones with utility profiles $\langle 0, \ldots, 0\rangle$ and $\langle 1000, \ldots, 1000, 0\rangle$, even if the second one appears to be much better than the first one. In other words, the min aggregation function can lead to non Pareto-optimal decisions, which is not desirable.

The leximin preorder is a well-known refinement of the order induced by the min function that overcomes this drawback. It is classically introduced in the social choice literature (see [15]) as the social welfare ordering that reconcile egalitarianism and Pareto-efficiency, and also in fuzzy CSP [6]. It is defined as follows:

[2] In other words, they are expressed using a common utility scale.

[3] A decision is Pareto-efficient if and only if we cannot strictly increase the satisfaction of an agent unless we strictly decrease the satisfaction of another agent. Pareto-efficiency is generally taken as a basic postulate in collective decision making.

[4] Compromises between these two extremes are possible. See *e.g.* [16, page 68] or [22] (*OWA aggregators*).

Definition 1 (leximin preorder [15]) *Let $\overrightarrow{x}$ and $\overrightarrow{y}$ be two vectors of $\mathbb{N}^n$. $\overrightarrow{x}$ and $\overrightarrow{y}$ are said leximin-indifferent (written $\overrightarrow{x} \sim_{leximin} \overrightarrow{y}$) if and only if $\overrightarrow{x}^{\uparrow} = \overrightarrow{y}^{\uparrow}$. The vector $\overrightarrow{y}$ is leximin-preferred to $\overrightarrow{x}$ (written $\overrightarrow{x} \prec_{leximin} \overrightarrow{y}$) if and only if $\exists i \in [\![0, n-1]\!]$ such that $\forall j \in [\![1, i]\!]$, $x_j^{\uparrow} = y_j^{\uparrow}$ and $x_{i+1}^{\uparrow} < y_{i+1}^{\uparrow}$. We write $\overrightarrow{x} \preceq_{leximin} \overrightarrow{y}$ for $\overrightarrow{x} \prec_{leximin} \overrightarrow{y}$ or $\overrightarrow{x} \sim_{leximin} \overrightarrow{y}$. The binary relation $\preceq_{leximin}$ is a total preorder.*

In other words, the leximin preorder is the lexicographic preorder over ordered utility vectors. For example, we have $\langle 4, 1, 5, 1\rangle \prec_{leximin} \langle 2, 2, 1, 2\rangle$.

A known result is that no collective utility function can represent the leximin preorder[5], unless the set of possible utility profiles is finite. In this latter case, it can be represented by the following non-linear functions: $g_1 : \overrightarrow{x} \mapsto -\sum_{i=i}^{n} n^{-x_i}$ (adapted for leximin from a remark in [7]) and $g_2 : \overrightarrow{x} \mapsto -\sum_{i=1}^{n} x_i^{-q}$, where $q > 0$ is large enough [15]. The major drawback of using this kind of representation is that it rapidly becomes unreasonable to use it when the upper bound of the possible values of $\overrightarrow{x}$ increases. Moreover, it hides the semantics of the leximin preorder, and hinders the computational benefits we could possibly take advantage of.

In the following, we will use the leximin preorder as a criterion for ensuring fairness and Pareto-efficiency, and we will seek the non-dominated solutions in the sense of the leximin preorder. Those solutions will be called leximin-optimal. This problem will be expressed in the next section in a CP framework.

3 Leximin and Constraint programming

The constraint programming framework is an effective and flexible tool for modeling and solving many different combinatorial problems such as planning and scheduling problems, resource allocation problems, or configuration problems. This paradigm is based on the notion of *constraint network* [14]. A constraint network consists of a set of variables $\mathcal{X} = \{x_1, \ldots, x_p\}$, a set of associated domains $\mathcal{D} = \{d_{x_1}, \ldots, d_{x_p}\}$, d_{x_i} being the set of possible values for x_i, and a set of constraints $\mathcal{C}$, where each $C \in \mathcal{C}$ specifies a set of allowed tuples $R(C)$ over a set of variables $X(C)$. We will also suppose that all the domains are in $\mathbb{N}$, and use the following notations: $\underline{x} = \min(d_x)$ and $\overline{x} = \max(d_x)$.

An instantiation v of a set $\mathcal{S}$ of variables is a function that maps each variable $x \in S$ to a value $v(x)$ of its domain d_x. If $S = \mathcal{X}$, this instantiation is said to be complete, otherwise it is partial. If $\mathcal{S}' \subsetneq \mathcal{S}$, the projection of an instantiation of $\mathcal{S}$ over $\mathcal{S}'$ is the restriction of this instantiation to $\mathcal{S}'$ and is written $v_{\downarrow \mathcal{S}'}$. An instantiation is said to be consistent if and only if it satisfies all the constraints. A complete consistent instantiation of a constraint network is called a solution. The set of solutions of $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is written $sol(\mathcal{X}, \mathcal{D}, \mathcal{C})$.

Given a constraint network, the problem of determining whether it has a solution is called a Constraint Satisfaction Problem (CSP) and is NP-complete.

[5] In other words there is no g such that $\overrightarrow{x} \preceq_{leximin} \overrightarrow{y} \Leftrightarrow g(\overrightarrow{x}) \leq g(\overrightarrow{y})$. See [15].

The CSP can be classically adapted to become an optimization problem in the following way. Given a constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ and an *objective* variable $o \in \mathcal{X}$, find the value M of d_o such that $M = \max\{v(o) \mid v \in sol(\mathcal{X}, \mathcal{D}, \mathcal{C})\}$. We will write $max(\mathcal{X}, \mathcal{D}, \mathcal{C}, o)$ for the subset of those solutions that maximize the objective variable o.

Expressing a collective decision making problem with a numerical collective utility criterion as a CSP with objective variable is straightforward: consider the collective utility as the objective variable, and link it to the variables representing individual utilities with a constraint. However this cannot directly encode our problem of computing a leximin-optimal solution, which is a kind of multicriteria optimization problem. We introduce formally the MaxLeximinCSP problem as follows :

Definition 2 (Problem MaxLeximinCSP)
Input: *a constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C})$; a vector of variables $\overrightarrow{u} = \langle u_1, \ldots, u_n \rangle \in \mathcal{X}^n$, called the* objective vector.
Output: *"Inconsistent" if $sol(\mathcal{X}, \mathcal{D}, \mathcal{C}) = \emptyset$. Otherwise a solution $\widehat{v}$ such that $\forall v \in sol(\mathcal{X}, \mathcal{D}, \mathcal{C})$, $v(\overrightarrow{u}) \preceq_{leximin} \widehat{v}(\overrightarrow{u})$.*

We describe in the next section several generic constraint programming algorithms that solve this problem. The second one is entirely original, and the other ones are based on existing works that are adapted to fit with our problem.

4 Proposed algorithms

4.1 Using a sorting constraint

Our first algorithm is directly based on the definition 1 of the leximin preorder, which requires to sort the vectors to be compared before performing a lexicographic comparison. We can therefore introduce, using additional variables, the sorted version of the objective vector. This can be done naturally in the CP paradigm by introducing a vector of variables $\overrightarrow{y}$ and enforcing the constraint $\mathbf{Sort}(\overrightarrow{u}, \overrightarrow{y})$ which is defined as follows:

Definition 3 (Constraint Sort) *Let $\overrightarrow{x}$ and $\overrightarrow{x}'$ be two vectors of variables of the same length, and v be an instantiation. The constraint $\mathbf{Sort}(\overrightarrow{x}, \overrightarrow{x}')$ holds on the set of variables being either in $\overrightarrow{x}$ or in $\overrightarrow{x}'$, and is satisfied by v if and only if $v(\overrightarrow{x}')$ is the sorted version of $v(\overrightarrow{x})$ in increasing order.*

This constraint has been particularly studied in two works, which both introduce a filtering algorithm for enforcing bound consistency on this constraint. The first algorithm comes from [1] and runs in $O(n \log n)$ (n being the size of $\overrightarrow{x}$). [13] designed a simpler algorithm that runs in $O(n)$ plus the time required to sort the interval endpoints of $\overrightarrow{x}$, which can asymptotically be faster than $O(n \log n)$.

The algorithm 1 intuitively works as follows : having introduced the sorted version $\overrightarrow{y}$ of the objective vector $\overrightarrow{u}$, it successively maximizes the components of this vector, provided that the leximin-optimal solution is the solution that maximizes y_1, and, given this maximal value, maximizes y_2, and so on until y_n.

Algorithm 1: Solving the MaxLeximinCSP using a sorting constraint.

input : A const. network $(\mathcal{X}, \mathcal{D}, \mathcal{C})$; $\langle u_1, \ldots, u_n \rangle \in \mathcal{X}^n$
output: A solution to the MaxLeximinCSP problem

if solve$(\mathcal{X}, \mathcal{D}, \mathcal{C})$ ="*Inconsistent*" **return** *"Inconsistent"*;
$\mathcal{X}' \leftarrow \mathcal{X} \cup \{y_1, \ldots, y_n\}$;
$\mathcal{D}' \leftarrow \mathcal{D} \cup \{d_{y_1}, \ldots, d_{y_n}\}$ with $d_{y_i} = [\![\min_j(\underline{u_j}), \max_j(\overline{u_j})]\!]$;
$\mathcal{C}' \leftarrow \mathcal{C} \cup \{\mathbf{Sort}(\overrightarrow{u}, \overrightarrow{y})\}$;
for $i \leftarrow 1$ **to** n **do**
  $\widehat{v}_{(i)} \leftarrow$ **maximize**$(\mathcal{X}', \mathcal{D}', \mathcal{C}', y_i)$;
  $d_{y_i} \leftarrow \{\widehat{v}_{(i)}(y_i)\}$;
return $\widehat{v}_{(n)\downarrow\mathcal{X}}$;

In the algorithm 1 (and in the following ones also), the functions **solve** and **maximize** (the detail of which is the concern of solving techniques for constraints satisfaction problems) respectively return one solution $v \in sol(\mathcal{X}, \mathcal{D}, \mathcal{C})$ (or "Inconsistent" if such a solution does not exist), and an optimal solution $\widehat{v} \in max(\mathcal{X}_i, \mathcal{D}_i, \mathcal{C}_i, y_i)$ (or "Inconsistent" if $sol(\mathcal{X}_i, \mathcal{D}_i, \mathcal{C}_i) = \emptyset$). We assume – contrary to usual constraint solvers – that these two functions do not modify the input constraint network.

Proposition 1 *If the two functions* **maximize** *and* **solve** *are both correct and both halt, then algorithm 1 halts and solves the* MaxLeximinCSP *problem.*

Proof: If $sol(\mathcal{X}, \mathcal{D}, \mathcal{C}) = \emptyset$ and if **solve** is correct, then algorithm 1 obviously returns "Inconsistent". We will suppose in the following that $sol(\mathcal{X}, \mathcal{D}, \mathcal{C}) \neq \emptyset$ and we will use the following notations: $\mathcal{S}_i$ and $\mathcal{S}'_i$ are the sets of solutions of $(\mathcal{X}', \mathcal{D}', \mathcal{C}')$ respectively at the beginning and at the end of iteration i.

We have obviously $\forall i \in [\![1, n-1]\!]$ $\mathcal{S}_{i+1} = \mathcal{S}'_i$, which proves that if $\mathcal{S}_i \neq \emptyset$, then the call to **maximize** at line 1 does not return "Inconsistent", and $\mathcal{S}_{i+1} \neq \emptyset$. Thus $\widehat{v}_{(n)}$ is well-defined, and obviously $(\widehat{v}_{(n)})_{\downarrow\mathcal{X}}$ is a solution of $(\mathcal{X}, \mathcal{D}, \mathcal{C})$.

We note $\widehat{v} = \widehat{v}_{(n)}$ the instantiation computed by the last **maximize** in algorithm 1. Suppose that there is an instantiation $v \in sol(\mathcal{X}, \mathcal{D}, \mathcal{C})$ such that $\widehat{v}(\overrightarrow{u}) \prec_{leximin} v(\overrightarrow{u})$. We define v^+ the extension of v that instantiates each y_i to $v(\overrightarrow{u})^{\uparrow}_i$. Then, due to constraint **Sort**, $\widehat{v}(\overrightarrow{y})$ and $v^+(y)$ are the respective sorted version of $\widehat{v}(\overrightarrow{u})$ and $v^+(u)$. Following definition 1, there is an $i \in [\![0, n-1]\!]$ such that $\forall j \in [\![1, i]\!]$, $\widehat{v}(y_j) = v^+(y_j)$ and $\widehat{v}(y_{i+1}) < v^+(y_{i+1})$. Due to line 1, we have $\widehat{v}(y_{i+1}) = \widehat{v}_{(n)}(y_{i+1}) = \widehat{v}_{(i+1)}(y_{i+1})$. Thus v^+ is a solution in $max(\mathcal{X}', \mathcal{D}', \mathcal{C}', y_{i+1})$ with objective value $v^+_{(i+1)}(y_{i+1})$ strictly greater than $\widehat{v}_{(i+1)}(y_{i+1})$, which contradicts the hypothesis about **maximize**. ■

4.2 Using a cardinality combinator

Our second algorithm is based on an alternative definition of the sorting of the objective vector. In fact, it can be noticed that, given two vectors of numbers $\overrightarrow{x}$ and $\overrightarrow{x}'$, $\overrightarrow{x}'$ is the sorted version of $\overrightarrow{x}$ in increasing order if and only if for all i, x'_i is the maximal value such that at least $n-i+1$ values from vector $\overrightarrow{x}$ are greater than or equal to x'_i.

Like the first algorithm, this algorithm works by successively computing the sorted components of the leximin-optimal objective vector, but contrary to the first one, this new algorithm does not explicitely introduce the sorted version of the objective vector. This new algorithm informally works as follows. It first computes the maximal value y_1 such that there is a solution v with $\forall i$, $y_1 \leq v(u_i)$, or in other words $\sum_i (y_1 \leq v(u_i)) = n$, where by convention the value of $(y_1 \leq v(u_i))$ is 1 if the inequality is satisfied and 0 otherwise[6]. Then, after having fixed this value for y_1, it computes the maximal value y_2 such that there is a solution v with $\sum_i (y_2 \leq v(u_i)) \geq n-1$, and so on until the maximal value y_n such that there is a solution v with $\sum_i (y_n \leq v(u_i)) \geq 1$.

To enforce the constraint on the y_i, we make use of the meta-constraint **AtLeast**, derived from a *cardinality combinator* introduced by [21], and present in most CP systems:

Definition 4 (Meta-constraint AtLeast) *Let Γ be a set of p constraints, and $k \in [\![1, p]\!]$ be an integer. The meta-constraint* **AtLeast**(Γ, k) *holds on the union of the scopes of the constraints in Γ, and allows a tuple if and only if at least k constraints from Γ are satisfied.*

Due to its genericity, this meta-constraint cannot provide very efficient filtering procedures. Fortunately, in our case where each constraint in Γ is of the form $x_i \geq y$, bound-consistency can be enforced using algorithm 2.

Algorithm 2: Enforcing bound-consistency on the **AtLeast** meta-constraint with linear constraints.

input : A vector of variables $\langle x_1, \ldots, x_n \rangle$, a variable y, an integer $k \leq n$.
output: The domain reductions of $\langle x_1, \ldots, x_n \rangle$ and y to enforce bound consistency on **AtLeast**$(\{x_1 \geq y, \ldots, x_n \geq y\}, k)$, or "Inconsistent"

1 $\overline{y} \leftarrow (\sup(\overrightarrow{x}))^{\uparrow}_{n-k+1}$; `/* where` $\sup(\overrightarrow{x}) = \langle \overline{x_1}, \ldots, \overline{x_n} \rangle$ `*/`
2 **if** $\sum_i (\overline{x_i} < \underline{y}) > n-k$ **return** *"Inconsistent"*;
3 **if** $\sum_i (\overline{x_i} < \underline{y}) = n-k$
4 ⌊ **forall** i *such that* $\overline{x_i} \geq \underline{y}$ **do** $\underline{x_i} \leftarrow \max(\underline{y}, \underline{x_i})$

This algorithm runs in $O(n)$, since the selection of the $n-k+1^{\text{st}}$ lowest value of $\sup(\overrightarrow{x})$ can be done in $O(n)$ [2]. We can notice that this algorithm is well-suited for event-based implementation of constraint programming: in case of an update of one of the $\overline{x_i}$, only line 1 needs to be run ; in case of an update

[6]This convention is inspired by the constraint modeling language OPL [20].

of $\underline{y}$, only lines 2 and 3 need to be run ; any other update do not need the algorithm to be run. The procedure can also benefit from storing the ordered vector $(\sup(\overrightarrow{x}))^{\uparrow}$ and updating it when one of the $\overline{x_i}$ changes. By doing so, we can access $(\sup(\overrightarrow{x}))^{\uparrow}_{n-k+1}$ in $O(1)$.

It can also be noticed that since all of the constraints of Γ are linear, the meta-constraint **AtLeast** can be expressed using a set of linear constraints, therefore allowing our algorithm to be processed with a linear solver. The classical idea [8, p.11] is to express our constraint **AtLeast** by introducing n 0–1 variables $\{\delta_1, \ldots, \delta_n\}$, and a set of linear constraints $\{x_1 + \delta_1 \overline{y} \geq y, \ldots, x_n + \delta_n \overline{y} \geq y, \sum_{i=1}^{n} \delta_i \leq n - k\}$.

This second approach is presented in algorithm 3.

Algorithm 3: Solving the MaxLeximinCSP using a cardinality constraint.

input : A const. network $(\mathcal{X}, \mathcal{D}, \mathcal{C})$; $\langle u_1, \ldots, u_n \rangle \in \mathcal{X}^n$
output: A solution to the MaxLeximinCSP problem

1 **if solve**$(\mathcal{X}, \mathcal{D}, \mathcal{C})$ ="*Inconsistent*" **return** *"Inconsistent"*;
2 $(\mathcal{X}_0, \mathcal{D}'_0, \mathcal{C}_0) \leftarrow (\mathcal{X}, \mathcal{D}, \mathcal{C})$;
3 **for** $i \leftarrow 1$ **to** n **do**
4 $\quad \mathcal{X}_i \leftarrow \mathcal{X}_{i-1} \cup \{y_i\}$;
5 $\quad \mathcal{D}_i \leftarrow \mathcal{D}'_{i-1} \cup \{d_{y_i}\}$ with $d_{y_i} = [\![\min_j(\underline{u_j}), \max_j(\overline{u_j})]\!]$;
6 $\quad \mathcal{C}_i \leftarrow \mathcal{C}_{i-1} \cup \{$**AtLeast**$(\{y_i \leq u_1, \ldots, y_i \leq u_n\}, n - i + 1)\}$;
7 $\quad \widehat{v}_{(i)} \leftarrow$ **maximize**$(\mathcal{X}_i, \mathcal{D}_i, \mathcal{C}_i, y_i)$;
8 $\quad \mathcal{D}'_i \leftarrow \mathcal{D}_i$ with $d_{y_i} \leftarrow \{\widehat{v}_{(i)}(y_i)\}$;
9 **return** $\widehat{v}_{(n)\downarrow\mathcal{X}}$;

	a_1	a_2	a_3
o_1	3	3	3
o_2	5	9	7
o_3	7	8	1

The following example illustrates the behavior of the algorithm. It is a simple resource allocation problem, where 3 objects must be allocated to 3 agents, with the following constraints: each agent must get one and only one object, and one object cannot be allocated to more than one agent (i.e. a perfect matching agent/objects). A utility is associated with each pair (agent,object) with respect to the array above.

This problem has 6 feasible solutions (one for each permutation of $[\![1, 3]\!]$), producing the 6 utility profiles shown in the columns of the array aside.

	p_1	p_2	p_3	p_4	p_5	p_6
u_1	3	3	5	5	7	7
u_2	9	8	3	8	3	9
u_3	1	7	1	3	7	3

The algorithm runs in 3 steps: **Step 1:** After having introduced one variable y_1, we look for the maximal value $\widehat{y_1}$ of y_1 such that each (**at least** 3) agent gets at least y_1. We find $\widehat{y_1} = 3$. The variable y_1 is fixed to this value, implicitly removing profiles p_1 and p_3. **Step 2:** After having introduced one variable y_2, we look for the maximal value $\widehat{y_2}$ of y_2 such that **at least** 2 agents get at least y_2. We find $\widehat{y_2} = 7$. The variable y_2 is fixed to this value, implicitly removing profile p_4. **Step 3:** After having introduced one variable y_3, we look for the maximal value $\widehat{y_3}$ of y_3 such that **at least** 1 agent gets at least y_3. We

find $\widehat{y_3} = 9$. Only one instantiation maximizes y_3: p_6. Finally, the returned leximin-optimal allocation is: $a_1 \leftarrow o_3$, $a_2 \leftarrow o_2$ and $a_3 \leftarrow o_1$.

Proposition 2 *If the two functions* **maximize** *and* **solve** *are both correct and both halt, then algorithm 3 halts and solves the* MaxLeximinCSP *problem.*

The complete proof of this proposition can be found in the article published in the proceedings of IJCAI'07. We just give here a proof sketch.

Proof sketch: The proposition can be proved using the following steps.

- We first prove the initial remark : if $\overrightarrow{x}$ is a vector of size n, then at least $n-i+1$ components of $\overrightarrow{x}$ are greater than or equal to $x_i^{\uparrow}$.
- Then we must prove that if the initial constraint network has a solution then $\widehat{v}_{(n)}$ is well-defined and not equal to "Inconsistent".
- We then prove that $\widehat{v}_{(n)}(\overrightarrow{y})$ is equal to $\widehat{v}_{(n)}(\overrightarrow{u})^{\uparrow}$ if $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ has a solution.
- By putting things together, we can finally prove that $\widehat{v}_{(n)\downarrow\mathcal{X}}$ is really the leximin-optimal solution, using the fact that if there was a better solution (in the sense of the leximin preorder), the call to **maximize** at some iteration would have eliminated the solution actually returned by the algorithm. ■

4.3 Using a multiset ordering constraint

Our third algorithm computing a leximin-optimal solution is probably the most intuitive one. This algorithm proceeds in a pseudo branch and bound manner: it computes a first solution, then it tries to improve it by specifying that the next solution has to be better (in the sense of the leximin preorder) than the current one, and so on until the constraint network becomes inconsistent. This approach is based on the following constraint:

Definition 5 (Constraint Leximin) *Let $\overrightarrow{x}$ be a vector of variables, $\overrightarrow{\lambda}$ be a vector of integers, and v be an instantiation. The constraint* **Leximin**$(\overrightarrow{\lambda}, \overrightarrow{x})$ *holds on the set of variables belonging to $\overrightarrow{x}$, and is satisfied by v if and only if $\overrightarrow{\lambda} \prec_{leximin} \overrightarrow{v(x)}$.*

Although this constraint does not exist in the literature, the work of [7] introduces an algorithm for enforcing generalized arc-consistency on a quite similar constraint: the multiset ordering constraint, which is, in the context of multisets, the equivalent of a leximax[7] constraint on vectors of variables. At the price of some slight modifications, the algorithm they introduce can easily be used to enforce the latter constraint **Leximin**.

Proposition 3 *If the function* **solve** *is correct and halts, then algorithm 4 halts and solves the* MaxLeximinCSP *problem.*

The proof is rather straightforward, so we omit it.

[7] The leximax is based on an increasing reordering of the values, instead of a decreasing one for leximin.

Algorithm 4: Solving the MaxLeximinCSP using a constraint **Leximin**.

input : A const. network $(\mathcal{X}, \mathcal{D}, \mathcal{C})$; $\langle u_1, \ldots, u_n \rangle \in \mathcal{X}^n$
output: A solution to the MaxLeximinCSP problem

$\widehat{v} \leftarrow$ **null**; $v \leftarrow$ **solve**$(\mathcal{X}, \mathcal{D}, \mathcal{C})$;
while $v \neq$ "$Inconsistent$" **do**
  $\widehat{v} \leftarrow v$;
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{$**Leximin**$(\widehat{v}(\overrightarrow{u}), \overrightarrow{u})\}$;
  $v \leftarrow$ **solve**$(\mathcal{X}, \mathcal{D}, \mathcal{C})$;
if $\widehat{v} \neq$ **null then return** $\widehat{v}$ **else return** *"Inconsistent"*;

4.4 Other approaches

In the context of fuzzy constraints, two algorithms dedicated to the computation of leximin-optimal solutions have been published by [4]. These algorithms work by enumerating, at each step, all the subsets of fuzzy constraints (corresponding to our agents) having a property connected to the notion of consistency degree.

[5, p. 162] describes two very simple algorithms for solving the closely related "Lexicographic Max-Ordering" problem (in our terms, finding the "leximax-optimal"). They however do not seem realistic in the context of combinatorial problems, since they are based on an enumeration of all utility profiles.

5 Experimental results

Combinatorial auctions[3, 19] – auctions in which bidders place unrestricted bids for bundles of goods – are subject of increasing study in the recent years. Their central problem is the *Winner Determination Problem* (WDP), which has been extensively studied. It definitely corresponds to an utilitarian point of view, namely maximizing the revenue of the auctioneer, which is the sum of the selected bids, whoever receive them. Even if fairness does not seem to be a relevant issue in combinatorial auctions, the WDP can however inspire us a fair resource allocation problem with indivisible goods, where the agents express their preferences over bundles of items:

Definition 6 (Fair CA instance) *Given a set of agents $\mathcal{N}$ and a set of objects $\mathcal{O}$, a* bid *b is a triple $\langle s(b), p(b), a(b) \rangle \in 2^{\mathcal{O}} \times \mathbb{N} \times \mathcal{N}$ (a bundle of objects, a price and an agent). Given a set of non-intersecting bids $\mathcal{W}$ and an agent i, the utility of i regarding $\mathcal{W}$ is $u_i(\mathcal{W}) = \sum\{p(b) \mid b \in \mathcal{W}$ and $a(b) = i\}$. A* fair combinatorial auctions instance *is defined as follows:*
Input: *A set of n agents $\mathcal{N}$, a set of objects $\mathcal{O}$ and a set of bids $\mathcal{B}$.*
Output: *A set of non-intersecting bids $\mathcal{W} \subseteq \mathcal{B}$ such that there is no $\mathcal{W}' \subseteq \mathcal{B}$ with $\langle u_1(\mathcal{W}'), \ldots, u_n(\mathcal{W}') \rangle \succ_{\text{leximin}} \langle u_1(\mathcal{W}), \ldots, u_n(\mathcal{W}) \rangle$.*

The algorithms 3, 1, 4 and the first algorithm from [4] have been implemented and tested on CA instances using the constraint programming tool

kind	Algorithm 1 (**Sort**)				Algorithm 3 (**AtLeast**)			
	avg	min	max	N%	avg	min	max	N%
1	122.6	4.5	482.7	100%	**121.2**	5.1	470.1	100%
2	394.8	162.5	600	80%	**158.6**	82.8	350.6	100%
3	**480**	66	600	30%	480.8	64	600	30%
4	600	600	600	0%	**506.6**	196.2	600	30%
5	12.1	5.6	23	100%	**4.8**	2.6	7.9	100%
6	78.8	47.9	156.4	100%	**68.5**	44.1	131.6	100%

Algorithm 4 (**Leximin**)				Algorithm from [4]				Sum-optimal			
avg	min	max	N%	avg	min	max	N%	avg	min	max	N%
380	42.4	600	60%	488	32.6	600	20%	485	158	600	40%
479	161	600	50%	600	600	600	0%	485	158	600	40%
600	600	600	0%	600	600	600	0%	600	600	600	0%
600	600	600	0%	600	600	600	0%	600	600	600	0%
62.4	26.4	128	100%	600	600	600	0%	19.4	2.1	49.1	100%
94.7	26.4	203	100%	600	600	600	0%	18.8	3.7	45.7	100%

Table 1: CPU times (in sec.) and percentage of instances solved within 10 minutes (each algorithm tested on 10 instances of each kind).

Choco [10]. The test instances have been generated using CATS [12], which aims at making realistic and economically motivated bids for combinatorial auctions, *e.g* by simulating some kind of relations such as substitutabilities and complementarities between the goods. We used six different kind of instances (see [12] for the definitions of the different kinds of relationships between the goods): (1) 5 agents, 200 objects, 200 bids, arbitrary relationships, (2) 30 agents, 200 objects, 200 bids, arbitrary relationships, (3) 5 agents, 200 objects, 200 bids, regions-based relationships, (4) 30 agents, 200 objects, 200 bids, regions-based relationships, (5) 20 agents, 200 objects, 100 bids, arbitrary relationships, (6) 20 agents, 50 objects, 200 bids, arbitrary relationships.

The running times of the tests are shown in table 1. They show that the most efficient algorithm on these kinds of instances is algorithm 3, followed by algorithm 1. Conversely, algorithm 4 and the algorithm from [4] are inefficient. It is interesting to notice that, whereas the algorithms 1 and 4 are affected by the increasing of the number of agents (see *e.g* kinds 1 and 2), the running time of algorithm 3 only slightly increases (in spite of the fact that the number of calls to **maximize** is exactly the number of agents). For each instance, we also solved the WDP using our contraint programming model, which is – due to the genericity of the CP framework – far less efficient than the dedicated algorithms. It is surprising to see that solving the WDP using our CP model requires much more time than solving the MaxLeximinCSP with algorithm 3. This is rather counterintuitive since, all other parameters being equal, the running time tends to decrease with the number of agents, and solving the WDP in our constraint programming framework comes down to solve the MaxLeximinCSP on a one-agent instance.

These results must however be considered with care, since they are subject to our implementation of the algorithms. For example, not every optimizations given in [13] for the constraint **Sort** have been implemented yet. The also depend on our modeling of the combinatorial auctions problem: we used a bid-centered modeling (that is, the decision variables are the bid allocations), with binary exclusion constraint to model the incompatibilities between the bids.

Anyway, it is interesting to notice that the performances of the algorithms have been dramatically increased by using the following variable choice heuristics. Choose as the next bid to allocate the first among the non-instantiated ones, according to the lexicographic increasing order on the two following criteria: 1) the current utility of the bid's owner, 2) the price of the bid. In other words, the next bid that the algorithm will try to select is the one with the highest price among those of the currently unhappiest agent.

It is also of interest to compare the quality of the leximin-optimal solution and the sum-optimal solution in term of fairness. One visual indicator of the fairness level of a solution is its Lorenz curve [15]. Formally, given a vector $\langle u_1, \ldots, u_n \rangle$, its Lorenz curve is the following vector: $\langle u_1^{\uparrow}, u_1^{\uparrow} + u_2^{\uparrow}, \cdots \sum_{i=1}^{n} u_i^{\uparrow} \rangle$. For a perfectly equitable utility vector, the Lorenz curve is a regular staircase line from the origin $(0, 0)$ to the point $(n, \sum_i u_i)$. On the opposite, a perfectly unfair utility vector (all agents having $u_i = 0$ except one) is very far from the regular staircase line. So the unfairness of a utility vector can be appreciated by the "distance" of the Lorenz curve to the regular staircase[8]. The Lorenz curve of a vector is always convex[9], and the less convex a Lorenz curve is, the fairer the vector is. Figure 1 shows the Lorenz curves of the utility vectors of the sum- and leximin-optimal solutions in a CA instance with 20 agents.

6 Conclusion

The leximin preorder cannot be ignored when dealing with optimization problems in which some kind of fairness must be enforced between utilities of agents or equally important criteria. This paper brings a contribution to the computation of leximin-optimal solutions of combinatorial problems. It describes, within a constraint programming framework, three generic algorithms solving this problem, the second one being entirely new. These algorithms have been tested on combinatorial auctions instances. The experimental results show that our algorithm is better than the others in all of the tested cases.

References

[1] N. Bleuzen-Guernalec and A. Colmerauer. Narrowing a block of sortings in quadratic time. In *Proc. of CP'97*, pages 2–16, Linz, Austria, 1997.

[8] In other words, its relative "convexity".

[9] That is: $u_{i+1}^{\uparrow} - u_i^{\uparrow} \geq u_{j+1}^{\uparrow} - u_j^{\uparrow} \Leftrightarrow i \geq j$.

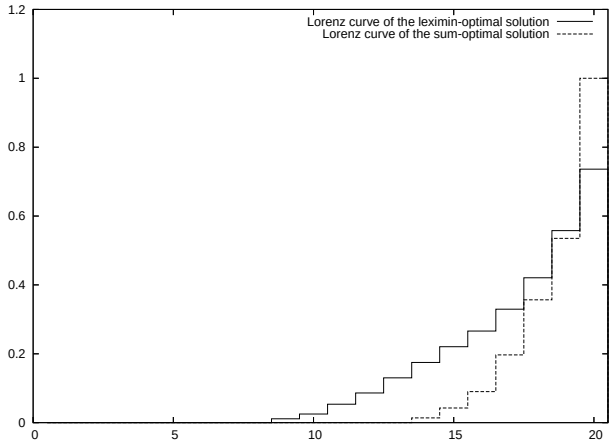


Figure 1: Lorenz curves of the utility vectors of the sum-optimal and leximin-optimal solutions in an combinatorial instance with 20 agents.

[2] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms, second edition*. MIT Press, 2001.

[3] Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[4] D. Dubois and P. Fortemps. Computing improved optimal solutions to max-min flexible constraint satisfaction problems. *European Journal of Operational Research*, 1999.

[5] M. Ehrgott. *Multicriteria Optimization*. Number 491 in Lecture Notes in Economics and Mathematical Systems. Springer, 2000.

[6] H. Fargier, J. Lang, and T. Schiex. Selecting preferred solutions in fuzzy constraint satisfaction problems. In *Proc. of EUFIT'93*, Aachen, 1993.

[7] A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, and T. Walsh. Multiset ordering constraints. In *Proc. of IJCAI'03*, February 2003.

[8] R. S. Garfinkel and G. L. Nemhauser. *Integer Programming*. Wiley, 1972.

[9] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons, 1976.

[10] F. Laburthe. CHOCO: implementing a CP kernel. In *Proc. of TRICS'2000, Workshop on techniques for implementing CP systems*, Singapore, 2000. `http://sourceforge.net/projects/choco`.

[11] M. Lemaître, G. Verfaillie, and N. Bataille. Exploiting a Common Property Resource under a Fairness Constraint: a Case Study. In *Proc. of IJCAI-99*, Stockholm, 1999.

[12] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of ACM Conference on Electronic Commerce (EC-00)*, 2000.

[13] K. Mehlhorn and S. Thiel. Faster algorithms for bound-consistency of the sortedness and the alldifferent constraint. In *Proc. of CP'00*, 2000.

[14] U. Montanari. Network of constraints: Fundamental properties and applications to picture processing. *Inf. Sci.*, 7:95–132, 1974.

[15] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[16] H. Moulin. *Fair division and collective welfare*. MIT Press, 2003.

[17] W. Ogryczak and T. Śliwiński. On solving linear programs with the ordered weighted averaging objective. *Europen Journal of O.R.*, 148:80–91, 2003.

[18] G. Pesant and J-C. Régin. SPREAD: A balancing constraint based on statistics. In *Proc. of CP'05*, Sitges, Spain, 2005.

[19] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 134:1–54, 2002.

[20] P. Van Hentenryck. *The OPL Optimization Programming Language*. The MIT Press, 1999.

[21] P. Van Hentenryck, H. Simonis, and M. Dincbas. Constraint satisfaction using constraint logic programming. *A.I.*, 58(1-3):113–159, 1992.

[22] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. on Systems, Man, and Cybernetics*, 18:183–190, 1988.

Sylvain Bouveret
ONERA-DCSD. 2, avenue Édouard Belin. BP4025. 31055 Toulouse cedex 4.
IRIT, Université Paul Sabatier. 31062 Toulouse cedex.
Email: `sylvain.bouveret@cert.fr`

Michel Lemaître
ONERA-DCSD. 2, avenue Édouard Belin. BP4025. 31055 Toulouse cedex 4.
Email: `michel.lemaitre@cert.fr`

The Computational Complexity of Choice Sets*

Felix Brandt Felix Fischer Paul Harrenstein

Abstract

Social choice rules are often evaluated and compared by inquiring whether they fulfill certain desirable criteria such as the *Condorcet criterion*, which states that an alternative should always be chosen when more than half of the voters prefer it over any other alternative. Many of these criteria can be formulated in terms of choice sets that single out reasonable alternatives based on the preferences of the voters. In this paper, we consider choice sets whose definition merely relies on the pairwise majority relation. These sets include the *Copeland set*, the *Smith set*, the *Schwartz set*, and *von Neumann-Morgenstern stable sets* (a concept originally introduced in the context of cooperative game theory). We investigate the relationships between these sets and completely characterize their computational complexity. This allows us to obtain hardness results for entire classes of social choice functions.

1 Introduction

Given a profile of individual preferences over a number of alternatives, the simple majority rule—choosing the alternative which the majority of agents prefer over the other alternative—is an attractive way of aggregating social preferences over any pair of alternatives. It has an intuitive appeal to democratic principles, is simple to understand and, most importantly, has some formally attractive properties. May's theorem shows that a number of rather weak and intuitively acceptable principles completely characterize the majority rule in settings with two alternatives (see May, 1952). Moreover, almost all common social choice rules satisfy May's axioms and thus coincide with the majority rule in the two alternative case. Thus it would seem that the existence of a majority of individuals preferring alternative a to alternative b signifies something fundamental and generic about the group's preferences over a and b. We will say that in any such case alternative a *dominates* alternative b.

Based on the simple majority rule, this dominance relation is obviously *asymmetric* in the strong sense that a dominating b implies that b does not dominate a. *A fortiori* the dominance relation is also irreflexive, *i.e.*, no alternative dominates itself. Conversely, any asymmetric binary relation on the set of alternatives, is induced as the dominance relation of some preference profile, provided that the number of voters is large enough compared to the number of alternatives (McGarvey, 1953). As is well known from Condorcet's paradox (de Condorcet, 1785), however, the dominance relation may very well contain cycles. This implies that the dominance relation need not

*This material is based upon work supported by the Deutsche Forschungsgemeinschaft under grant BR 2312/3-1.

have a maximum, or even a maximal, element, even if the underlying individual preferences do all have a maximum or maximal element. Thus, the concept of maximality has been rendered untenable in most cases.

There are several ways to get around this problem. One of which is, of course, to abandon the simple majority rule altogether. We will not consider such attempts here. Another would be to take more structure of the underlying individual preference profiles into account. We will not consider these here either. A third way would be to take the dominance relation for granted and define alternative concepts to take over the role of the maximality. As such we will be concerned with criteria for social choice correspondences that are based on the dominance relation only, *i.e.*, those that Fishburn (1977) called $C1$ functions. Formally, by a $C1$ social choice concept we will understand a concept that is invariant for all preference profiles that give rise to the same dominance relation. Examples of such concepts are *the Condorcet winner*, defined as the alternative, if any, that dominates all other alternatives. Other examples are:

- the *Copeland set, i.e.*, the set of all alternatives for which the difference between the number of alternatives it dominates and the number of alternatives that it is dominated by is maximal,
- the *Smith set, i.e.*, the smallest set of alternatives that dominate all alternatives that are not in the set,
- the *Schwartz set, i.e.*, the union of all minimal sets of alternatives that are not dominated by any alternative outside that set, and
- *von Neumann-Morgenstern stable sets, i.e.*, any set U consisting precisely of those alternatives that are not dominated by any alternative in U.

Social choice literature often mentions that one choice rule "is more difficult to compute" than another. The main goal of this paper is to provide formal grounds for such statements and, in particular, to obtain lower bounds for the computational complexity of entire classes of choice functions. This approach is inspired by Bartholdi, III et al. (1989) who proved the NP-hardness of any social *welfare* functional that is neutral, consistent, and Condorcet. They admit that "since only the Kemeny rule satisfies the hypotheses, this corollary is not entirely satisfying" (Bartholdi, III et al., 1989). During the last years, the computational complexity of various existing voting rules (such as the Dodgson, Kemeny, or Young rule) has been completely characterized (see Faliszewski et al., 2006, for a recent survey). However, we are not aware of any hardness results regarding broader classes of rules.

It is interesting to note that social choice theory literature almost exclusively deals with *tournaments*, *i.e.*, asymmetric and complete relations on a set of alternatives. For any odd number of *linear* individual preferences, the simple majority dominance relation is indeed a tournament. From a social choice perspective these could be taken as relatively mild and technically convenient restrictions. For one, the transitivity of a tournament implies its acyclicity and *vice versa*. Moreover, there can be at most

one maximal element in a tournament, and if there is one it is the *Condorcet winner*, the alternative that has a simple majority against any other alternative. Without these restrictions, the simple majority rule allows for ties and the dominance relation need not be complete. From the perspective of computational complexity, however, the restriction to tournaments is not as harmless as it might seem from a social choice point of view. We will find that some problems we consider are computationally significantly easier for tournaments than for the general case. Furthermore, in settings of computational interest such as webpage ranking there is usually a large number of alternatives over which the voters only have partial preferences with possibly many indifferences (see *e.g.*, Altman and Tennenholtz, 2005).

The remainder of this paper is structured as follows. The social choice setting we consider is introduced in Section 2. Section 3 motivates, introduces, and analyzes four choice sets whose computational complexity is investigated in Section 4. Section 5 concludes the paper with an overview and interpretation of the results.

2 Preliminaries

In a social choice setting, agents from a finite set N choose among a finite set A of alternatives. The cardinalities of these sets will be denoted n and m, respectively. For each agent $i \in N$ there is a binary preference relation $\succsim_i$ over the alternatives in A. We have $a \succsim_i b$ denote that player i values alternative a at least as much as alternative b. As usual, we write $\succ_i$ for the strict part of $\succsim_i$, *i.e.*, $a \succ_i b$ if $a \succsim_i b$ but not $b \succsim_i a$. Similarly, $\sim_i$ denotes i's indifference relation, *i.e.*, $a \sim_i b$ if both $a \succsim_i b$ and $b \succsim_i a$. We make no specific structural assumptions individual preferences should fulfill, apart from the indifference relation being reflexive and symmetric. Obviously, this includes all *linear orders—i.e.*, reflexive, transitive, complete and anti-symmetric relations—over the alternatives. On the other end of the spectrum, the definition also allows for *incomplete* or *quasi-transitive* preferences.[1]

Given a *preference profile* $(\succsim_i)_{i \in N}$, we say that alternative a *dominates* alternative b, in symbols $a \succ b$, whenever the number of voters for which $a \succsim_i b$ exceeds the number of voters for which $b \succsim_i a$. Obviously, the dominance relation is *asymmetric*. Despite the fact that most of the social choice literature has focused on *tournaments* (see *e.g.*, Laslier, 1997; Laffond et al., 1995), *i.e.*, complete dominance relations, the dominance relation need not in general be *complete*.[2] In fact, McGarvey (1953) shows that *any* dominance relation can be realized by a particular preference profile for a number of voters polynomial in m, even if individual preferences are transitive, complete and anti-symmetric. In the presence of *incomplete* or *quasi-transitive* preferences, incomplete dominance relations are more than just a theoretical possibility. In the remainder of this paper, we will be mainly concerned with dominance relations and tacitly assume appropriate underlying individual preferences.

[1] We say a relation $\geq$ is *asymmetric* whenever $x \geq y$ implies $y \not\geq x$. We say $\geq$ is *anti-symmetric* whenever $x \geq y$ and $y \geq x$ imply $x = y$. The relation $\geq$ is quasi-transitive, if $>$ (the strict part of $\geq$) is transitive.

[2] Obviously, one is guaranteed to obtain a complete dominance relation if the number of voters is odd and individual preferences are linear.

3 Choice sets

In this section, we motivate and introduce four choice sets based on the pairwise majority dominance relation and analyze the relationships between these sets.

We say that an alternative $a \in A$ is *undominated* in $X \subseteq A$ relative to $\succ$, whenever there are no alternatives $b \in X$ with $b \succ a$. We say that an element is *undominated* if it is undominated in A. A special type of undominated alternative is the *Condorcet winner*, which is an alternative that dominates every other alternative and is dominated by none. The concept of a *maximal element* we reserve in this paper for transitive (and possibly reflexive) relations $\geq$. An alternative $a \in A$ is said to be *maximal* in such a transitive relation, if there is no $b \in A$ such that $b \geq a$ but not $a \geq b$. Equivalently, the maximal elements of $\geq$ can be defined as the undominated elements in the strict (*i.e.*, asymmetric) part of $\geq$.

Given its asymmetry, transitivity of the dominance relation implies its acyclicity. The implication in the other direction holds for tournaments but not for the more general case. Failure of transitivity or completeness makes that a Condorcet winner need not exist; failure of acyclicity, moreover, that the dominance relation need not even contain maximal elements. As such, the obvious notion of maximality is no longer available to single out the "best" alternatives among which the social choice should be selected. Other concepts had to be devised to take over its role. In this paper, we will be concerned with four of these concepts: the Copeland set, the Smith set, the Schwartz set and von Neumann-Morgenstern stable sets.

3.1 Definitions

If a Condorcet winner exists, it is obviously the alternative that dominates the greatest number of alternatives, *viz.* all but itself, and is dominated by the smallest number, *viz.* by none. The *Copeland set* varies on this theme, by singling out those alternatives that maximize the difference between the number of alternatives they dominate and the number of alternatives they are dominated by (Copeland, 1951).

Definition 1 (Copeland score and Copeland set) *The* Copeland score $c(a)$ *of an alternative a given a dominance relation $\succ$ on a set of alternatives A equals* $|\{x \in A \mid a \succ x\}| - |\{x \in A \mid x \succ a\}|$. *The* Copeland set *$C$ is given by* $\{x \in A \mid c(a) \geq c(b)$, *for all* $b \in A\}$, i.e., *the set of alternatives with maximum Copeland score.*

Obviously, the Copeland set never fails to be non-empty and contains the Condorcet winner as its only element if there is one.

A set of alternatives X has the *Smith property* if any alternative in X dominates any alternative not in X, *i.e.*, if $x \succ y$ holds for all $x \in X$ and all $y \notin X$. Note that the set of all alternatives satisfies this property, and hence the existence of at least one subset of alternatives with the Smith property is trivially guaranteed. As is not hard to prove, the sets with the Smith property are, moreover, totally ordered by set inclusion. Hence, having assumed the set of alternatives to be finite, a unique *smallest* non-empty subset

of alternatives with the Smith property cannot fail to exist. This set, as it was originally proposed by Smith (1973), we refer to as the *Smith set*.[3]

Definition 2 (Smith set) *The* Smith set S *is the smallest non-empty set of alternatives with the Smith property,* i.e., *such that* $a \succ b$*, for all* $a \in S$ *and all* $b \notin S$.

If the Smith set contains only one element, this alternative is the Condorcet winner. Numerous choice rules always pick alternatives from the Smith set, *e.g.*, Nanson, Kemeny, or Fishburn (see, *e.g.*, Fishburn, 1977).

We say that a subset X of alternatives has the *Schwartz property* whenever no alternative in X is dominated by some alternative not in X, *i.e.*, for no $x \in X$ there is a $y \notin X$ with $y \succ x$. Vacuously the set of all alternatives satisfies the Schwartz property and so the existence of a non-empty subset with the Schwartz property is guaranteed. In contradistinction to the subsets with the Smith property, however, there need not be in general a *unique* minimal non-empty subset with the Schwartz property. With the set of alternatives having been assumed to be finite, we can single out those subsets with the Schwartz property that are both non-empty and are minimal ('smallest') with respect to set inclusion. We say that an alternative is in the *Schwartz set*, whenever it is an alternative of some such minimal subset with the Schwartz property (Schwartz, 1972).

Definition 3 (Schwartz set) *The* Schwartz set $T \subseteq A$ *is the union of all sets* $T' \subseteq A$ *such that:*

(i) there is no $b \notin T'$ *and no* $a \in T'$ *with* $b \succ a$*, and*

(ii) there is no non-empty proper subset of T' *that fulfills property* (*i*).

Alternatively, the Schwartz set could be defined as the set of maximal elements of the transitive closure of the dominance relation (*cf.* Lemma 1). It is also worth observing that, if the dominance relation is acyclic, the Schwartz set consists precisely of all undominated alternatives. Moreover, unlike the Smith set (and stable sets below), the Schwartz set can contain a single alternative without this alternative being the Condorcet winner. If there is a Condorcet winner, however, it will invariably be the only element of the Schwartz set. The Schwartz set coincides with the Smith set if the dominance relation is complete, *i.e.*, in the case of tournaments. Well-known choice rules that always pick alternatives from the Schwartz set are Schulze and ranked pairs (see, *e.g.*, Schulze, 2003).

The intuition behind *stable sets* can perhaps best be understood by thinking of the social choice situation as one in which the voters have to settle upon a selection of alternatives from which the eventual social choice is to be selected by lot or some other mechanism beyond their control. One could argue that any such selection should at least satisfy two properties. No majority can be found in favor of restricting the

[3]The Smith set appears in the literature under various names such as *top cycle*, *minimal undominated set*, or *Condorcet set*. It is also sometimes confused with the Schwartz set because in *tournaments* both sets coincide.

selection by excluding some alternative from it. In a similar vein, it must be possible to find a majority against each proposal to include an outside alternative in the selection. Formally, stable sets are defined as follows.

Definition 4 (Stable set) *A set of alternatives $U \subseteq A$ is* stable *if it satisfies the following two properties, also known as* internal *and* external *stability, respectively:*

(i) $a \succ b$*, for no* $a, b \in U$*, and*

(ii) for all $a \notin U$ *there is some* $b \in U$ *with* $b \succ a$*.*

Equivalently, stable sets can be given a single fixed point characterization:

> The alternatives in a *stable* set U are precisely those that are undominated by any alternative in U.

Observe that this definition does not exclude the possibility that an alternative outside a stable set dominates an alternative inside it.

Stable sets were proposed by von Neumann and Morgenstern (1944) to deal with intransitive dominance relations on imputations in the absence of a sensible concept of maximality. Originally, they were introduced as a solution concept for cooperative games and as such they have been studied extensively, especially in the 1950s. Richardson (1953), although also driven by game-theoretic motives, researched their formal properties in a more abstract setting. Within the context of social choice, stable sets have been paid considerably less attention to. If considered at all, it is only for a restricted class of situations (see, *e.g.*, Lahiri, 2004) or the concept is modified to some extent (see, *e.g.*, Dutta, 1988; van Deemen, 1991). One reason might be that in tournaments, a stable set exists if and only if there is a Condorcet winner, which it then contains as its only element. In the general case, however, neither uniqueness nor existence of stable sets is guaranteed. If the dominance relation is transitive, there is a unique stable set, which consists precisely of its maximal elements (and thus equals the Schwartz set). Moreover, a stable set is unique and a singleton if and only if there is Condorcet winner.

We conclude this section by stating without proof that none of the proposed sets may contain the Condorcet loser, *i.e.*, an alternative that is dominated by all other alternatives.

3.2 Dominance and Digraphs

It is very convenient to view the dominance relation derived from the voters' preferences as a directed graph $G = (V, E)$ where the set V of vertices equals the set A of alternatives and there is a directed edge $(a, b) \in E$ for $a, b \in V$ if and only if $a \succ b$ (see, *e.g.*, Miller, 1977). Figure 1 shows the digraph obtained for a set of six alternatives and the following profile of partial preferences for five voters (to improve readability, we only give the strict part of the preference ordering $\succsim_i$ for each voter $i \in N$): $e \succ_1 d \succ_1 c \succ_1 b \succ_1 a$, $b \succ_2 a \succ_2 e$, $d \succ_2 c \succ_2 f$, $a \succ_3 c$, $f \succ_3 e \succ_3 d$, $a \succ_4 c \succ_4 e$, $a \succ_4 b \succ_4 d$, and $e \succ_5 c \succ_5 a$. Since all choice sets considered in this paper are defined

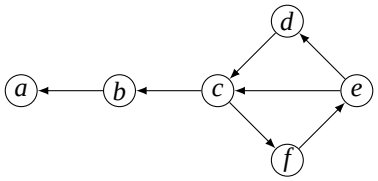


Figure 1: Dominance graph over a set of six alternatives and with Copeland set $C = \{e\}$, Smith set $S = \{a, b, c, d, e, f\}$, Schwartz set $T = \{c, d, e, f\}$, and the unique stable set $U = \{b, d, f\}$

in terms of the dominance relation only, we will henceforth restrict our attention to dominance graphs. From a computational perspective, we merely make the assumption that determining the dominance relation from a preference profile is easy, *i.e.*, no harder than computing the majority function on a string of bits. This is a reasonable assumption, since hardness of this operation obviously would mean hardness of any choice rule that takes individual preferences into account.

3.3 Relationships Between Choice Sets

Laffond et al. (1995) have conducted a thorough comparison of choice sets and derived various inclusions. However, their study is restricted to tournaments and does not cover stable sets. For this reason, this section provides an exhaustive set-theoretic analysis of the concepts defined in Section 3.1. We start by observing that all sets we consider are contained in the Smith set. Due to space restrictions, the proof of the following theorem is omitted.

Theorem 1 *The Copeland set, the Schwartz set, and every stable set are contained in the Smith set.* □

We leave it to the reader to verify that no other inclusion relationships between the discussed sets hold. In order to further investigate the significance of stable sets in the context of social choice, we now consider the relationship between the Schwartz set and stable sets. We start by providing a useful alternative characterization of the Schwartz set.

Lemma 1 *An alternative $a \in A$ is in the Schwartz set if and only if for every $b \in A$ such that there is a path from b to a in the dominance graph, there also is a path from a to b.*

Proof: Consider the Schwartz set T for a set A of alternatives and an arbitrary preference profile over A. For an alternative $a \in A$, let $D^*(a)$ denote the set of alternatives $b \neq a$ reachable from a in the dominance graph, and $\overline{D}^*(a)$ the set of alternatives $b \neq a$ from which a can be reached. Since the statement is trivially satisfied for alternatives that are undominated (*i.e.*, vertices with indegree zero), we only need to consider alternatives for which $\overline{D}^*(a) \neq \emptyset$.

To see the implication from left to right, assume for contradiction that $a \in T$, and that some $b \in D^*(a)$ is not reachable from a, *i.e.*, $\bar{D}^*(a) \setminus D^*(a) \neq \emptyset$. Since $a \in T$, there must be a minimal set $T_a \subseteq T$ with the Schwartz property and $a \in T_a$. Furthermore, by induction on the length of a shortest path from any $c \in \bar{D}^*(a)$ to a, it is easily verified that $\bar{D}^*(a) \subsetneq T_a$. On the other hand, there can be no alternative $c \in A \setminus \bar{D}^*(a)$ that dominates any alternative of $\bar{D}^*(a)$, since then there would be a path from c to a and thus $c \in \bar{D}^*(a)$. This contradicts the assumption that T_a is a minimal set with the Schwartz property.

Conversely assume that $a \notin T$ and that $\bar{D}^*(a) \subseteq D^*(a)$. Again, we only consider the case where a is dominated by at least one other alternative, hence $D^*(a) \neq \emptyset$. Then, however, $\bar{D}^*(a) \cup \{a\}$ satisfies the Schwartz property, and this does not hold for any proper nonempty subset, contradicting the assumption that a is not in the Schwartz set. □

Building on the previous lemma, it can be shown that the intersection of any stable set and the Schwartz set is always non-empty. We omit the proof to meet space restrictions.

Theorem 2 *Every stable set intersects with the Schwartz set.* □

4 Complexity Results

In the remainder of the paper, we investigate the computational complexity of the considered choice sets. We start by defining decision problems for the Condorcet winner and each of the four choice sets defined in Section 3.1 as follows: given a set A of alternatives, a particular alternative $a \in A$, and a preference profile $\{\succsim_i\}_{i \in N}$, IS-CONDORCET asks whether alternative a is the Condorcet winner for preference profile $\{\succsim_i\}_{i \in N}$, and IN-COPELAND, IN-SMITH, IN-SCHWARTZ, and IN-STABLE ask whether a is contained in the Copeland set, the Smith set, the Schwartz set, and a stable set for $\{\succsim_i\}_{i \in N}$, respectively. We further assume the reader to be familiar with the well-known chain of complexity classes $\mathrm{TC}^0 \subseteq \mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{NC} \subseteq \mathrm{P} \subseteq \mathrm{NP}$, and the notions of constant-depth and polynomial-time reducibility (see, *e.g.*, Johnson, 1990). TC^0 is the class of problems solvable by uniform constant-depth Boolean circuits with unbounded fan-in, a polynomial number of gates, and allowing so-called threshold gates which yield *true* if and only if the number of *true* inputs exceeds a certain threshold. Basic functions computable in this class have been investigated by Chandra et al. (1984). NC is the class of problems solvable by Boolean circuits with bounded fan-in and a polynomial number of gates. L and NL are the classes of problems solvable by deterministic and nondeterministic Turing machines using only logarithmic space, respectively. P and NP are the classes of problems that can be solved in polynomial time by deterministic and nondeterministic Turing machines, respectively.

First of all, we observe that a particular entry in the adjacency matrix of the dominance graph for a preference profile $(\succsim_i)_{i \in N}$ is given by the majority function for a particular pair of alternatives, and that the complete adjacency matrix can be computed in TC^0. Showing that IS-CONDORCET is in TC^0 is also straightforward. We

just have to check whether all entries in the row of the adjacency matrix corresponding to a are 1. Hardness, on the other hand, follows from the fact that the case $m = 2$ is equivalent to computing the majority function on a string of bits, which in turn is hard for TC^0. For IN-COPELAND, we have to check whether the difference between outdegree and indegree of the vertex corresponding to a is maximal over all vertices in the dominance graph. We can do this by computing, for each row of the adjacency matrix in parallel, the sum of all entries in this row and subtract the sum of all entries in the corresponding column. Finally, we check whether the result for the row (and column) corresponding to a attains the maximum over all pairs of rows (and corresponding columns). Hardness follows from the fact that IN-COPELAND and IS-CONDORCET are equivalent for the case of two alternatives and an odd number of voters with linear preferences.

It is well-known that both the Smith set and the Schwartz set can be computed in polynomial time by applying the algorithm of Kosaraju for finding strongly connected components in the dominance graph. In graph-theoretic terms, the Smith set is the maximal strongly connected component in the digraph for the *majority-or-tie* dominance relation, while the Schwartz set is the maximal strongly connected component for the *majority* dominance relation. Our approach for computing the Smith set is quite different and based on the in- and outdegree of vertices inside and outside that set. Assume there exists a Smith set $S \subseteq A$ of size k. Since by definition every member of S must dominate every non-member, the outdegree of every element of S in the dominance graph for A must be at least $n - k$, while every alternative not in S must have indegree at least k. Furthermore, no alternative can satisfy both properties because the sum of in- and outdegree for each vertex in an asymmetric digraph is bounded by $n-1$. Given a particular k, we can thus try to partition A into two sets S' and $\overline{S'} = A \setminus S'$ by the above criterion, such that S' is the unique candidate for a set of size k that satisfies the Smith property. We can then easily check whether S' actually satisfies the Smith property, and find the Smith set by repeating this process for $1 \leq k \leq n$. We proceed to show that this algorithm can be implemented using a constant depth threshold circuit, and that checking membership in the Smith set is actually complete for the class TC^0.

Theorem 3 *IN-SMITH is TC^0-complete.*

Proof: Hardness is immediate from the equivalence of IN-SMITH and IS-CONDORCET for the case of two alternatives and an odd number of voters with linear preferences.

For *membership*, we construct a constant depth threshold circuit that decides whether there exists a set of size k with the Smith property. We can then perform the checks for all possible values of k in parallel, and decide whether a particular alternative is in the smallest such set. We start by computing the adjacency matrix $M = (m_{ij})$ of the dominance graph from the preference profile. This amounts to a polynomial number of majority votes over pairs of alternatives and can obviously be done in TC^0. We then apply a threshold of $n - k$ to each row of M to obtain a vector v such that v_i is *true* if and only if the ith alternative is in the potential Smith set S'. To decide whether S' actually satisfies the Smith property, we have to check whether the outdegree of vertices in S' is still high enough if we only consider edges to vertices in $\overline{S'}$,

i.e., whether the properties regarding in- and outdegree are satisfied for the *bipartite part* of A with respect to S' and $\overline{S'}$. We thus compute the adjacency matrix $M^b = (m^b_{ij})$ for the bipartite part of A as $m^b_{ij} = (m_{ij} \wedge \neg v_j)$ and again apply a threshold of $n - k$ to each row to yield a vector v^b. S' satisfies the Smith property if and only if a threshold of k applied to v^b yields *true*. In this case, the ith alternative is contained in this set if $v^b_i = \mathit{true}$. □

The previous theorem implies that any choice rule that picks its winner from the Smith set is TC^0-hard, and thus in principle not harder than any Condorcet choice rule. As noted above, the Smith set and the Schwartz set differ only by their treatment of ties in the pairwise comparison. Nevertheless, and quite surprisingly, deciding membership in the Schwartz set is computationally harder unless TC^0=NL.

Theorem 4 *IN-SCHWARTZ is NL-complete.*

Proof: Given a dominance graph and using Lemma 1, membership of an alternative $a \in A$ in the Schwartz set can be shown by checking for every other alternative $b \in A$ that either b is reachable from a or a is not reachable from b. Clearly, the existence of a particular edge in the dominance graph and hence the existence of a path between a pair of vertices can be decided by a nondeterministic Turing machine using only logarithmic space. Membership in the Schwartz set can then be decided using an additional pointer into the input to store alternative b.

For *hardness*, we provide a reduction from the NL-complete problem of digraph reachability (see, *e.g.*, Johnson, 1990). Given a particular digraph $G = (V, E)$ and two designated vertices $s, t \in V$, we construct a dominance graph $G' = (V', E')$ by adding an additional vertex u, an edge from t to u, and edges from u to any vertex but t, *i.e.*, $V' = V \cup \{u\}$ and $E' = E \cup \{(t, u)\} \cup \{\, (u, v) \mid v \in V, v \neq t \,\}$. It is easily verified that G' can be computed from G by a Boolean circuit of constant depth. We claim that s is contained in the Schwartz set for G' if and only if there exists a path from s to t in G. First of all, we observe that a path from s to t in G' exists if and only if such a path already existed in G, since we have not added any outgoing edges to s or any incoming edges to t. By construction, every vertex of G', including s, can be reached from t. Hence, by Lemma 1, s cannot be contained in the Schwartz set if t cannot be reached from s. Conversely assume that t is reachable from s. Then this property holds as well for every vertex of G', particularly those from which s can be reached. In virtue of Lemma 1, we may conclude that s is in the Schwartz set. □

For all choice sets considered so far, we can check efficiently whether they contain a particular alternative or not. Unfortunately, this is not case for stable sets (unless P=NP).

Theorem 5 *IN-STABLE is NP-complete, even if a non-empty stable set is guaranteed to exist.*

Proof: Membership in NP is obvious. Given a dominance graph over a set A of alternatives and a particular alternative $a \in A$, we can simply guess a subset $U \subseteq A$ such

that $a \in U$, and verify that for every $b \notin U$ there is an edge from some element of U to b and that there are no edges between vertices of U.

For *hardness*, we provide a reduction from satisfiability of a Boolean formula B (SAT) to the problem of deciding whether a designated alternative $a \in A$ is contained in a stable set (or the union of all stable sets). The reduction is based on the reduction by Chvátal (1973) to show NP-hardness of the problem of deciding whether a digraph has a kernel. Let $B = \bigwedge_{1 \leq i \leq m} \bigvee_{1 \leq j \leq k_i} p_{ij}$ be a SAT instance over variables X. We construct an asymmetric dominance graph $G = (V, E)$ with three vertices c_{i1}, c_{i2}, and c_{i3} for each clause of B, four vertices x_i, $\overline{x_i}$, x'_i, and $\overline{x'_i}$ for each variable of B, and four additional vertices d_1, d_2, d_3, and d_4, such that d_1 is contained in a stable set if and only if B has a satisfying assignment. Vertices c_{ij} will henceforth called clause vertices, x_i and $\overline{x_i}$ will be referred to as positive and negative literal vertices, respectively. Edges are such that the vertices of each clause form a directed cycle of length three, and the vertices of each variable as well as the decision vertices form a cycle of length four according to the sequence given above. Furthermore, there is an edge from a positive or negative literal vertex to all clause vertices of a clause in which the respective literal appears. Finally, there is an edge from d_2 to every clause vertex. More formally, we have

$$
\begin{aligned}
E = &\{ (d_1, d_2), (d_2, d_3), (d_3, d_4), (d_4, d_1) \} \cup \\
&\{ (c_{i1}, c_{i2}), (c_{i2}, c_{i3}), (c_{i3}, c_{i1}) \mid 1 \leq i \leq m \} \cup \\
&\{ (x_i, \overline{x_i}), (\overline{x_i}, x'_i), (x'_i, \overline{x'_i}), (\overline{x'_i}, x_i) \mid 1 \leq i \leq |X| \} \cup \\
&\{ (x_i, c_{j1}), (x_i, c_{j2}), (x_i, c_{j3}) \mid p_{j\ell} = x_i \text{ for some } 1 \leq \ell \leq k_j \} \cup \\
&\{ (\overline{x_i}, c_{j1}), (\overline{x_i}, c_{j2}), (\overline{x_i}, c_{j3}) \mid p_{j\ell} = \overline{x_i} \text{ for some } 1 \leq \ell \leq k_j \} \cup \\
&\{ (d_2, c_{i1}), (d_2, c_{i2}), (d_2, c_{i3}) \mid 1 \leq i \leq m \}.
\end{aligned}
$$

Figure 2 illustrates this construction for a particular Boolean formula. We observe the following facts: G can be constructed from B in polynomial time. $\{ x_i, x'_i \mid 1 \leq i \leq m \} \cup \{d_2, d_4\}$ is a stable set of G irrespective of the structure of B. Every stable set of G must either contain d_1 and d_3 or d_2 and d_4, but not both. For each i, every stable set must either contain x_i and x'_i or $\overline{x_i}$ and $\overline{x'_i}$, but not both. A stable set of G cannot contain a pair of clause vertices for the same clause. In turn, a stable set must contain vertices with outgoing edges to at least two of the three vertices for every clause. However, every vertex that has an outgoing edge to any vertex for some clause has an outgoing vertex to all three vertices for that clause. Hence, a stable set cannot contain any clause vertices. A stable set must contain either d_2 or a subset of the literal vertices containing at least one vertex for a literal in every clause. Since a stable set cannot contain both x_i and $\overline{x_i}$, the latter corresponds to a satisfying assignment B. Hence, a stable set containing d_1 exists if and only if B is satisfiable. □

We can actually derive a stronger result, concerning the computational complexity of *any* choice rule that is guaranteed to select an alternative from a stable set, if such an alternative exists.

Theorem 6 *Consider a choice rule that selects an alternative from a stable set if one exists and an arbitrary alternative otherwise. This choice rule cannot be executed in worst-case polynomial time unless P=NP.*

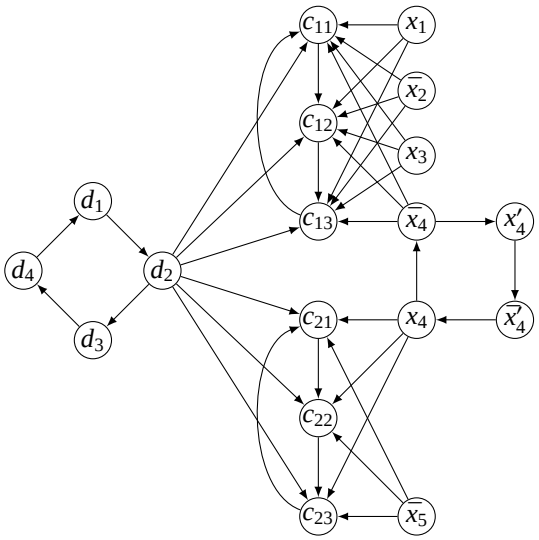


Figure 2: Dominance graph for the Boolean formula $(x_1 \vee \overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (x_4 \vee \overline{x_5})$ according to the construction used in the proof of Theorem 5. If a certain variable appears exclusively as either positive or negative literal, the other three vertices for the variable can be omitted.

Proof: Again consider the construction used in the proof of Theorem 5 and illustrated in Figure 2. In this construction, four designated vertices d_1 to d_4 have been used to guarantee the existence of a stable set, no matter whether the underlying Boolean formula B has a satisfying assignment or not. This guarantee also means that finding *some* alternative that belongs to a stable set is trivial. It is easily verified that if we remove vertices d_1 to d_4, a stable set in graph G exists if and only if B has a satisfying assignment, and the vertices in such a stable set are those corresponding to the literals set to true in a particular satisfying assignments.

Now consider a Turing machine with an oracle that computes a single alternative belonging to a stable set, if such a set exists, and an arbitrary alternative otherwise. Using this machine, the existence of a satisfying assignment for a particular Boolean formula B can be decided as follows. First, compute the dominance graph $G = (V, E)$ corresponding to B. Then, iteratively reduce the graph by requesting a vertex v from the oracle and removing vertices as follows: if $v = x_i$ or $v = x'_i$ for some $1 \leq i \leq |X|$, remove x_i, x'_i, $\overline{x_i}$, $\overline{x'_i}$ and all c_{ij} such that $(x_i, c_{ij}) \in E$; if $v = \overline{x_i}$ or $v = \overline{x'_i}$ for some $1 \leq i \leq |X|$, remove x_i, x'_i, $\overline{x_i}$, $\overline{x'_i}$ and all c_{ij} such that $(\overline{x_i}, c_{ij}) \in E$. If at some point there no longer exists any vertex c_{ij}, let the machine halt and accept. If at some point there no longer exists any x_i or $\overline{x_i}$ but there still is some c_{ij}, or if the oracle returns c_{ij} for some $1 \leq i \leq m$, $j \in \{1, 2, 3\}$, let the machine halt and reject.

As already pointed out in the proof of Theorem 5, the graph G can be computed from B in polynomial time. In every later step, the machine either halts or removes at least one vertex, of which there are only polynomially many. Hence, the machine

	tournaments	general dominance graphs
IS-CONDORCET	TC^0-complete	TC^0-complete
IN-COPELAND		
IN-SMITH		
IN-SCHWARTZ		NL-complete
IN-STABLE		NP-complete

Table 1: Complexity of choice sets

is guaranteed to halt after a polynomial number of steps. Furthermore, if the machine accepts, the set of all vertices returned by the oracle form a stable set of G, which can only exist if B has a satisfying assignment. We have thus provided a Cook reduction from SAT to the problem of selecting an arbitrary element of a stable set, showing that a polynomial-time algorithm for the latter would imply P=NP. □

While the union of all stable sets need not in general be contained in the Schwartz set (see *e.g.*, Figure 1), this is the case for the dominance graphs used in the proofs of the previous two theorems. Hence, hardness holds as well for deciding whether an alternative lies in the intersection of a stable set and the Schwartz set, and for any choice rule that selects an alternative that is both in a stable set and in the Schwartz set.

5 Conclusion

We have investigated the relationships and computational complexity of various choice sets based on the pairwise majority relation. Table 1 summarizes our complexity-theoretic results, which can be interpreted as follows. All considered problems except IN-STABLE are computationally tractable. Moreover, these problems are contained in the complexity class NC of problems amenable to parallel computation. All problems except IN-SCHWARTZ and IN-STABLE can be solved on a deterministic Turing machine using only logarithmic space. These results can be used to make statements regarding the complexity of entire classes of choice rules, *e.g.*, the hardness of every choice rule that picks an alternative from a stable set.

In addition, Table 1 underlines the significant difference between tournaments and general dominance graphs. Surprisingly, the Smith set turned out to be computationally easier than the Schwartz set in general dominance graphs (unless TC^0=NL), while both concepts coincide in tournaments. Deciding whether an alternative is included in a stable set is NP-complete in general dominance graphs, while in tournaments the same problem is equivalent to the TC^0-complete problem of deciding whether the alternative is the Condorcet winner.

Finally, it should be noted that our results are fairly general in the sense that they only rely on the *asymmetry* of the dominance relation. As a matter of fact, all considered sets are reasonable substitutes for maximality in the face of non-transitive relations, no matter whether these relations stem from aggregated preferences or not.

References

A. Altman and M. Tennenholtz. On the axiomatic foundations of ranking systems. In *Proc. of 19th IJCAI*, pages 917–922. Professional Book Center, 2005.

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(3):157–165, 1989.

A. K. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *SIAM Journal on Computing*, 13(2):423–439, 1984.

V. Chvátal. On the computational complexity of finding a kernel. Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal, 1973.

A. H. Copeland. A 'reasonable' social welfare function. mimeographed, University of Michigan Seminar on Applications of Mathematics to the social sciences, 1951.

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.

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

P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. Technical Report TR-903, Department of Computer Science, University of Rochester, 2006.

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

D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2, pages 67–161. Elsevier, 1990.

G. Laffond, J. F. Laslier, and M. Le Breton. Condorcet choice correspondences: A set-theoretical comparison. *Mathematical Social Sciences*, 30:23–35, 1995.

S. Lahiri. Stable sets of weak tournaments. *Yugoslav Journal of Operations Research*, 14:33–40, 2004.

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

K. May. A set of independent, necessary and sufficient conditions for simple majority decisions. *Econometrica*, 20:680–684, 1952.

D. C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

N. R. Miller. Graph-theoretic approaches to the theory of voting. *American Journal of Political Science*, 21 (4):769–803, 1977.

M. Richardson. Solutions of irreflexive relations. *The Annals of Mathematics*, 58(3):573–590, 1953.

M. Schulze. A new monotonic and clone-independent single-winner election method. *Voting Matters*, 17: 9–19, 2003.

T. Schwartz. Rationality and the myth of the maximum. *Noûs*, 6(2):97–117, 1972.

J. H. Smith. Aggregation of preferences with variable electorate. *Econometrica*, 41(6):1027–1041, 1973.

A. M. A. van Deemen. A note on generalized stable sets. *Social Choice and Welfare*, 8:255–260, 1991.

J. von Neumann and O. Morgenstern. *The Theory of Games and Economic Behavior*. Princeton University Press, 1944.

Felix Brandt, Felix Fischer, and Paul Harrenstein
University of Munich
80538 Munich, Germany
Email: {brandtf,fischerf,harrenst}@tcs.ifi.lmu.de

Natural Rules for Optimal Debates
(Preliminaries for a Combinatorial Exploration)

Yann Chevaleyre and Nicolas Maudet

Abstract

Two players hold contradicting positions regarding a given issue, which depends on a (fixed) number of aspects or criteria they both know. Suppose, as a third-party, that you want to make a decision based on what will report the players. Unfortunately, what the players can communicate is limited. How should you design the rules of your protocol so as to minimize the mistakes induced by these communication constraints? This paper discusses this model originally due to [2] in a specific case variant, and introduces preliminary results of a combinatorial exploration of this problem.

1 Introduction

The situation is the following: Two debaters have contradicting positions regarding a given issue, which depends on a (fixed) number of aspects, or criteria. The value of these aspects being given, there is common knowledge of the decision rule which will eventually selects the outcome (for instance, the majority). They both know what the "actual" state of the world is (so they both know who should be the actual winner). Unlike the players, a third-party agent is not aware of the real state. Now they exchange arguments (e.g. claiming that a given aspect of the state supports their opinion) during a debate, with the aim of convincing this external observer of their position. Of course, what makes the problem interesting is that there is a limitation on the number of communications they can make.

This problem introduced by Glazer and Rubinstein in [2] is a *mechanism design* problem: Designing the rules of the debates such that the probability for the observer to reach the "right" (the one that would be taken with full knowledge of the state) decision is actually maximal.

Basically, a debate consists of two elements:

- *procedural rule*– specifies the protocol constraining the arguments that the debater agent can raise (here some assumptions are made: an agent can just raise arguments supporting his favoured outcome, and nothing else);
- *persuasion rule*– specifies how the observer should make his decision based on the arguments advanced during the debate.

As far as the procedural rules are concerned, the authors discuss three canonical types of debates: (i) only one debater is allowed to speak (*single-speaker*

debate); (ii) two debaters argue simulatenously (*simultaneous debate*), and (iii) debaters raise sequentially arguments (*sequential debate*). In [2], the authors investigate the three types of debate in the restricted 5-aspects setting (where the numbers of arguments to be communicated is limited to 2), and show in particular that the optimal rule in this context is necessarily sequential. In this preliminary work, we want to initiate the investigation of the extremal behaviour of this problem (when n becomes very large), and we start with the simple case where only one player is allowed to raise arguments (*single-speaker debate*).

The rest of the paper is as follows. In the next section we introduce the basic definitions that will be used throughout this paper. Section 3 then presents the analysis of different sorts of "natural" persuasion rules that a designer may wish to use in order to make his decision . By natural we mean that they can be simply stated in natural language by the designer. We provide an analytical analysis of two very simple rules ("give me any set of size k", and "give me that set"), and offer some preliminary insights of the behaviour of the rules that fall within the vast region in between. These latest findings are mostly supported by experimentations. Section 4 concludes and draws some connections with related works.

2 Basic Definitions

In this section we introduce more formally the problem as stated by [2], sometimes slightly deviating from the original version to introduce are own notations.

A *state* is a binary vector $\{0,1\}^n$, and each player (0,1) "controls" the bits (arguments) of his colour (that is, he cannot lie and cannot play the bits of the other player). We say that a state is an *objectively winning state* for agent x if a fully-informed designer would declare x winner in that state. For instance, the state $\{0,1,1,1,1\}$ means that the first argument is in favour of agent 0, while all the others are supporting agent's 1 view. This is an objectively winning situation for agent 1 (we assume the majority rule).

Typically, only k bits of communication will be allowed in our debates (with $k < n/2$ for obvious reasons as we consider the majority rule). A *persuasion rule* is defined in extension as a set

$$E = \{S_1, S_2, \ldots, S_n\}$$

where each set S_i is a subset of $[n]$ of size k (k-subset). Such a rule must be interpreted as follows: "I would declare you winner if you can raise all the arguments contained in S_1, or all the arguments contained in S_2, etc.". For instance, the persuasion rule $E = \{\{1,2\},\{2,3\}\}$ means that the agent must either show arguments 1 and 2, or 2 and 3 (but 1 and 3 is not sufficient) to be declared winner. In this paper we will be interested in persuasion rules that can be simply stated in natural language (typically because they exploit some properties of the k-subsets composing the rules).

The *error ratio* (ϵ) induced by a rule is the number of states where you would take an erroneous decision when compared to what a fully-informed designer would do (n_{err}), normalised over all possible states. If you take a closer look at the notion of error, it actually occurs that two types of errors can be distinguished:

- *minority errors*, correponding to states where you would declare an agent winner, although this agent doesn't hold a winning position
- *majority errors*, corresponding to states where you would declare an agent loser, although the state is objectively winning for him.

Take the example given above, and assume a 5-bits debate. In states $\{1,1,0,0,0\}$ and $\{0,1,1,0,0\}$, agent 1 can convince the designer despite the state being objectively losing for him. On the other hand, in states $\{0,1,0,1,1\}$, $\{1,0,1,0,1\}$, $\{1,0,1,1,0\}$, and $\{1,0,1,1,1\}$ agent cannot convince the designer that its position is winning. This makes 6 errors overall (2 in favour of agent 1, 4 in favour of the other agent). Although, as correctly noticed by a reviewer, one type of error is the dual of the other (a minority error for one agent is a majority error for the other agent; or, to put it differently, any error is either a minority error for one agent or a minority error for the other agent), it is still useful to distinguish both types. The main reason is that it provides some information concerning which agent is favoured by a given rule.

In the following we will also make use of some additional notions. We say that a persuasion rule is *covered* by a state vector when at least one of its composing rule is covered by that state vector, that is when any argument required by that set is in that set. In these terms, the optimization problem we are faced with is to find the persuasion rule that will minimize the covering over the set of vectors containing $[k, \frac{n}{2}[$ bits (objectively losing situations), while maximizing the covering over the universe of vectors containing $n/2$ or more bits (objectively winning situations). We shall note these two measures respectively c_m and c_M from now on.

It is worth noting that in general (for $k \leq n/2$) the following holds:

$$n_{err} = c_m + (2^{n-1} - c_M)$$

The number of errors is simply the number of covering minority states, added up to the number of majority states (2^{n-1}) that are not covered by the rule.

3 Natural Rules

In this section we discuss the properties of some natural persuasion rules. Natural must be understood here as the fact that they can be simply stated in natural language by the designer (which does not necessarily imply that E will exhibit a simple structure in its extensive form). We refer the reader to [4] for an enlightning discussion on that topic. There are many "natural" rules you

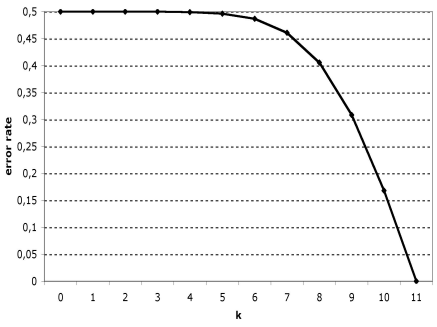


Figure 1: Error ratio induced by the "give me any set of size k" rule ($n = 20$)

can possibly come up with, and some examples are given in [2], like for instance "give me k adjacent bits". In what follows, we first discuss two very (arguably, the most) simple rules, before moving on to the general case lying between these two extreme rules.

3.1 "Give me *any* of size k"

We start with what may be the simplest rule, simply enonced as follows: "give me any subset of size k". Or maybe even more naturally as "give me k bits", without any further constraints. In other words, the set E would consist of the set exhausting any k-subsets of $\{0,1\}^n$. What would be the error induced by this rule? Note first that the majority error is bound to be 0 when $k \in [1, \frac{n}{2}]$. In general, the overall number of errors would then be equal to the number of losing situations covered by the rule (c_m). Take t as being the number of bits to still be placed to make a losing situation once you have covered the rule. There are

$$n_{err} = c_m = \sum_{t=k}^{\lfloor n/2 \rfloor} \binom{n}{t}$$

such situations, that is, the number of errors is given by the sum of binomial coefficients from k to $n/2$. This means that this rule is pretty ineffective: only when the number of bits allowed to be communicated becomes very close to $n/2$ does it give a good error ratio (see Fig. 1). And indeed, if you were allowed to ask the agent to communicate any number of bits, this is the perfect rule you would of course use: by requesting the agent to put forward $n/2$ aspects in favour of his view, you are sure that no agent can fool you in a losing state, while not missing any winning state at the same time.

3.2 "Give me *that* set"

In that case we assume that the designer can ask the agent to simply give just one set ($|E| = 1$), of arbitrary size k. (We assume n to be odd.) The minority and majority covering are as follows:

$$c_m = \sum_{i=0}^{\lfloor n/2-k \rfloor} \binom{n-k}{i}$$

$$c_M = \sum_{i=\lceil n/2-k \rceil}^{n-k} \binom{n-k}{i}$$

In that case, we have $c_M \geq c_m$.
Observe that $c_M + c_m = 2^{n-k}$, hence we have

$$\begin{aligned} n_{err} &= c_m + 2^{n-1} - (2^{n-k} - c_m) \\ &= 2c_m + (2^{n-1} - 2^{n-k}) \end{aligned}$$

The error ratio is then

$$\begin{aligned} \epsilon &= \frac{2c_m + (2^{n-1} - 2^{n-k})}{2^n} \\ &= \frac{c_m}{2^{n-1}} + \frac{1}{2} - 2^{-k} \end{aligned}$$

We will now show that this is an increasing monotonic function.

Lemma 1 *For odd values of n and for $k \geq 1$, the error rate of the "give me* that *set" rule increases as k grows.*

Proof. Let n be odd. We will show that $\frac{n_{err}-2^{n-1}}{2} = c_m - 2^{n-k-1}$ is a increasing function of k. More precisely, we will show that the value c_m decreases as k grows, but that 2^{n-k-1} decreases faster, thus ensuring that n_{err} increases as k grows. To achieve this, it suffices to show that $c_m^k - c_m^{k+1} \leq 2^{n-k-1} - 2^{n-k-2} \leq 2^{n-k-2}$. In the following, we make use of the binomial formula : $\binom{x}{y} = \binom{x-1}{y-1} + \binom{x-1}{y}$.

$$
\begin{aligned}
c_m^k - c_m^{k+1} &= \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor - k} \binom{n-k}{i} - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor - k - 1} \binom{n-k-1}{i} \\
&= \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor - k - 1} \left\{ \binom{n-k}{i} - \binom{n-k-1}{i} \right\} + \binom{n-k}{\lfloor \frac{n}{2} \rfloor - k} \\
&= \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor - k - 1} \binom{n-k-1}{i-1} + \binom{n-k}{\lfloor \frac{n}{2} \rfloor - k} \\
&= \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - k - 2} \binom{n-k-1}{i} + \binom{n-k}{\lfloor \frac{n}{2} \rfloor - k} \\
&= \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - k - 2} \binom{n-k-1}{i} + \binom{n-k-1}{\lfloor \frac{n}{2} \rfloor - k - 1} + \binom{n-k-1}{\lfloor \frac{n}{2} \rfloor - k} \\
&= \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - k} \binom{n-k-1}{i}
\end{aligned}
$$

First, it can be easily verifyied that $\lfloor \frac{n}{2} \rfloor - k \leq \left\lfloor \frac{n-k-1}{2} - 1 \right\rfloor$ for all $k \geq 1$ and $n \geq 1$. Exploiting the fact that $\sum_{i=0}^{\lfloor \frac{x-1}{2} \rfloor} \binom{x}{i} \leq 2^{x-1}$ for any $x \in \mathbb{N}$, and substituting x with $n-k-1$ we can now write the following, which completes the proof.

$$
c_m^k - c_m^{k+1} \leq \sum_{i=0}^{\lfloor \frac{n-k-1}{2} - 1 \rfloor} \binom{n-k-1}{i} \leq 2^{n-k-2}
$$

What does it tell us? Well, simply that if you have only one set to ask, then the smaller subset the better—in other words, just ask one bit. Of course you should not expect a very good error ratio (for instance, for $n = 20$ the error ratio starts at 40% for the singleton set and then tends towards 50% when k grows.)

3.3 The mostly unnatural region in between

So far we have studied two extreme natural cases: the case where only one set is asked, and the case when any k-subset is asked. It would be interesting to observe the behaviour of the persuasion when the number sets composing the persuasion lies in between (although it would be unlikely in general that the obtained rule would be natural). To do that, we first derived an analytical formula (shown in Appendix) representing the error rate in the general case. Unfortunately, deriving upper and lower bounds on such a formula proved to

be difficult, and we did not get any satisfying result yet. For this reason, we choose to set up an experimental study, whose most strinking result is reported below ($n = 21$, a number $|E|$ of k-subsets is randomly generated to create a rule). Note that the axis representing the cardinality of E is logarithmic

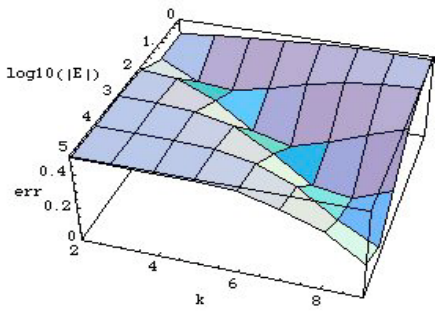


Figure 2: Error ratio of randomly generated rules of size $|E|$, depending on k

($log_{10}|E|$). This is so because we observed that the value of k for which the error is minimized depends logarithmically on the size of E. During all our experiments, we noticed that, while measuring the error rate as a function of k, having set the other parameters of the simulation, the error rate always decreases until k reaches a particular value which we will refer to as k_{opt} (this value depends on the other parameters), and then increases again. This can also apply to the extremal persuasion rules described above : consider the "give me that set" rule ($|E|$=1). Its error rate is best at $k = 1$, and then increases. Thus, k_{opt} for this rule equals to one. On the contrary, the error rate of the "give me any set" rule ($|E| = \binom{n}{n/2}$) always decreases as a function of k, until k reaches $\frac{n}{2}$. Thus, setting $k_{opt} = \frac{n}{2}$ also fits our framework. Thus, finding the value of k_{opt} is highly relevant to our problem. Further experimentations not reported here strongly suggest that this optimal value, for $n = 21$, is $2.log_{10}|E| + 2$. Fig. 2 shows the output of that particular experiment: for instance, when $log_{10}(|E|) = 1$ (10 k-subsets) we have $k_{opt} = 2$, when $log_{10}(|E|) = 2$, $k_{opt} = 4$, and when $log_{10}(|E|) = 5$, we finally have $k_{opt} = 9$ (the error ratio is then 8%).

3.4 Partitions of "Give me k bits within that set" sets

We now briefly discuss a case of historical interest, which represents a special (rather natural, see) family of rules for which the arguments can be, in some sense, clustered. Recall that in the case n=5, it has been proven by [2] that the optimal rule for this kind of debate is the rule consisting of asking the player to raise two bits either within the set $\{1, 2, 3\}$ or within $\{4, 5\}$. This rule could of course be represented in extension as $E = \{\{1, 2\}, \{2, 3\}, \{1, 3\}, \{4, 5\}\}$, but

its appealing naturalness lies in the use of the IN-like operator which allows its compact representation, together with the fact that the obtained rule is a partition. To give us some hints as to whether that OR of IN partitions exhibited a good behaviour when the value of n becomes larger, we have conducted limited experiments. As means of example, the preliminary experiments that we ran with such partitions (with $n = 21$ and $k = 3$) give an error rate of 47% with 3 IN-subsets (we report here the optimal error ratio found after the random generation of such rules), of 36% with 5 IN-subsets, and 26% for 6 subsets, to eventually reach 21% for the partition consisting of exactly 7 subsets of size 3.

What these experiments show is that the error ratio constantly decreases when the number of IN-subsets augments. Although this may seem somewhat surprising at first sight, you have to notice that when the number of IN-subsets augments, the cardinality of E (defined in extension as k-subsets) actually decreases. This confirms the observations made in the previous subsection. Overall, the best rule possible belonging to this class seems then to be the rule composed of $\lfloor n/2 \rfloor$ IN-subsets of size k (or $k + 1$ for some number of agents $< n$) each, although we need more evidence to be able to firmly conclude. Note, however, that this is indeed in line with the result reported in [2].

4 Related and Future Works

Our ambition with this preliminary work is to initiate the study of the extremal behaviour of a mechanism problem introduced in [2]. The first results that we obtain here mainly concern two very simple kind of rule: "give me any set of size k", and "give me that set". Although the "give me any set of size k" rule is the only perfect rule when the communication is unrestricted, we show here that it is pretty ineffective in general when we put some limit on the number of bits to be transmitted. As for the "give me that set" rule, our result remarkably shows that the best strategy in that case is to simply ask the agent to report *one* bit (even if you are allowed more bits to be transmitted), as this is the optimal value of k in the case of E containing a single subset. These results are complemented with some experiments which show, for the instances of the problem that we studied, that the value of k for which the error is minimized depends logarithmically on the size of E. Finally, we briefly focused on the case of partitions of IN-subsets (where you ask the agent to raise k bits within that set, whatever the bits), which happens to be the generalization of the rule proven to be optimal for $n = 5$ by [2].

There are many ways to develop the line of research initiated with this preliminary work, the first being to refine our understanding of the behaviour of the type of rules discussed here. In particular, we have not precisely studied the influence of the way the subsets of E intersect with each other. We also aim at studying the other sort of debates introduced in [2], in particular the case of sequential debate which look very interesting.

There are also many possible connections to be made with others areas of

research. We just mention here two obvious ones, as a way of conclusion.

As it happens, a persuasion rule is a *set system*, a combinatorial object well studied in the combinatorics litterature, see for instance [1]. However, to the best of our knowledge, the kind of properties that we study here are not classicaly investigated by that community.

Another area of research which seems (at first sight at least) pretty concerned with the problem discussed here is communication complexity. Communication complexity is concerned with the minimal number of bits that need to be exchanged in order to mutually compute some given function [3]. One main difference lies in the fact that agents are cooperative, whereas in our context they try to manipulate the designer to get the result of their wish. Also, communication complexity is typically concerned with finding bounds on the number of bits to be exchanged to be able to compute the function without any possible mistake, whereas here we assume to start with some communication constraints and try to design the rule so as to minimize the errors necessarily induced by these constraints.

Appendix: A Formula for the General Case

In this Appendix, we present the analytical formula of the total number of errors in the general case. As quoted in Section 3.3, we did not manage to derive satisfying bounds for this formula yet.

Suppose that $E = \{e_1, \ldots, e_q\}$ where each e_i is a k-subset. In the following, $|\cup F|$ stands for $|\cup_{f\in F} f|$. Let us first compute c_m, the minority coverage.

$$\begin{aligned} c_m &= \sum_{i=1}^{q} \sum_{x=0}^{\lfloor \frac{n}{2} \rfloor - k} \binom{n-k}{x} - \sum_{i<j} \sum_{x=0}^{\lfloor \frac{n}{2} \rfloor - |e_i \cup e_j|} \binom{n-|e_i \cup e_j|}{i} + \ldots \\ &= \sum_{F \subseteq E, F \neq \emptyset} \left[(-1)^{|F|-1} \sum_{x=0}^{\lfloor \frac{n}{2} \rfloor - |\cup F|} \binom{n-|\cup F|}{x} \right] \end{aligned}$$

$$c_M = \sum_{F \subseteq E, F \neq \emptyset} \left[(-1)^{|F|-1} \sum_{x=\lceil \frac{n}{2} \rceil - |\cup F|}^{n-|\cup F|} \binom{n-|\cup F|}{x} \right]$$

By adding both coverages, we get $c_m + c_M = \sum_{F\subseteq E, F\neq\emptyset} (-1)^{|F|-1} 2^{n-|\cup F|}$. The error is thus $n_{err} = c_m + 2^{n-1} - c_M = 2c_m + 2^{n-1} - \sum_{F\subseteq E, F\neq\emptyset} (-1)^{|F|-1} 2^{n-|\cup F|}$. Simplifying, we get the following general formula:

$$n_{err} - 2^{n-1} = 2 \sum_{F \subseteq E, F \neq \emptyset} (-1)^{|F|} H_n(|\cup F|)$$

Where $H_n(x) = 2^{n-x-1} - \sum_{t=0}^{\lfloor \frac{n}{2} \rfloor - x} \binom{n-x}{t}$.

References

[1] B. Bollobas. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors, and Combinatorial Probability*. Cambridge University Press, 1986.

[2] J. Glazer and A. Rubinstein. Debates and decisions: On a rationale of argumentation rules. *Games and Economic Behaviour*, 36:158–173, 2001.

[3] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[4] A. Rubinstein. *Economics and Language*. Cambridge University Press, 2000.

Yann Chevaleyre
LAMSADE
75775 Paris Cedex 16, France
Email: `yann.chevaleyre@lamsade.dauphine.fr`

Nicolas Maudet
LAMSADE
75775 Paris Cedex 16, France
Email: `maudet@lamsade.dauphine.fr`

On Complexity of Lobbying in Multiple Referenda [1]

Robin Christian, Mike Fellows, Frances Rosamond and Arkadii Slinko

Abstract

In this paper we show that lobbying in conditions of "direct democracy" is virtually impossible, even in conditions of complete information about voters preferences, since it would require solving a very computationally hard problem. We use the apparatus of parametrized complexity for this purpose.

1 Direct and Representative Democracy

Countrywide votes on a specific issue are an accepted way of resolving political issues in many countries around the world. Such votes are usually termed "referenda." A referendum gives the people the chance to vote directly on a specific issue. Although people can also make choices at general elections, these elections are usually fought on a number of issues and often no clear verdict on any one issue is delivered. So instead of voting for only representatives, referenda allow citizens to vote directly on some federal matters. In Switzerland and California, for example, referenda are very common.

It is a commonplace that an ideal democratic political system should combine both referenda and representative government. A key issue is the relative weightings of these two ingredients. Referenda are costly. However, in the fully computerized society, to which we are gradually moving, referenda could be cheap and fast. Hence the relative weightings of the two ingredients may be expected to change.

Another development that might drive this change is the relative simplicity of lobbying such legislative bodies as the American Congress and House of Representatives. In his book, Phillips observes that Washington has become increasingly dominated by an interest-group elite which is now so deeply entrenched and so resistant to change that the proper functioning of government is impossible [20]. He suggests that representative democracy be restored to Athenian direct democracy through the use of referenda.

In this paper we show that lobbying in conditions of "direct democracy" is computationally virtually impossible, even in conditions of complete information about voters' preferences. We use the apparatus of parametrized complexity for this purpose. We envision that computational complexity may play a positive role in voting, protecting the integrity of social choice. Such a role

[1]The research reported in this paper was supported by the New Zealand Marsden Fund, and by the Australian Research Council, through the Australian Centre for Bioinformatics.

would resemble the situation in public-key cryptography [7] where computational complexity protects the privacy of communication. As far as we know, this is the first paper which considers applications of parametrized complexity to social choice. Previously, complexity issues in social choice were considered in [1, 2, 3, 4, 5, 6, 10, 11, 12, 16, 17, 18, 21].

2 Parametrized Complexity

For those not familiar with computational complexity, we provide a quick sketch of concepts and terminology. The reader should consult [8, 14] for more details.

The standard paradigm of complexity theory is embodied in the contrast between P and NP problems. Problems in P are those which admit an algorithm that, given any input x of size n, produces the output $Output(x)$ required by the problem specification in time $O(n^\alpha)$, that is in time bounded by Cn^α, where α and C are constants. The notation P designates the class of problems solvable in polynomial time. Such algorithms are generally considered to be tractable. NP denotes the class of non-deterministic polynomial time solvable problems. For such problems, for each input x, there is a polynomial time algorithm that justifies that $Output(x)$ is indeed the output required by the specifications of the problem. NP contains P and it is believed that $P \neq NP$. The hardest problems in NP are called NP-complete. They are all equivalent in a sense that any such problem can be reduced to an instance of any other NP-complete problem and such reduction can be made in polynomial time. So, if one NP-complete problem can be solved in polynomial time, then all of them can be solved in this way and it would follow that $NP = P$. NP-completeness is therefore taken as evidence of inherent intractability.

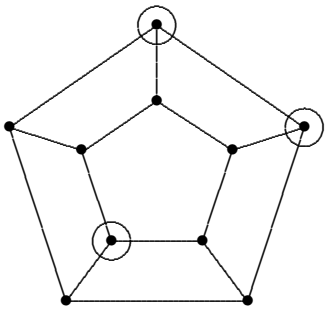
However, in reality we are often interested in the tractability of problems when values of a certain parameter k (representing some aspect of the input) are small. In this case we need to undertake the parametrized complexity analysis as developed by Downey and Fellows in [8]. A problem is said to be in the class FPT (Fixed Parameter Tractable) if there exists an algorithm solving the problem and running in time $f(k)n^c$, where c is a fixed constant and f is an arbitrary computable function. If our problem belongs to this class, then it is tractable for small values of k. Unlike the P versus NP paradigm, here we obtain a hierarchy of parametrized complexity classes

$$FPT = W[0] \subseteq W[1] \subseteq W[2] \subseteq \ldots$$

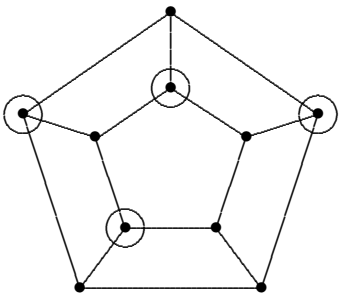
(see [8] for exact definition of these classes). Being $W[2]$-complete is considered strong evidence that the problem is not tractable even for small values of the parameter. Two $W[2]$-complete problems that will be important later in this paper are described below.

Given a graph $G = (V, E)$ with a set of vertices V and the set of edges E, we say that a subset of the set of vertices $V' \subseteq V$ is a *dominating set* if every vertex in V is adjacent to at least one vertex in V'. If V' is dominating and consists

of k vertices we will say that it is a *k-dominating* set. The set V' is called *independent* if no two vertices of V' are adjacent. The picture below shows a 3-dominating set which is not independent and an independent 4-dominating set.



3-dominating set



Independent 4-dominating set

The k-DOMINATING SET problem takes as input a graph G and a positive integer k, which is considered as parameter. The question asks whether there exists a k-dominating set in G. The k-DOMINATING SET problem has been shown to be $W[2]$-complete by Downey and Fellows (1999). They consider that "k-DOMINATING SET problem represents some fundamental 'wall of intractability' where there is no significant alternative to trying all k-subsets for solving the problem." [8], p.15.

The INDEPENDENT k-DOMINATING SET problem is also $W[2]$-complete [8]. The input is the same as for the k-DOMINATING SET, and the question asks whether G has an independent dominating set of size k.

3 Lobbying on a Restricted Budget

We consider the problem faced by an actor that wishes to influence the vote of a certain legislative body or a referendum on a number of issues by trying to exert influence on particular agents. We will refer to this actor as "The Lobby". It is assumed that The Lobby has complete information about agents' preferences. The Lobby has a fixed budget and has to be selective in choosing agents to distribute the limited budget among them. It is reasonable to assume that the number of agents k that can realistically be influenced is relatively small, and hence this aspect of the input is appropriate as a parameter for the complexity analysis. Hence the use of parametrized complexity developed by Downey and Fellows (1999) is completely appropriate for this problem. Our formal model of the problem is as follows:

The problem: Optimal Lobbying (OL)

> *Instance:* An n by m 0/1 matrix $\mathcal{E}$, a positive integer k, and a length m 0/1 vector x. (Each row of $\mathcal{E}$ represents an agent. Each column represents a referendum in the election or a certain issue to be voted on by the legislative body. The 0/1 values in a given row represent the natural inclination of the agent with respect to the referendum questions put to a vote in the election. The vector x represents the outcomes preferred by The Lobby.)
>
> *Parameter:* k (representing the number of agents to be influenced)
>
> *Question:* Is there a choice of k rows of the matrix, such that these rows can be edited so that in each column of the resulting matrix, a majority vote in that column yields the outcome targeted by The Lobby?

Proposition 1. Optimal Lobbying *is* $W[2]$*-hard.*

Proof. One of the standard techniques of proving a problem is $W[2]$-hard is to reduce a problem that is already known to be $W[2]$-hard to our problem. We reduce from the $W[2]$-complete k-Dominating Set problem. Given a graph $G = (V, E)$, and a positive integer k for which we wish to determine whether G has a k-element dominating set, we produce the following set of inputs to the Optimal Lobbying problem. (We will assume that the number of vertices n is odd, and that the minimum degree of G is at least k, since k-Dominating Set remains $W[2]$-complete under these restrictions.)

- The 0/1 matrix $\mathcal{E}$ consists of two sets of rows, the *top set*, indexed by $V = \{1, ..., n\}$, and the *bottom set*, consisting of $n - 2k + 1$ additional rows. The matrix $\mathcal{E}$ has $n + 1$ columns, with the first column being the *template column*, and the remaining n columns indexed by V.
- The template column has 0's in all of the top set row entries, and 1's in all of the bottom set row entries.
- A column indexed by a vertex v, in the top row positions, has 0's in those rows that are indexed by vertices $u \in N[v]$. In the bottom row positions, the entries can be computed by first setting all of these entries to 1, and then changing (arbitrarily) $n - k - |N[v]| + 1$ of these entries to 0. (This ensures that in every column indexed by a vertex the total number of 0's is one more than the total number of 1's.)
- The vector $x = (1, 1, \ldots, 1)$ of length $n + 1$ has a 1 in each position.
- The parameter k remains the same.

We claim that this is a yes-instance of OL if and only if G has a k-dominating set.

One direction is easy. If G has a k-dominating set, then The Lobby corrupts the corresponding agents, or formally, we edit the corresponding rows. With respect to the first (template) column, we thus have the opportunity to change k of the 0's to 1's. Since in the first column, initially, the "1" outcome was losing by $2k-1$ votes, and since each of these k edit operations decreases the *difference* by 2 (as there is one more 1 and one less 0), the outcome in the first (template) column is a victory for the "1" outcome, by 1. Since the chosen rows for editing represent a dominating set in G, we are similarly able to advantage each vertex column contest by at least 2, and since each of these was losing by one vote, we are able to secure majorities of 1 in every column.

Conversely, suppose the described instance of OL has a solution. Necessarily, the rows chosen to be edited must be in the top set of rows (indexed by vertices of G), since otherwise obtaining a majority of 1's in the first column will not be possible. Any solution that consists of rows in the top set of rows must therefore provide at least one opportunity, for each vertex column (indexed by v), of editing in a row that is indexed by a vertex $u \in N[v]$. Thus, any such solution corresponds to a k-dominating set in G. □

Proposition 2. Optimal Lobbying (OL) *is in* $W[2]$.

Proof. One of the standard techniques of proving that a problem is in the class $W[2]$ is to reduce our problem to another problem which is already known to be in $W[2]$. We reduce to the $W[2]$-complete Independent k-Dominating Set problem [8], page 464. Given an n by m 0/1 matrix $\mathcal{E} = (e_{ij})$, a positive integer k, and a length m 0/1 vector x, proceed as follows:

1. Calculate $w = \lfloor n/2 \rfloor + 1$, which is the number of votes required to pass any particular referendum question.

2. For $1 \leq j \leq m$, let
$$\delta(j) = \begin{cases} \max(0, w - \sum_i e_{ij}), & x_j = 1, \\ \max(0, \sum_i e_{ij} - w + 1), & x_j = 0. \end{cases}$$

3. Since $\delta(j)$ is the number of votes that The Lobby is away from the desired outcome in the jth referendum, when $\delta(j) > k$, for at least one j, we have a trivial negative instance.

4. For each $J = 1, \ldots, m$, let $C_j = \{i \mid e_{ij} \neq x_j,\ 1 \leq i \leq n\}$. Then C_j is the set of voters who are naturally inclined to vote against the interests of The Lobby in the jth referendum.

An OL solution of size k will be any set $K \subseteq \{1, \ldots, n\}$ such that the cardinality of K is k and $|K \cap C_j| \geq \delta(j)$ for every $j = 1, \ldots, m$.

Let us construct the graph G as specified below. The vertex set of G consists of the following vertices:

- x_{ab} is a vertex, for $1 \leq a \leq k$, $1 \leq b \leq n$.

- $x_{a\infty}$ is a vertex, for $1 \leq a \leq k$.
- y_{cd} is a vertex, for $1 \leq c \leq m$, $1 \leq d \leq \binom{k}{k-\delta(c)+1}$.

The edges of G are as follows:

- For every $1 \leq a \leq k$, the subgraph induced on $\{x_{ab} \mid 1 \leq b \leq n \text{ or } b = \infty\}$ is complete.
- For every $1 \leq b \leq n$ (but not $b = \infty$) the subgraph induced on $\{x_{ab} \mid 1 \leq a \leq k\}$ is complete.
- For every $1 \leq c \leq m$, let f_c be a bijection from $\{1, 2, \ldots, \binom{k}{k-\delta(c)+1}\}$ to the set of all subsets of $\{1, \ldots, k\}$ of cardinality $k - \delta(c) + 1$. Then the vertex y_{cd} is connected by an edge to each member of $\{x_{ab} \mid a \in f_c(d),\, b \in C_c\}$.

We will show now that G has a k-Independent Dominating Set S if and only if $(\mathcal{E}, k, x)$ is a positive instance of OL. First, assume that G has a k-Independent Dominating Set S. Then each $x_{a\infty}$ is dominated, and, since it is connected only to vertices x_{ab}, where $1 \leq b \leq n$, at least one vertex x_{ab} must be in S for each $1 \leq a \leq k$.As S is of size k, it includes exactly one of the x_{ab} for each a. As S is independent, it cannot include x_{sb} and x_{tb} for $s \neq t$.

Now, let $K = \{b \mid x_{ab} \in S \text{ for some } a\}$. The cardinality of K is k, so, if $|K \cap C_j| \geq \delta(j)$ for every j, then K is an OL solution of size k.

For every j, consider the set $Y_j = \{y_{jd} \mid 1 \leq d \leq \binom{k}{k-\delta(j)+1}\}$. Since each of these vertices is dominated, some member of $\{x_{ab} \mid a \in f_j(d),\, b \in C_j\}$ is in S for each d. Since $f_j(d)$ ranges over all subsets of $\{1, \ldots, k\}$ of cardinality k, at least $\delta(j)$ members of $\{x_{ab} \mid a \in \{1, \ldots, k\}, b \in C_j\}$ are in S and therefore at least $\delta(j)$ members of C_j are in K. Thus K is an OL solution.

Conversely, imagine that K is an OL solution of size k. Choose an arbitrary enumeration θ of elements of K and denote $S = \{x_{i\theta(i)} \mid 1 \leq i \leq k\}$. S is independent, because there is no edge between $x_{i\theta(i)}$ and $x_{j\theta(j)}$ unless $i = j$. Since i ranges over $1, \ldots, k$, each vertex x_{ab} is dominated. Since K is an OL solution, for each j at least $\delta(j)$ members of C_j are in K. Thus, by the construction of S, at least $\delta(j)$ members of $\{x_{ab} \mid a \in \{1, \ldots, k\}, b \in C_j\}$ are in S, so that some member of $\{x_{ab} \mid a \in f_j(d),\, b \in C_j\}$ is in S for each d, and y_{jd} is dominated for each j and each d. Thus S is an Independent Dominating Set of size k. □

Together, the two propositions above give the following complete classification of the parametrized complexity of the problem.

Theorem 1. Optimal Lobbying *is* $W[2]$*-complete.*

4 Conclusion

This paper shows that parameterized complexity is a very appropriate tool for analyzing the computational difficulty of problems in social choice. We believe

that the methods of parameterized complexity will be especially useful when dealing with problems regarding voting and rank aggregation. Any voting situation stipulates the existence of two parameters: the number of voters n and the number of alternatives m. The sizes of these two parameters are very different. While the number of voters can be, and usually is, very large, the number of alternatives is small, seldom exceeding 20. In rank aggregation the situation is virtually the opposite. There we aim to combine several rankings of alternatives into one 'social' ranking. For example, the rankings to be aggregated could be rankings of web pages produced by several search engines [9]. In this case the number of search engines is small and the number of web pages is astronomical. The existence of a small parameter should be reflected in the method of investigation. We believe the best way to do so is to use the conceptual framework of parameterized complexity.

Some 15 years ago, Bartholdi, Tovey and Trick [1] pioneered the study of voting procedures from the viewpoint of complexity theory. In particular, they proved that Dodgson Score and Kemeny Score are NP-complete and Dodgson Winner and Kemeny Winner are NP-hard. The latter two problems were proved to be complete for parallel access to NP [16, 17]. A similar result was also established for Young score and Young winner [21].

It has been known for some time as folklore that the problems Dodgson score and Kemeny Score, as well as Dodgson winner and Kemeny Winner, are Fixed Parameter Tractable if the number of alternatives is chosen as parameter (see, e.g. [19]). The same is true for Young winner [15]. It looks like the number of voters has relatively small impact on complexity in comparison to the number of alternatives. This view is supported by the fact that Kemeny ranking remains NP-complete even for four voters [9].

It may also happen that the two obvious parameters — the number of alternatives and the number of voters — are not the most natural parameters for measuring the exact complexity of such problems. In this respect we note that the parametrized complexity of Dodgson Score (and similarly Dodgson Winner) in the following formulation remains open and is of considerable interest.

The problem: Dodgson Score (DS)

> *Instance:* Set of candidates A, and a distinguished member $a \in A$; a profile of preference orders on A.
>
> *Parameter:* k (representing the bound for the Dodgson's score)
>
> *Question:* Is the Dodgson score of candidate a less then or equal to k?

This is a parametrized version of the original question studied by Bartholdi et al [1].

5 Acknowledgements

The authors thank Prof. Lane Hemaspaandra for his encouraging open review of this paper and helpful comments, especially bringing to our attention the paper [13].

References

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

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

[3] Bartholdi, J.J.,III, Tovey, C.A., and Trick, M.A. (1992) How hard is to control an election? *Mathematical and Computer Modelling* 16 (8/9): 27–40.

[4] Bartholdi, J.J.,III, Narasimhan, L.S, and Tovey, C.A. (1991) Recognizing majority rule equilibrium in spatial voting. *Social Choice and Welfare* 8: 183–197.

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

[6] Conitzer, V., and Sandholm T. (2002) Complexity of Manipulating Elections with Few Candidates. In: *Proceedings of the National conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, 2002, to appear; available at http://www.cs.cmu.edu/ sandholm.

[7] Diffe, W., and Hellman, M. (1976) New directions in cryptography. *IEEE Transactions on Information Theory* IT–22: 644–654.

[8] Downey, R.G, and Fellows, M.R. (1999) *Parametrized complexity.* New York, Springer.

[9] Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001) Rank aggregation methods for the web. *WWW*: 613–622.

[10] Ephrati, E. (1994) A non-manipulable meeting scheduling system. In: *Proceedings 13th International Distributed Artificial Intelligence Workshop*, Lake Quinalt, Washington, July, AAAI Press Technical Report WS-94-02.

[11] Ephrati, E., and Rosenschein, J. (1991) The Clarke tax as a consensus mechanism among automated agents. In: *Proceedings of the National conference on Artificial Intelligence (AAAI)*, 173–178. Anaheim, CA.

[12] Ephrati, E., and Rosenschein, J. (1993) Multi-agent planning as a dynamic search for social consensus. In: *Proceedings 13th International Joint Conference on Artificial Intelligence (IJCAI)*, 423–429. Chambery, France.

[13] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L, and Rothe, J. (2006) A richer understanding of the complexity of election systems. Available at http://www.arxiv.org/abs/cs.GT/0609112

[14] Garey, M., and Johnson, D. (1979) *Computers and Intractability: A Guide to The Theory of NP-completeness*. San Francisco, Freeman.

[15] Hemaspaandra, L. (2006) Private communication.

[16] Hemaspaandra, E., and Hemaspaandra, L. (2000) Computational politics: electoral systems. In: *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science*, pages 64-83. Springer-Verlag Lecture Notes in Computer Science #1893, August/September.

[17] Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. (1997) 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.

[18] Hemaspaandra, E., Spakovski, H., and Vogel, J. (2005) The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3): 383–391.

[19] McCabe-Dansted, J. (2006) Approximability and Computational Feasibility of Dodgson's Rule. Master's thesis. The University of Auckland.

[20] Phillips, K. (1994) *Arrogant Capital*. Little, Brown and Company.

[21] Rothe, J., Spakovski, H., and Vogel, J. (2003) Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4): 375–386.

Robin Christian
Department of Combinatorics and Optimization
University of Waterloo
200 University Avenue West
Waterloo, Ontario N2L 3G1, Canada
Email: `r3christ@math.uwaterloo.ca`

Mike Fellows
Parameterized Complexity Research Unit,
Office of DVC(Research),
The University of Newcastle
Calaghan NSW 2308, Australia
and
The Australian Centre for Bioinformatics
Email: `mfellows@cs.newcastle.edu.au`

Frances Rosamond
Parameterized Complexity Research Unit,
Office of DVC(Research),
The University of Newcastle
Calaghan NSW 2308, Australia
and
The Australian Centre for Bioinformatics
Email: fran@cs.newcastle.edu.au

Arkadii Slinko
Department of Mathematics
The University of Auckland
Private Bag 92019, Auckland, New Zealand
Email: a.slinko@auckland.ac.nz

Eliciting Single-Peaked Preferences Using Comparison Queries

Vincent Conitzer

Abstract

Voting is a general method for aggregating the preferences of multiple agents. Each agent ranks all the possible alternatives, and based on this, an aggregate ranking of the alternatives (or at least a winning alternative) is produced. However, when there are many alternatives, it is impractical to simply ask agents to report their complete preferences. Rather, the agents' preferences, or at least the relevant parts thereof, need to be *elicited*. This is done by asking the agents a (hopefully small) number of simple queries about their preferences, such as *comparison* queries, which ask an agent to compare two of the alternatives. Prior work on preference elicitation in voting has focused on the case of unrestricted preferences. It has been shown that in this setting, it is sometimes necessary to ask each agent (almost) as many queries as would be required to determine an arbitrary ranking of the alternatives. By contrast, in this paper, we focus on single-peaked preferences. We show that such preferences can be elicited using only a linear number of comparison queries, if either the order with respect to which preferences are single-peaked is known, or at least one other agent's complete preferences are known. We also show that using a sublinear number of queries will not suffice. Finally, we present experimental results.

1 Introduction

In multiagent systems, a group of agents often has to make joint decisions even when the agents have conflicting preferences over the alternatives. For example, agents may have different preferences over possible joint plans for the group, allocations of tasks or resources among members of the group, potential representatives (*e.g.* presidential candidates), *etc.* In such settings, it is important to be able to *aggregate* the agents' individual preferences. The result of this aggregation can be a single alternative, corresponding to the group's collective decision, or a complete aggregate (compromise) ranking of all the alternatives (which can be useful, for instance, if some of the alternatives later turn out not to be feasible). The most general framework for aggregating the agents' preferences is to have the agents *vote* over the alternatives. That is, each agent announces a complete ranking of all alternatives (the agent's *vote*), and based on these votes an outcome (*i.e.* a winning alternative or a complete aggregate ranking of all alternatives) is chosen according to some *voting rule*.[1]

[1] One may argue that this approach is not fully general because it does not allow agents to specify their preferences over *probability distributions* over alternatives. For example, it is

One might try to create an aggregate ranking as follows: for given alternatives a and b, if more votes prefer a to b than vice versa (*i.e.* a wins its *pairwise election* against b), then a should be ranked above b in the aggregate ranking. Unfortunately, when the preferences of the agents are unrestricted and there are at least three alternatives, *Condorcet cycles* may occur. A Condorcet cycle is a sequence of alternatives $a_1, a_2, \ldots, a_k$ such that for each $1 \leq i < k$, more agents prefer a_i to a_{i+1} than vice versa, and more agents prefer a_k to a_1 than vice versa. In the presence of a Condorcet cycle, it is impossible to produce an aggregate ranking that is consistent with the outcomes of all pairwise elections. Closely related to this phenomenon are numerous impossibility results that show that every voting rule has significant drawbacks in this general setting. For example, when there are at least three alternatives, Arrow's impossibility theorem [Arrow, 1963] shows that any voting rule for which the relative order of two alternatives in the aggregate ranking is independent of how agents rank alternatives other than these two (*i.e.* any rule that satisfies *independence of irrelevant alternatives*) must either be *dictatorial* (*i.e.* the rule simply copies the ranking of a fixed agent, ignoring all other agents) or conflicting with *unanimity* (*i.e.* for some alternatives a and b, the rule sometimes ranks a above b even if all agents prefer b to a). As another example, when there are at least three alternatives, the Gibbard-Satterthwaite theorem [Gibbard, 1973; Satterthwaite, 1975] shows that for any voting rule that is *onto* (for every alternative, there exist votes that would make that alternative win) and nondictatorial, there are instances where an agent is best off casting a vote that does not correspond to the agent's true preferences (*i.e.* the rule is not *strategy-proof*).

1.1 Single-peaked preferences

Fortunately, these difficulties can disappear if the agents' preferences are restricted, *i.e.* they display some structure. The best-known, and arguably most important such restriction is that of *single-peaked preferences* [Black, 1948]. Suppose that the alternatives are ordered on a line, from left to right, representing the alternatives' *positions*. For example, in a political election, a candidate's position on the line may indicate whether she is a left-wing or a right-wing candidate (and how strongly so). As another example, the alternatives may be numerical values: for example, agents may vote over the size of a budget. As yet another example, the alternatives may be locations along a road (for example, if agents are voting over where to construct a building, or where to meet for dinner, *etc.*). We say that an agent's preferences are *single-peaked* with respect to the alternatives' positions if, on each side of the agent's most preferred alternative (the agent's *peak*), the agent prefers alternatives that are closer to its peak. For example, if the set of alternatives is $\{a, b, c, d, e, f\}$, their

impossible to know from an agent's vote whether that agent prefers its second-ranked alternative to a 1/2 - 1/2 probability distribution over its first-ranked and third-ranked alternatives. In principle, this can be addressed by voting over these probability distributions instead, although in practice this is usually not tractable.

positions may be represented by $d < b < e < f < a < c$, in which case the vote $f \succ e \succ b \succ a \succ c \succ d$ is single-peaked, but the vote $f \succ e \succ a \succ d \succ c \succ b$ is not (b and d are on the same side of f in the positions, and b is closer to f, so d should not be ranked higher than b if f is the peak). (Throughout, we will assume that all preferences are strict, that is, agents are never indifferent between two alternatives.) Preferences are likely to be single-peaked if the alternatives' positions are of primary importance in determining an agent's preferences. For example, in political elections, if voters' preferences are determined primarily by candidates' proximity to their own stance on the left-to-right spectrum, preferences are likely to be single-peaked. On the other hand, if other factors are also important, such as the perceived amicability of the candidates, then preferences are not necessarily likely to be single-peaked.

When all agents' preferences are single-peaked (with respect to the same positions for the alternatives), it is known that there can be no Condorcet cycles. If, in addition, we assume that the number of agents is odd, then no pairwise election can result in a tie. Hence, our aggregate ranking can simply correspond to the outcomes of the pairwise elections. In this case, there is also no incentive for an agent to misreport its preferences, since by reporting its preferences truthfully, it will, in each pairwise election, rank the more desired alternative higher.

1.2 Preference elicitation

A key difficulty in aggregating the preferences of multiple agents is the *elicitation* of the agents' preferences. In many settings, particularly those with large sets of alternatives, having each agent communicate all of its preferences is impractical. For one, it can take up a large amount of communication bandwidth. Perhaps more importantly, in order for an agent to communicate all of its preferences, it must first *determine* exactly what those preferences are. This can be a complex task, especially when no guidance is provided to the agent as to what the key questions are that it needs to answer to determine its preferences.

An alternative approach is for an *elicitor* to sequentially ask the agents certain natural *queries* about their preferences. For example, the elictor can ask an agent which of two alternatives it prefers (a *comparison query*). Three natural goals for the elicitor are to (1) learn enough about the agents' preferences to determine the winning alternative, (2) learn enough to determine the entire aggregate ranking, and (3) learn each agent's complete preferences. (1) and (2) have the advantage that in general, not all of each agent's preferences need to be determined. For example, for (1), the elicitor does not need to elicit an agent's preferences among alternatives for which we have already determined (from the other agents' preferences) that they have no chance of winning. But even (3) can have significant benefits over not doing any elicitation at all (*i.e.* having each agent communicate all of its preferences on its own). First, the elicitor provides the agent with a systematic way of assessing its preferences: all that the agent needs to do is answer simple queries. Second, and perhaps more

importantly, once the elicitor has elicited the preferences of some agents, the elicitor will have some understanding of which preferences are more likely to occur (and, perhaps, some understanding of why this is so). The elicitor can then use this understanding to guide the elicitation of the next agent's preferences, and learn these preferences more rapidly.

In this paper, we will study the elicitation of single-peaked preferences using only comparison queries. We will focus on approach (3), *i.e.* learning each agent's complete preferences. We will study both the setting where the elicitor knows the positions of the alternatives (Section 4), and the setting where the elicitor (at least initially) does not (Section 5). We will assume that preferences are always single-peaked.[2] Our elicitation algorithms completely elicit one agent's preferences before moving on to the next agent (as opposed to going back and forth between agents). This gives the algorithms a nice online property: if agents arrive over time, then we can elicit an agent's preferences when it arrives, after which the agent is free to leave and pursue other things (as opposed to being forced to wait until the arrival of the next agent).

2 Related research

A significant body of work on preference elicitation in multiagent systems focuses on combinatorial auctions (for an overview of this work, see Sandholm and Boutilier [2006]). Much of this work focuses on approach (1), *i.e.* learning enough about the bidders' valuations to determine the optimal allocation. (Sometimes, additional information must be elicited from the bidders to determine the payments that they should make according to the Clarke [Clarke, 1971], or more generally, a Groves [Groves, 1973], mechanism.) Example elicitation approaches include ascending combinatorial auctions (for an overview, see Parkes [2006]) as well as frameworks in which the auctioneer can ask queries in a more flexible way [Conen and Sandholm, 2001]. A significant amount of the research on preference elicitation in combinatorial auctions is also devoted to approach (3), *i.e.* learning an agent's complete valuation function. In this research, typically valuation functions are assumed to lie in a restricted class, and given this it is shown that an agent's complete valuation function can be elicited using a polynomial number of queries of some kind. Various results of this nature have been obtained by Zinkevich *et al.* [2003], Blum *et al.* [2004], Lahaie and Parkes [2004], and Santi *et al.* [2004].

There has also been some work on elicitation in voting settings (the setting of this paper). All of that work so far has focused on approach (1), eliciting enough information from the agents to determine the winner, without

[2] We note that if it is possible that some agent's preferences are not single-peaked, we can always elicit them as if they were, and then *verify* that we have learned them correctly using an additional $m-1$ comparison queries. This is done by asking the agent whether it prefers (what we think is) its most preferred alternative to its second-most preferred alternative, its second-most preferred alternative to its third-most preferred alternative, *etc.* If this verification step fails, we can use some other method to re-elicit the agent's preferences.

any restriction on the space of possible preferences. Conitzer and Sandholm [2002] studied the complexity of deciding whether enough information has been elicited to declare a winner, as well as the complexity of choosing which votes to elicit given very strong suspicions about how agents will vote. They also studied what additional opportunities for strategic misreporting of preferences elicitation introduces, as well as how to avoid introducing these opportunities. (Strategic misreporting is not a significant concern in the setting of this paper: under the restriction of single-peaked preferences, reporting truthfully is a dominant strategy when agents simultaneously report their complete preferences, and hence responding truthfully to the elicitor's queries is an *ex-post* equilibrium. As such, in this paper we will make no distinction between an agent's vote and its true preferences.) Conitzer and Sandholm [2005] studied elicitation algorithms for determining the winner under various voting rules (without any suspicion about how agents will vote), and gave lower bounds on the worst-case amount of information that agents must communicate.

3 Eliciting general preferences

As a basis for comparison, let us first analyze how difficult it is to elicit arbitrary (not single-peaked) preferences using comparison queries. We recall that our goal is to extract the agent's complete preferences, *i.e.* we want to know the agent's exact ranking of all m alternatives. This is exactly the same problem as that of sorting a set of m elements, when only binary comparisons between elements can be used to do the sorting. This is an extremely well-studied problem, and it is well-known that it can be solved using $O(m \log m)$ comparisons, for example using the MergeSort algorithm (which splits the set of elements into two halves, solves each half recursively, and then merges the solutions using a linear number of comparisons). It is also well-known that $\Omega(m \log m)$ comparisons are required (in the worst case). One way to see this is that there are $m!$ possible orders, so that an order encodes $\log(m!)$ bits of information—and $\log(m!)$ is $\Omega(m \log m)$. Hence, in general, *any* method for communicating an order (not just methods based on comparison queries) will require $\Omega(m \log m)$ bits (in the worst case).

Interestingly, for some common voting rules (including Borda, Copeland, and Ranked Pairs), it can be shown using techniques from communication complexity theory that even just determining whether a given alternative is the winner requires the communication of $\Omega(nm \log m)$ bits (in the worst case), where n is the number of agents [Conitzer and Sandholm, 2005]. That is, even if we do not try to elicit agents' complete preferences, (in the worst case) it is impossible to do more than a constant factor better than having each agent communicate all of its preferences! These lower bounds even hold for *nondeterministic* communication, but they do assume that preferences are unrestricted. By contrast, by assuming that preferences are single-peaked, we can elicit an agent's *complete* preferences using only $O(m)$ queries, as we will show in this

paper. Of course, once we know the agents' complete preferences, we can execute any voting rule. This shows how useful it can be for elicitation to know that agents' preferences lie in a restricted class.

4 Eliciting with knowledge of alternatives' positions

In this section, we focus on the setting where the elicitor knows the positions of the alternatives. Let $p : \{1, \ldots, m\} \rightarrow A$ denote the mapping from positions to alternatives, *i.e.* $p(1)$ is the leftmost alternative, $p(2)$ is the alternative immediately to the right of $p(1)$, ..., and $p(m)$ is the rightmost alternative. Our algorithms make calls to the function $\mathsf{Query}(a_1, a_2)$, which returns *true* if the agent whose preferences we are currently eliciting prefers a_1 to a_2, and *false* otherwise. (Since one agent's preferences are elicited at a time, we do not need to specify which agent is being queried.)

The first algorithm serves to find the agent's peak (most preferred alternative). The basic idea of this algorithm is to do a binary search for the peak. To do so, we need to be able to assess whether the peak is to the left or right of a given alternative a. We can discover this by asking whether the alternative immediately to the right of a is preferred to a: if it is, then the peak must be to the right of a, otherwise, the peak must be to the left of, or equal to, a.

```
FindPeakGivenPositions(p)

l ← 1
r ← m
while l < r {
  m1 ← ⌊(l + r)/2⌋
  m2 ← m1 + 1
...
```

```
...
  if Query(p(m1), p(m2))
    r ← m1
  else
    l ← m2
}
return l
```

Once we have found the peak, we can continue to construct the agent's ranking of the alternatives as follows. We know that the agent's second-ranked alternative must be either the alternative immediately to the left of the peak, or the one immediately to the right. A single query will settle which one is preferred. Without loss of generality, suppose the left alternative was preferred. Then, the third-ranked alternative must be either the alternative immediately to the left of the second-ranked alternative, or the alternative immediately to the right of the peak. Again, a single query will suffice—*etc.* Once we have determined the ranking of either the leftmost or the rightmost alternative, we can construct the remainder of the ranking without asking any more queries (by simply ranking the remaining alternatives according to proximity to the peak). The algorithm is formalized below. It uses the function $\mathsf{Append}(a_1, a_2)$, which makes a_1 the alternative that immediately succeeds a_2 in the current ranking

(*i.e.* the current agent's preferences as far as we have constructed them). In the pseudocode, we will omit the (simple) details of maintaining such a ranking as a linked list. The algorithm returns the highest-ranked alternative; this is to be interpreted as including the linked-list structure, so that effectively the entire ranking is returned. c is always the alternative that is ranked last among the currently ranked alternatives.

```
FindRankingGivenPositions(p)
t ← FindPeakGivenPositions(p)
s ← p(t)
l ← t − 1
r ← t + 1
c ← s
while l ≥ 1 and r ≤ m {
  if Query(p(l), p(r)) {
    Append(p(l), c)
    c ← p(l)
    l ← l − 1
  } else {
    Append(p(r), c)
    c ← p(r)
...
```

```
...
    r ← r + 1
  }
}
while l ≥ 1 {
  Append(p(l), c)
  c ← p(l)
  l ← l − 1
}
while r ≤ m {
  Append(p(r), c)
  c ← p(r)
  r ← r + 1
}
return s
```

Theorem 1 FindRankingGivenPositions *requires at most* $m - 2 + \lceil \log m \rceil$ *comparison queries.*

Proof: FindPeakGivenPositions requires at most $\lceil \log m \rceil$ comparison queries. Every query after this allows us to add an additional alternative to the ranking, and for the last alternative we will not need a query, hence there can be at most $m - 2$ additional queries. ∎

Thus, the number of queries that the algorithm requires is linear in the number of alternatives. It is impossible to succeed using a sublinear number of queries, because an agent's single-peaked preferences can encode a linear number of bits, as follows. Suppose the alternatives' positions are as follows: $a_{m-1} < a_{m-3} < a_{m-5} < \ldots < a_4 < a_2 < a_1 < a_3 < a_5 < \ldots < a_{m-4} < a_{m-2} < a_m$. Then, any vote of the form $a_1 \succ \{a_2, a_3\} \succ \{a_4, a_5\} \succ \ldots \succ \{a_{m-1}, a_m\}$ (where the set notation indicates that there is no constraint on the preference between the alternatives in the set, that is, $\{a_i, a_{i+1}\}$ can be replaced either by $a_i \succ a_{i+1}$ or $a_{i+1} \succ a_i$) is single-peaked with respect to the alternatives' positions. The agent's preference between alternatives a_i and a_{i+1} (for even i) encodes a single bit, hence the agent's complete preferences encode $(m-1)/2$ bits. Since the answer to a comparison query can communicate only a single bit of information, it follows that a linear number of queries is in fact necessary.

5 Eliciting without knowledge of alternatives' positions

In this section, we study a more difficult question: how hard is it to elicit the agents' preferences when the alternatives' positions are not known? Certainly, it would be desirable to have elicitor software that does not require us to enter domain-specific information (namely, the positions of the alternatives) before elicitation begins, for two reasons: (1) this information may not be available to the entity running the election, and (2) entering this information may be perceived by agents as unduly influencing the process, and perhaps the outcome, of the election. Rather, the software should *learn* (relevant) information about the domain from the elicitation process itself.

It is clear that this learning will have to take place over the process of eliciting the preferences of multiple agents. Specifically, without any knowledge of the positions of the alternatives, the first agent's preferences could be *any* ranking of the alternatives, since any ranking is single-peaked with respect to some positions. Hence, eliciting the first agent's preferences will require $\Omega(m \log m)$ queries. Once the elicitor knows the first agent's preferences, though, some ways in which the alternatives may be positioned will be eliminated (but many will remain).

Can the elicitor learn the exact positions of the alternatives? The answer is no, for several reasons. First of all, we can invert the positions of the alternatives, making the leftmost alternative the rightmost, *etc.*, without affecting which preferences are single-peaked with respect to these positions. This is not a fundamental problem because the elicitor could choose either one of the positionings. More significantly, the agents' preferences may simply not give the elicitor enough information to determine the positions. For example, if all agents turn out to have the same preferences, the elicitor will never learn anything about the alternatives' positions beyond what was learned from the first agent. In this case, however, the elicitor could simply try to verify that the next agent whose preferences are to be elicited has the same preferences, which can be done using only a linear number of queries. More generally, one might imagine an intricate elicitation scheme which *either* requires few queries to elicit an agent's preferences, *or* learns something new and useful from these preferences that will shorten the elicitation process for later agents. Then, one might imagine a complex accounting scheme, in the spirit of amortized analysis, showing that the total elicitation cost over many agents cannot be too large.

Fortunately, it turns out that we do not need anything so complex. In fact, knowing even *one* agent's (complete) preferences is enough to elicit any other agent's preferences using only a linear number of queries! (And a sublinear number will not suffice, since we already showed that a linear number is necessary even if we know the alternatives' positions.) To prove this, we will give an elicitation algorithm that takes as input one (the first) agent's preferences (*not* the positions of the alternatives), and elicits another agent's preferences using a linear number of queries.

First, we need a subroutine for finding the agent's peak. We cannot use the algorithm FindPeakGivenPositions from the previous section, since we do not know the positions. However, even the trivial algorithm that examines the alternatives one by one and maintains the most-preferred alternative so far requires only a linear number of queries, so we will simply use this algorithm.

```
FindPeak()

s ← a_1
for all a ∈ {a_2, ..., a_m}
  if Query(a, s)
    s ← a
return s
```

Once we have found the agent's peak, we next find the alternatives that lie between this peak, and the peak of the known vote (*i.e.* the peak of the agent whose preferences we know). The following lemma is the key tool for doing so.

Lemma 1 *Consider votes v_1 and v_2 with peaks s_1 and s_2, respectively. Then, an alternative $a \notin \{s_1, s_2\}$ lies between the two peaks if and only if both $a \succ_{v_1} s_2$ and $a \succ_{v_2} s_1$.*

Proof: If a lies between the two peaks, then for each i, a lies closer to s_i than s_{3-i} (the other vote's peak) lies to s_i. Hence $a \succ_{v_1} s_2$ and $a \succ_{v_2} s_1$. Conversely, $a \succ_{v_i} s_{3-i}$ implies that a lies on the same side of s_{3-i} as s_i (otherwise, v_i would have ranked s_{3-i} higher). But since this is true for both i, it implies that a must lie between the peaks. ■

Thus, to find the alternatives between the peak of the known vote and the peak of the current agent, we simply ask the current agent, for each alternative that the known vote prefers to the peak of the current agent, whether it prefers this alternative to the known vote's peak. If the answer is positive, we add the alternative to the list of alternatives between the peaks.

The two votes must rank the alternatives between their peaks in the exact opposite order. Thus, at this point, we know the current agent's preferences over the alternatives that lie between its peak and the peak of the known vote (including the peaks themselves). The final and most complex step is to integrate the remaining alternatives into this ranking. (Some of these remaining alternatives may be ranked higher than some of the alternatives between the peaks.) The strategy will be to integrate these alternatives into the current ranking one by one, in the order in which the known vote ranks them, starting with the one that the known vote ranks highest. When integrating such an alternative, we first have the current agent compare it to the worst-ranked alternative already in the ranking. We note that the *known* vote must prefer the latter alternative, because this latter alternative is either the known vote's

peak, or an alternative that we integrated earlier and that was hence preferred by the known vote. If the latter alternative is also preferred by the current agent, we add the new alternative to the bottom of the current ranking and move on to the next alternative. If not, then we learn something useful about the positions of the alternatives, namely that the new alternative lies on the *other side* of the current agent's peak from the alternative currently ranked last. The following lemma proves this.

Lemma 2 *Consider votes v_1 and v_2 with peaks s_1 and s_2, respectively. Consider two alternatives $a_1, a_2 \neq s_2$ that do* not *lie between s_1 and s_2. Suppose $a_1 \succ_{v_1} a_2$ and $a_2 \succ_{v_2} a_1$. Then, a_1 and a_2 must lie on opposite sides of s_2.*

Proof: If a_1 and a_2 lie on the same side of s_2—without loss of generality, the left side—then, because neither lies between s_1 and s_2, they must also both lie on the left side of s_1 (possibly, one of them is equal to s_1). But then, v_1 and v_2 cannot disagree on which of a_1 and a_2 is ranked higher. ∎

Knowing that the new alternative lies on the other side of the current agent's peak from the currently worst-ranked alternative will not help us to integrate the new alternative; in fact, our algorithm may still have to ask the agent to compare the new alternative to every alternative in the current ranking (other than the peak and the currently worst-ranked alternative). However, once we have integrated the new alternative, we know that *all alternatives that we integrate later must end up ranked below this alternative*. This is because of the following reason. Let us refer to the newly integrated alternative as c_1, and to the currently worst-ranked alternative as c_2. Because we have already taken care of the alternatives between the peaks of the current agent and the known vote, any later alternative that we integrate must lie on the same side of both peaks, on the same side as one of the two c_i. Because we are integrating alternatives in the order in which they are ranked by the known vote, the new (later) alternative must be further from the known vote's peak than that c_i. Hence, it must also be further from the current agent's peak than that c_i, so it must be ranked below c_1 by the current agent (since c_1 is ranked higher than c_2).

We now present the algorithm formally. The algorithm again uses the function $\mathsf{Append}(a_1, a_2)$, which makes a_1 the alternative that immediately succeeds a_2 in the current ranking. It also uses the function $\mathsf{InsertBetween}(a_1, a_2, a_3)$, which inserts a_1 between a_2 and a_3 in the current ranking. The algorithm will (eventually) set $m(a)$ to *true* if a lies between the peaks of the current agent and the known vote v, or if a is the peak of v; otherwise, $m(a)$ is set to *false*. $v(i)$ returns the alternative that the known vote ranks ith (and hence $v^{-1}(a)$ returns the ranking of alternative a in the known vote, and $v^{-1}(a_1) < v^{-1}(a_2)$ means that v prefers a_1 to a_2). $n(a)$ returns the alternative immediately following a in the current ranking. Again, only the peak is returned, but this includes the linked-list structure and hence the entire ranking.

```
FindRankingGivenOtherVote(v)

s ← FindPeak()
for all a ∈ A
  m(a) ← false
for all a ∈ A − {s, v(1)}
  if v^{-1}(a) < v^{-1}(s)
    if Query(a, v(1))
      m(a) ← true
c_1 ← s
c_2 ← s
m(v(1)) ← true
for i = m to 1 step −1 {
  if m(v(i)) = true {
    Append(v(i), c_2)
...
```

```
...
    c_2 ← v(i)
  }
}
for i = 1 to m
  if not (m(v(i)) or v(i) = s)
    if Query(c_2, v(i)) {
      Append(v(i), c_2)
      c_2 ← v(i)
    } else {
      while Query(n(c_1), v(i))
        c_1 ← n(c_1)
      InsertBetween(v(i), c_1, n(c_1))
      c_1 ← v(i)
    }
return s
```

Theorem 2 FindRankingGivenOtherVote *requires at most* $4m - 6$ *comparison queries.*

Proof: FindPeak requires $m-1$ comparison queries. The next stage, discovering which alternatives lie between the current agent's peak and the known vote's peak, requires at most $m - 2$ queries. Finally, we must count the number of queries in the integration step. This is more complex, because integrating one alternative (which we may have to do up to $m - 2$ times) can require multiple queries. Certainly, the algorithm will ask the agent to compare the alternative currently being integrated to the current c_2. This contributes up to $m - 2$ queries in total. However, if the current alternative is preferred over c_2, we must ask more queries, comparing the current alternative to the alternative currently ranked immediately behind the current c_1 (perhaps multiple times). But every time that we ask such a query, c_1 changes to another alternative, and this can happen at most $m - 1$ times in total. ■

In practice, the algorithm ends up requiring on average roughly $3m$ queries, as we will see in Section 6.

6 Experimental results

The following experiment compares FindRankingGivenPositions, FindRankingGivenOtherVote, and MergeSort. As discussed in Section 3, MergeSort is a standard sorting algorithm that uses only comparison queries, and can therefore be used to elicit an agent's preferences without any knowledge of the alternatives' positions or of other votes.

In each run, first a random permutation of the m alternatives was drawn to represent the positions of the alternatives. Then, two random votes (rankings)

that were single-peaked with respect to these positions were drawn. For each vote, this was done by randomly choosing a peak, then randomly choosing the second-highest ranked alternative from the two adjacent alternatives, *etc.* Each algorithm then elicited the second vote; FindRankingGivenPositions was given (costless) access to the positions, and FindRankingGivenOtherVote was given (costless) access to the first vote. (For each run, it was also verified that each algorithm produced the correct ranking.) Figure 1 shows the results.

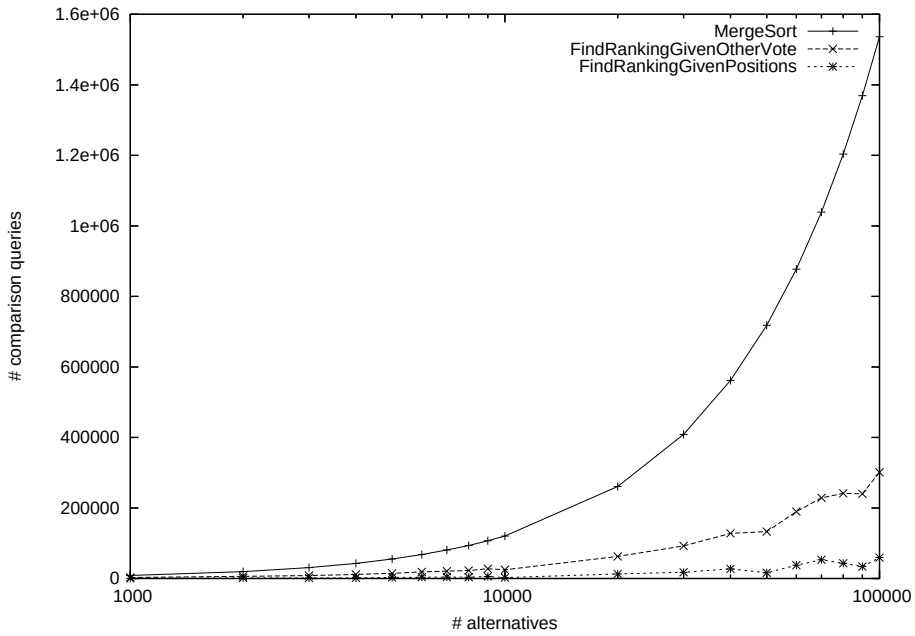


Figure 1: Experimental comparison of the two algorithms introduced in this paper, and MergeSort. Please note the logarithmic scale on the x-axis. Each data point is averaged over 5 runs.

FindRankingGivenPositions outperforms FindRankingGivenOtherVote, which in turn clearly outperforms MergeSort.

One interesting observation is that FindRankingGivenOtherVote sometimes repeats a query that it has asked before. Thus, by simply storing the results of previous queries, the number of queries can be reduced. However, in general, keeping track of which queries have been asked imposes a significant computational burden, as there are $\binom{m}{2}$ possible comparison queries. Hence, in the experiment, the results of previous queries were not stored. FindRankingGivenPositions and MergeSort never repeat a query.

7 Conclusions

Voting is a general method for aggregating the preferences of multiple agents. Each agent ranks all the possible alternatives, and based on this, an aggregate ranking of the alternatives (or at least a winning alternative) is produced. However, when there are many alternatives, it is impractical to simply ask agents to report their complete preferences. Rather, the agents' preferences, or at least the relevant parts thereof, need to be elicited. This is done by asking the agents a (hopefully small) number of simple queries about their preferences, such as comparison queries, which ask an agent to compare two of the alternatives. Prior work on preference elicitation in voting has focused on the case of unrestricted preferences. It has been shown that in this setting, it is sometimes necessary to ask each agent (almost) as many queries as would be required to determine an arbitrary ranking of the alternatives. By contrast, in this paper, we focused on single-peaked preferences. The agents' preferences are said to be single-peaked if there is some fixed order of the alternatives the alternatives' *positions* (representing, for instance, which alternatives are more "left-wing" and which are more "right-wing"), such that each agent prefers alternatives that are closer to the agent's most preferred alternative to ones that are further away. We first showed that if an agent's preferences are single-peaked, and the alternatives' positions are known, then the agent's (complete) preferences can be elicited using a linear number of comparison queries. If the alternatives' positions are not known, then the first agent's preferences can be arbitrary and therefore cannot be elicited using only a linear number of queries. However, we showed that if we already know at least one other agent's preferences, then we can elicit the (next) agent's preferences using a linear number of queries (albeit a larger number of queries than the first algorithm). We also showed that using a sublinear number of queries will not suffice. Experimental results confirmed that these algorithms outperform algorithms that do not make use of the alternatives' positions or of previously elicited agents' preferences.

Future research includes studying elicitation in voting for other restricted classes of preferences. The class of single-peaked preferences (over single-dimensional domains) was a natural one to study first, due to both its practical relevance (real-world preferences often have this structure) and its useful theoretical properties (no Condorcet cycles and, as a result, the ability to aggregate preferences in a strategy-proof manner). Classes that are practically relevant but do not have these nice theoretical properties are still of interest, though. For example, one may consider settings where alternatives take positions in two-dimensional rather than single-dimensional space. It is well-known that in this generalization, Condorcet cycles can once again occur. Nevertheless, this does not imply that efficient elicitation algorithms do not exist for this setting. Nor does it imply that such elicitation algorithms would be useless, since it is still often necessary to vote over alternatives in such settings. However, if we use a voting rule that is not strategy-proof, then we must carefully evaluate the strategic effects of elicitation. Specifically, from the queries that agents

are asked, they may be able to infer something about how other agents answered queries before them; this, in turn, may affect how they (strategically) choose to answer their own queries, since the rule is not strategy-proof. (This phenomenon is studied in more detail by Conitzer and Sandholm [2002].)

References

Kenneth Arrow. *Social choice and individual values*. New Haven: Cowles Foundation, 2nd edition, 1963. 1st edition 1951.

Duncan Black. On the rationale of group decision-making. *Journal of Political Economy*, 56(1):23–34, 1948.

Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning. *JMLR*, 5:649–667, 2004.

Ed H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. *ACM-EC*, pages 256–259, 2001.

Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. *AAAI*, pages 392–397, 2002.

Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. *ACM-EC*, pages 78–87, 2005.

Allan Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

Sebastién Lahaie and David Parkes. Applying learning algorithms to preference elicitation. *ACM-EC*, pages 180–188, 2004.

David Parkes. Iterative combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 3. MIT Press, 2006.

Tuomas Sandholm and Craig Boutilier. Preference elicitation in combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 10, pages 233–263. MIT Press, 2006.

Paolo Santi, Vincent Conitzer, and Tuomas Sandholm. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. *COLT*, pages 1–16, 2004.

Mark 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.

Martin Zinkevich, Avrim Blum, and Tuomas Sandholm. On polynomial-time preference elicitation with value queries. *ACM-EC*, pages 176–185, 2003.

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

How to Allocate Hard Candies Fairly

Marco Dall'Aglio and Raffaele Mosca
Dipartimento di Scienze
Università d'Annunzio
Viale Pindaro, 42
65127 — Pescara, Italy

Abstract

We consider the problem of allocating a finite number of indivisible items to two players with additive utilities. We design a procedure that looks for all the maximin allocations. The procedure makes repeated use of an extension of the Adjusted Winner, an effective procedure that deals with divisible items, to find new candidate solutions, and to suggest which items should be assigned to the players.

JEL classification: D61, D63

Keywords: Fair division, indivisible items, Adjusted Winner

1 Introduction

This paper presents a procedure for allocating a set of indivisible items between two players with subjective preferences over the items.

While most of the literature in fair division theory deals with one or more completely divisible goods (such as cakes or pieces of land), recent works by Brams with several coauthors drew attention on the problem of allocating several indivisible items. Brams, Edelman and Fishburn [4] point out how the most commonly accepted criteria for optimality may conflict with each other when players rank the items according to their preferences, so that

> achieving fairness requires some consensus on the ground rules and some delicacy in applying them.

In the same context of ordinal preferences, Brams and King [8] focus on the possible incompatibility between rank- and Borda-maxmin allocations on one side and envy-free ones on the other. Brams, Edelman and Fishburn [5], again, provide conditions for the existence of allocations which are optimal according to different criteria and study the relationship among those criteria for any number of players and items. An earlier work by Brams and Fishburn [6], focuses on the case of two players with the same ranking on the items.

When it comes to the design of specific procedures, however, it turns out that most of the proposals devise some technique to treat some, or all, of the contended items as divisible. This is the case, for instance, of the *Adjusted Winner* (AW) procedure, certainly the most popular and effective procedure so far conceived. In case no actual splitting is allowed, one may recur to such

surrogates as the use of side payments, as in the Knaster method, or that of randomization, where items are given according to a probability distribution, or of side payments to compensate the giving up of some item. As noted in [13], however, there are situations where these methods are impractical or impossible to implement.

If we focus on methods that deal exclusively with the allocation of indivisible items, with no side actions to mitigate the discontent of some players, we find, quite surprisingly, a more narrow choice. Classical methods for the 2-player case are described in [9] and, especially, [10]. The simplest method is that of *strict alternation* of the players' right to pick an item. This approach usually favors the player who picks first. To mitigate this advantage, Brams and Taylor propose a preliminary ranking of the items (a "query step") with the immediate assignment of the undisputed items, followed by a division of the remaining items (the "contested pile") via an alternation schemes that uses a more balanced sequence. The whole scheme is called *balanced alternation*. Another option is given by Lucas' *method of markers* which could be seen as a discrete variant of the sliding knife procedure.

A few recent additions complete the list. Herreiner and Puppe in [13] define a *descending demand* procedure where each player, in turn, declare their most preferred bundle (i.e. a collection of items) until a feasible arrangement is met that maximizes the bundle's rank of the least favored player. The findings in [6] suggest a procedure that Brams and Fishburn set out in the same work to single out an allocation which is Pareto-optimal, it ensures that the less well-off player does as well as possible, and, often, he/she does not envy the other player. In a similar fashion Brams and King [8] devise a simple procedure based on balanced alternation and sincere choices that yields Pareto-optimality and does not rule out envy-freeness[1]

All the above mentioned methods require the players' ability to rank items or bundles of them, and can be adapted to the simpler framework in which players are able to assess the subjective utility (or score) of each item and these evaluations are additive. Brams and Fishburn [6] show conditions that make preference relations compatible with additive utilities, and explain how to simplify their procedure in this situation. Anyway, we record the lack of a procedures specifically designed to work with additive utilities, in a manner similar to what the AW procedure does for the divisible case. Our aim indeed is to devise a specific procedure that makes repeated use of the original AW procedure as a guide to decide who gets the single items. The procedure mimics the branch-and-bound algorithms of Operations Research (OR), but keeps the procedural appeal of the original AW and it can be implemented as a simple set of instructions given to the players. The association of ideas from OR with fair division is not new: In [15] Kuhn defines a linear program that has the Knaster rule for the efficient allocation of items with side payments as its solution. Demko and Hill [12] define a maximin optimization problem. They show

[1] In the context of ordinal preferences allocations are divided into *envy-free*, *envy-possible* and *envy-ensuring* ones. The procedure returns an allocation belonging to the first two classes

that this problem is computationally intractable and provide a lower bound for optimal value. The second half of the paper deals with randomized solutions for the same problem and shows how these can be computed through linear programming and duality techniques.

We adopt the same framework, focusing on the case of two players. Each player assigns a non-negative value to each item. The evaluations are additive, but no normalization is required, so the total value of the items may differ for the two players.

This work does not deal with manipulability issues: is it advantageous for the players to reveal the items' true values? A discussion of the manipulability for the Adjusted Winner procedure appears in [9] and strategy proof procedures have recently been introduced in [7] for the divisible case.

A similar approach to the one presented here is being developed by Bezáková and Dani [3]. The purpose of the two works, however, differs. In [3] computationally efficient algorithms are given that approximate the optimal solution and are implemented by suitably programmed computer routines. Here we focus on exact solutions, with the main aim to extend the AW procedure to cover the case of indivisible items and keep its procedural nature.

Section 2 defines the problem. Section 3 takes another look at the AW algorithm and an extension is discussed to consider the situations where players own initial endowments. Incidentally, a more efficient version of the original procedure is considered for the case where a large number of items are at stake. Finally, section 4 illustrates the branch-and-bound algorithm that makes use of the AW procedure with initial endowments to find new candidate solutions, and to suggest which items should be forcedly assigned to the players.

2 The problem

We consider the following simple problem: two children (players), Alice and Bob, are given a set of m hard candies to be shared between themselves. Candies are indivisible and each of them is assigned to one of the children. Children value the sweets according to their own taste. An allocation is sought that is optimal according to some social welfare criterion.

More formally, let $M = \{1, \ldots, m\}$ be the set of disputed items and let $a_1, a_2, \ldots, a_m$ ($b_1, b_2, \ldots, b_m$ resp.) be the non-negative evaluations of the single items by Alice (Bob, resp.). An *integer allocation* for the m items is described by a vector $x = (x_1, \ldots, x_m) \in \{0, 1\}^m$. If $x_i = 1$ (resp. $x_i = 0$), then item i goes to Alice (Bob, resp.). The satisfaction (or score) of the two players is given by, respectively,

$$v_A(x) = \sum_{i \in M} a_i x_i \qquad \text{and} \qquad v_B(x) = \sum_{i \in M} b_i (1 - x_i) \tag{1}$$

There are many criteria that mediate between the conflicting interests of the players. We follow Brams and Fishburn [6], who

> recommend an alternative procedure that implements [their] fairness criteria when additive utilities are presumed. [...]the alternative procedure seeks a division that maximizes [$\min\{v_A, v_B\}$] over all divisions, subject to
>
> $$v_A((1,1,\ldots,1)) = v_B((0,0,\ldots,0)) \tag{2}$$

Therefore we look for an integer allocation that achieves

$$z^* = \max\{\min\{v_A(x), v_B(x)\} : x \in \{0,1\}^m\} \tag{IFD}$$

subject to (2). The same problem was considered earlier by Demko and Hill in [12], who noted its NP-hardness. In fact, assume that $a_i = b_i$ for every $i \in M$: then solving (IFD) gives an answer to the problem of finding a partition of a set of positive integers in two subsets of equal sum, which is NP-complete (see for instance [19]).

As pointed out by Brams, Edelman and Fishburn [4] in the context of ordinal preferences, a maximin allocation of indivisible items may generate envy between the players. Moreover the optimal partition may assign a different number of items to the players — thus being *unequal*. *Equitability*, i.e. the property that the scores of the two players coincide, is rarely obtained for a solution of (IFD). Moreover, the solution may not be unique. Quoting Brams and Fishburn again

> If two or more division maximize the min value, [the procedure] then finds an [allocation] within the maximin set that maximizes $\max\{v_A(x), v_B(x)\}$

This is referred to in the literature as the *equimax* (or Rawls, or Dubins-Spanier) allocation and has the property of (strong) Pareto-optimality: no other allocation weakly dominates it. Alternatively, we may choose a maximin allocation that minimizes $\max\{v_A(x), v_B(x)\}$. This time only weak Pareto-optimality is ensured (no other allocation strongly dominates it) but the resulting allocation would be closer to equitability. Neither one of the two restrictions, however, would ensure uniqueness in the solution. In what follows, we will distinguish between methods that are able to find one maximin solution, and those who can list them all.

3 Relaxing Integer Fair Division: The Adjusted Winner procedure

Suppose now that children are given muffins (with different flavors), instead of hard candies. Each muffin can be given in its entirety to one of the children — or it can be split in any proportion. We are now dealing with the allocation of m divisible items between two players.

It is further assumed that all items $i \in M$ are homogeneous. Thus player 1 can receive a part $x_i \in [0,1]$ of item i, while player 2 gets the rest. The two players will benefit, respectively, by $x_i a_i$ and $(1-x_i)b_i$ from the splitting. The overall satisfaction of each player is still given by (1). We now look for an allocation $x = (x_1, x_2, \ldots, x_m) \in [0,1]^m$ that achieves

$$z^+ = \max\{\min\{v_1(x), v_2(x)\} : x \in [0,1]^m\} \qquad \text{(DFD)}$$

with v_A and v_B satisfying (2). As noted in [12], (DFD) can be solved through linear programming, and in the OR jargon, this is the *linear relaxation* of (IFD).

Here we are going to show that a solution for (DFD) is readily provided by a popular and effective step-by-step procedure in fair division, known as *Adjusted Winner*.

3.1 The Adjusted Winner algorithm

The Adjusted Winner (AW) algorithm was introduced by Brams and Taylor in [9] (with many applications analyzed in [10]). Their aim was to provide a step-by-step procedure returning a partition that is equitable, Pareto optimal and envy-free (in the sense that none of the player feels that the other player has received more than him/herself). A brief sketch of the algorithm follows — for a more detailed account we refer to [9] and [10]. There are two phases:

the "winning" phase. each player temporarily receives the items that he/she values more than the other player does — ties being temporarily assigned to any of the players. The total score of each player, v_A and v_B respectively, is computed.

the "adjusting" phase. Items are transferred, one at a time from the "richer" player to the "poorer" one, starting with the items with ratio a_i/b_i closer to 1. To reach equitability one item may be split into two parts.

As an exemplification, suppose $v_A \geq v_B$. Then Alice begins transferring items to Bob, one at a time, starting with the item with ratio a_i/b_i closer to 1 (and greater than or equal to 1). The handover continues until perfect equitability is achieved, or the roles of the "richer" and "poorer" player are reversed. In the last case, suppose that after the handover of, say, item r we have $v_A < v_B$. Item r is then split, with Alice getting a fraction given by

$$x_r = \frac{b_r + v_B^{-r} - v_A^{-r}}{a_r + b_r}$$

where v_A^{-r} and v_B^{-r} are the scores obtained by the two players so far in the process *without* considering item r. Bob gets the remaining fraction. The item is split according to the same proportions also when Bob is favored

in the "winning" phase and the handover occurs in the opposite direction. Both players walk out of the procedure with a common score of

$$z^+ = v_A = v_B = \frac{v_B^{-r} a_r + v_A^{-r} b_r + a_r b_r}{a_r + b_r}$$

We next show that AW provides exactly what we are looking for in the maximin problem with divisible items.

Proposition 3.1. *The AW algorithm solves* (DFD). *Therefore, the AW solution is also maximin.*

Some preliminary results are required. First of all consider the *allocation range*.

$$\mathcal{D} = \{(v_A(x), v_B(x)) : x \in [0,1]^m\}$$

Lemma 3.2. *$\mathcal{D}$ is a convex and compact set in $\mathbb{R}^2$.*

Proof. Pick $x, y \in [0,1]^m$ and $\gamma \in [0,1]$. Then

$$v_A(\gamma x + (1-\gamma)y) = \gamma v_A(x) + (1-\gamma) v_A(y)$$

and the same holds for v_B, so $\mathcal{D}$ is convex. Compactness is a consequence of the compactness of $[0,1]^m$ and the continuity of v_A and v_B. More in detail, $\mathcal{D} \subset [0, \sum_{i \in M} a_i] \times [0, \sum_{i \in M} b_i]$, so $\mathcal{D}$ is bounded. Consider now a sequence $\{x_n\}$ in $[0,1]^m$ for which $(v_1(x_n), v_2(x_n))$ converges. Since $[0,1]^m$ is compact, there exists a subsequence $\{x_{n'}\}$ converging to some $x^* \in [0,1]^m$. Since v_A and v_B are continuous, we have

$$(v_A(x_{n'}), v_B(x_{n'})) \to (v_A(x^*), v_B(x^*)) \in \mathcal{D}$$

and $\mathcal{D}$ is closed. □

Next we characterize the maximin solutions.

Lemma 3.3. *A maximin solution always exists. An allocation is maximin if and only if it is Pareto optimal and equitable.*

Proof. We prove the "only if" part of the prove, since this is what is actually needed for Proposition 3.1.[2]

We consider the set $\mathcal{D}$ of all the allocations' values. An allocation x is Pareto if there is no other point of $\mathcal{D}$ in the upper quadrant pointed on $(v_A(x), v_B(x))$ (with the exception of x itself). The allocation is equitable if $(v_A(x), v_B(x))$ lies on the bisector of the positive quadrant.

Let $\mathcal{Q}$ be the family of upper quadrants pointed on the equitable allocations. A maximin solution is obtained by considering the supremum of the quadrants in $\mathcal{Q}$ that intersects $\mathcal{D}$. Since $\mathcal{D}$ is compact, the supremum is attained, and a maximin solution x^* exists.

[2]For the whole proof we refer to the longer version of the paper.

Suppose that x is Pareto and equitable. Equitability implies that the upper quadrant pointed on $(v_A(x), v_B(x))$ is in $\mathcal{Q}$. Pareto optimality implies that no other part of $\mathcal{D}$ lies on the same quadrant. Therefore x is maximin. □

Proof of Proposition 3.1. It is a straightforward consequence of Lemma 3.3 and the following

Theorem 3.4. *(Brams and Taylor, [9], Th,4.1) AW produces an allocation of the goods that is Pareto-optimal and equitable.*

□

An alternative proof of Theorem 3.4 which links AW to a cake cutting scheme is provided by Jones [14].

3.2 The maximin problem with initial endowments

The AW procedure is flexible enough to cover the situation where the two players own initial endowments. This variation is interesting in its own rights. An optimal allocation is sought when the utility of each player is the sum of the initial endowment and the values of the items (or fractions thereof) received. Our interest in this problem, however, is mainly instrumental. In order to implement a branch-and-bound method for the case of indivisible items we need to solve several instances of the corresponding problem with divisible items in which certain items are forcedly assigned to the players. These items represent their initial wealth. Let $\alpha \geq 0$ (resp. $\beta \geq 0$) the initial endowment of Alice (Bob, resp.) that add up to the players' utilities.

The problem of interest is now:

$$z^+ = \max\{\min\{\alpha + v_A(x), \beta + v_B(x)\} : x \in [0,1]^m\} \qquad \text{(DFD-ie)}$$

This time we do not impose a normalization condition such as (2), but rather assume all terms a_i, b_i to be strictly positive. We could in fact set up a preliminary step that deals with null values. If $a_i > 0$ and $b_i = 0$, then item i can be assigned to Alice with no harm for Bob, and increase her initial endowment. Similarly, Bob could take all items with null value to Alice (and items with no value for both could be thrown away). It may be worthwhile noticing that, by removing (2), we may lose envy-freeness. Consider for instance the case with no initial endowments and only one muffin M, with $v_A(M) = 10$ and $v_B(M) = 5$. Bob will now get 2/3 of the muffin, leaving Alice envious.

Once again the maximin solution coincides with the Pareto and equitable solution, but only when the value of the assignable items according to the poorer player is larger than or equal to the difference between the initial endowments. We propose the following:

The Adjusted Winner procedure with initial endowments (AW-ie)

Case 1 If $\sum_{i\in M} b_i \leq \alpha - \beta$ assign all the items to Bob. The maximin value will be $z^+ = \beta + \sum_{i\in M} b_i$

Case 2 If $\sum_{i\in M} a_i \leq \beta - \alpha$ then assign all the items to Alice and $z^+ = \alpha + \sum_{i\in M} a_i$

Case 3 If $-\sum_{i\in M} a_i < \alpha - \beta < \sum_{i\in M} b_i$ then start an AW procedure with the only difference that the initial endowment is taken into account to reach an equitable allocation. So, after a winning phase identical to the AW procedure, the total scores, inclusive of the initial endowment are computed. In the adjusting phase, items are transferred from the "richer" player to the "poorer" one, ordering the items in terms of the preference ratios a_i/b_i. The process stops when perfect equitability between the scores with the endowments is reached or the roles of the two players are reversed. In this case, the last transferred item, say r, is split and Alice gets a fraction given by

$$x_r = \frac{\beta - \alpha + b_r + v_B^{-r} - v_A^{-r}}{a_r + b_r}$$

while Bob gets the rest. Both players walk out with a common score of

$$z^+ = \alpha + v_A = \beta + v_B = \frac{(\beta + v_B^{-r})a_r + (\alpha + v_A^{-r})b_r + a_r b_r}{a_r + b_r}$$

Proposition 3.5. *The AW-ie procedure returns the solution for (DFD-ie).*

Proof. In this case the utility of the two players is given, respectively, by $v'_A(x) = \alpha + v_A(x)$ and $v'_B(x) = \beta + v_B(x)$, so the new allocation range $\mathcal{D}'$ is simply the allocation range $\mathcal{D}$ translated by (α, β). The three cases listed above correspond to different positions of $\mathcal{D}'$ with respect to the bisector of the first quadrant

Case 1: $\mathcal{D}'$ lies below the bisector. So $u \geq w$ for all $(u, w) \in \mathcal{D}'$ and the allocation that assigns all goods to Bob is maximin.

Case 2: $\mathcal{D}'$ lies above the bisector. Therefore $u \leq w$ for all $(u, w) \in \mathcal{D}'$ and all goods are given to Alice in order to have the highest maximin value.

Case 3: $\mathcal{D}'$ crosses the bisector Lemma 3.3 is still valid. Thus, the procedure looks for an allocation that is Pareto-optimal and equitable.

□

If case 3 holds, a more efficient version of AW can be implemented just as it was done for the original AW. This time $v_A^i(\lambda)$ and $v_B^i(\lambda)$, $i = 1, 2$ will include the initial endowments of the respective players.

4 A branch and bound algorithm

When solving the maximin allocation problem (IFD) there is a finite number of possible candidates to choose from. In principle the solution can be obtained in finite time by computing the value of each allocation for the two players. This process can be considerably speeded up if we consider a branch-and-bound technique that splits the original problem into smaller subproblems and uses upper bounds to avoid exploring certain parts of the set of feasible integer solutions. This approach makes repeated use of the Adjusted Winner procedure with initial endowment and keeps the procedural character of the latter.

In what follows, we will consider a series of constrained subproblems in which some of the items have already been assigned to the players. Let $A, B \subset M$, with $A \cap B = \emptyset$. Let $S(A, B)$ be the constrained problem in which the items in A (B, resp.) are assigned to Alice (Bob, resp.), i.e., $x_i = 1$ for each $i \in A$ ($x_i = 0$ for each $i \in B$). $S(\emptyset, \emptyset)$ denotes the original (unconstrained) problem.

For a given couple of disjoint index sets, A, B in M, let $\bar{x}(A, B)$ denote a feasible allocation for the constrained problem and let $\bar{z}(A, B)$ denote the corresponding value. Moreover, let $x^*(A, B)$ and $z^*(A, B)$ denote the solution and the value of $S(A, B)$. Finally let $x^+(A, B)$ and $z^+(A, B)$ be, respectively, the solution and value for the linear relaxation of $S(A, B)$, i.e. for the case where splitting of the contended items is allowed. Clearly, the following holds for each couple of A and B:

$$\bar{z}(A, B) \leq z^*(A, B) \leq z^+(A, B) \tag{3}$$

The results in Section 3 can be used to compute $x^+(A, B)$ and $z^+(A, B)$. In particular we set $\alpha = \sum_{i \in A} v_1(x_i)$ and $\beta = \sum_{i \in B} v_2(x_i)$, and divide the remaining $M' = M \setminus (A \cup B)$ items according to the AW-ie procedure. Since $x^+(A, B)$ contains at most one fractional component, $\bar{x}(A, B)$ may be obtained by approximating the fractional coordinate to the nearest integer, 0 or 1.

4.1 A variable elimination test

The branch-and-bound procedure defines a series of subproblems in which an increasing numbers are forcedly assigned to one player or the other. Since the procedure becomes simpler as the number of pre-assigned items increases,and following [18], p.452, we consider a variable elimination test that, for any given subproblem, checks whether additional items can be assigned priori to any further analysis.

Let $A, B \subset M$ be a couple of disjoint sets of items and take $i \in M'$.

Proposition 4.1. *(a) If*

$$z^+(A \cup \{i\}, B) < \bar{z}(A, B) \tag{4}$$

then $x^*(A, B \cup \{i\})$ solves $S(A, B)$, while $x^*(A \cup \{i\}, B)$ does not.

(*b*) *If*

$$z^+(A, B \cup \{i\}) < \bar{z}(A, B) \tag{5}$$

then $x^*(A \cup \{i\}, B)$ *solves* $S(A, B)$*, while* $x^*(A, B \cup \{i\})$ *does not.*

Proof. By assumption and (3) we have

$$z^*(A \cup \{i\}, B) \leq z^+(A \cup \{i\}, B) < \bar{z}(A, B) \leq z^*(A, B)$$

So $x^*(A \cup \{i\}, B)$ cannot be a solution for $S(A, B)$. If this is the case, then $x^*(A, B \cup \{i\})$ must be a solution for the same problem. Part (b) is established symmetrically. □

The result simply states that whenever condition (4) ((5), resp.) occurs, then $S(A, B)$ can be replaced by $S(A, B \cup \{i\})$ ($S(A \cup \{i\}, B)$, resp.). When the two sides of (4), or (5), attain equality, there is a partial extension of the previous result:

Proposition 4.2. (*a*) *If* $z^+(A \cup \{i\}, B) \leq \bar{z}(A, B)$*, then either* $x^*(A, B \cup \{i\})$ *or* $\bar{x}(A, B)$ *solve* $S(A, B)$*.*

(*b*) *If* $z^+(A, B \cup \{i\}) \leq \bar{z}(A, B)$*, then either* $x^*(A \cup \{i\}, B)$ *or* $\bar{x}(A, B)$ *solve* $S(A, B)$*.*

(*c*) *If* $z^+(A \cup \{i\}, B) \leq \bar{z}(A, B)$ *and* $z^+(A, B \cup \{i\}) \leq \bar{z}(A, B)$*, then* $\bar{x}(A, B)$ *solves* $S(A, B)$*.*

Proof. (a) By assumption

$$z^+(A \cup \{i\}, B) \leq \bar{z}(A, B) \leq z^*(A, B)$$

Assume now that $x^*(A, B \cup \{i\})$ does not solve $S(A, B)$. Then $x^*(A \cup \{i\}, B)$ will work instead, and thus

$$z^*(A, B) \leq z^*(A \cup \{i\}, B) \leq z^+(A \cup \{i\}, B)$$

Comparing the two inequalities, we conclude that $\bar{z}(A, B) = z^*(A, B)$ and $\bar{x}(A, B)$ solves $S(A, B)$. Part (b) is proved with a symmetrical argument.

(c) By definition

$$\bar{z}(A, B) \leq z^*(A, B) \leq z^+(A, B) \leq \max\{z^+(A \cup \{i\}, B), z^+(A, B \cup \{i\})\}$$

while the hypotheses reads

$$\max\{z^+(A \cup \{i\}, B), z^+(A, B \cup \{i\})\} \leq \bar{z}(A, B)$$

Thus $\bar{x}(A, B)$ solves $S(A, B)$. □

The use of Proposition 4.2 is more subtle: when situation (a) occurs, than we replace $S(A, B)$ with $S(A, B \cup \{i\})$ and continue with the sub-partitioning to obtain a solution $\tilde{x}$. This solution is then compared with $\bar{x}(A, B)$. The one with the higher value is the solution for (IFD).

At first sight, Proposition 4.2 is more powerful than Proposition 4.1 since it binds more items to the players, thus making the problem simpler. Using this result, however, may result in the loss of some solutions. Part (a) of the statement does not prevent $x^*(A \cup \{i\})$ from being a possible solution for $S(A, B)$ (and a symmetrical conclusion holds for part (b)). So, if the goal is to capture all the solutions for (IFD), Proposition 4.1 is the one to choose[3].

The problem remaining after the elimination test has been carried out is called the *reduced problem*. Note that the discriminating value λ is the same for the reduced problem as well as for the original problem.

4.2 The algorithm

All the elements are set to formulate a branch-and-bound algorithm for the maximin problem with indivisible items (IFD). The algorithm follows the general scheme for branch-and-bound, where the original problem $S(\emptyset, \emptyset)$ is recursively split into a series of constrained problems with some of the items assigned in advance to one player or the other. As usual for this kind of algorithms, it is convenient to represent the splitting process with a tree graph. When a subproblem cannot yield any more candidates for the solution of the original problem, the branch corresponding to that subproblem is cut (or pruned) and no other branch generates from that node of the tree.

The general framework is adapted to the peculiar features of the problem in question. For instance, the linear relaxation of each subproblem has a twofold purpose: on one hand it gives an upper bound for the value of the integer solution, but when the solution for the linear relaxation is not integer, it also suggests how to operate the splitting, by assigning the item corresponding to the unique fractional component to one player or the other.

In building the tree, several integer solutions are met and the best of them (in terms of objective function) are recorded. Here we are interested in finding all the solutions to (IFD). Therefore $\bar{X}$ will denote the set of best solutions met so far, while $\bar{z}$ is their common value.

Each subproblem $S(A, B)$ may have three different labels attached to it: "new", "open" or "close": a subproblem is new when its linear relaxation has not been computed yet; once the computation occurs, the problem is open or close depending on whether the solution for the relaxation is integer or not. Furthermore, a subproblem may also be closed when its upper bound is smaller than the best current admissible solution. Open problems are split according to the above mentioned rule. The algorithm ends when all the subproblem are closed.

The algorithm runs as follows:

[3]The use of Proposition 4.2 is explaines with more detail in the longer version of the paper.

Initialization. Set $\bar{X} = \emptyset$ and $\bar{z} = -\infty$. Label $S(\emptyset, \emptyset)$ as new.

The generic cycle is made of the following steps

Compute bounds. For any new subproblem $S(A, B)$ perform the variable elimination test derived from Proposition 4.1 and denote with $S(A', B')$ the resulting subproblem with (possibly) more items preassigned to the players.

- Compute $x^+(A', B')$ and $z^+(A', B')$ using the AW-ie algorithm.
- Examine $x^+(A', B')$.
  - If $x^+(A', B')$ is integer then set $\bar{x}(A', B') = x^+(A', B')$ and $\bar{z}(A', B') = z^+(A', B')$. Label $S(A', B')$ as close.
  - If $x^+(A', B')$ has a fractional component then set $\bar{x}(A', B') = \text{rnd}(x^+(A', B'))$ with corresponding value $\bar{z}(A', B')$. Label $S(A', B')$ as close.
- Update the optimal set
  - If $\bar{z}(A', B') > \bar{z}$ then set $\bar{z} = \bar{z}(A', B')$ and $\bar{X} = \{\bar{x}(A', B')\}$.
  - If $\bar{z}(A', B') = \bar{z}$ and $\bar{x}(A', B') \notin \bar{X}$ then append this solution to $\bar{X}$.

List and close List the open subproblems. Close all the $S(A, B)$ such that

$$z^+(A, B) < \bar{z} \,. \tag{6}$$

If there is no open subproblem left, then exit the algorithm and return $\bar{X}$ as the optimal solution set with value $\bar{z}$.

Choose and split Choose the open problem $S(A, B)$ with higher upper bound $z^+(A, B)$. The relaxed solution $x^+(A, B)$ has one fractional component $i \in M \setminus (A \cup B)$. Replace $S(A, B)$ (labelled close) with two subproblems $S(A \cup \{i\}, B)$ and $S(A, B \cup \{i\})$, labelling them as new. Continue with the next cycle.

Some of the rules in the algorithm may be changed. For instance another criterion may be selected to pick an open problem. A naïve motivation for the chosen rule is that the higher the bound, the more likely is the subproblem to deliver an optimal solution. Also, when $x^+(A', B')$ has a fractional component, we assign the fractional good to the player who holds more than 50% of it to get an integer admissible solution. Alternatively, we may check both options: assigning the split good to Alice and Bob, and choosing the one yielding a higher value for z. Although the latter seems a more efficient option, we prefer the simplicity of the former rule.

As noted previously, we may use a variable elimination test based on Proposition 4.2. The algorithm will be quicker, but some solutions may be left off of the solution set $\bar{X}$.

Since the IFD problem is NP-hard, in the worst case the algorithm may execute an exponential number of iterations to determine an optimal solution

(unless $P = NP$). To this end, let us consider the following instance of IFD: there are $m = 2n+1$ items and both players evaluate 2 each item. In this case: the value of an optimal solution is $2n$, and corresponds to any solution which assigns n items to one player, and $n+1$ items to the other player; the upper bound is $2n+1$ (n variables are equal to 1, one variable is equal to 0,5, and n variables are equal to 0); a node can be closed only if either at least $n+1$ variables are fixed equal to 1, or at least $n+1$ variables are fixed equal to 0. Then the enumeration tree of the branch and bound, independently to possible strategies (of searching), is totally explored till level n, i.e., the number of nodes which are examined is $2^{(n+1)}$.

At the same time, the efficiency of a branch and bound technique is directly linked to the quality of (i) the method to compute bounds for each subproblem, (ii) the method to possibly split each subproblem. In our case, (i) is given by the linear programming relaxation of each subproblem, and (ii) is given by the generation of two new subproblems obtained by splitting a binary variable. The proposed algorithm adopts methods which follow the ones used in the most popular (and empirically considered efficient) branch and bound algorithms for the solution of optimization 0-1 Knapsack, which is a problem very close to IFD.

When a small number of items is at stake, the algorithm can be run by humans — in the spirit of the original AW procedure. Finding methods that keep this feature for larger bundles is the subject of our current research.

5 Acknowledgements

This is an abridged version of the paper "How to allocate hard candies fairly" which is currently being submitted for publication. The authors wish to thank two anonymous referees for the careful revision of the work and the constructive remarks. We would also like to thank Steve Brams, Erio Castagnoli and Ted Hill for useful suggestions during the completion of the work. The authors are responsible for all the remaining errors.

References

[1] Akin, E., 1995. Vilfredo Pareto cuts the cake, Journal of Mathematical Economics 24, 23–44.

[2] Bertsimas, D., and J.N. Tsitsiklis, 1997, Introduction to linear optimization, Athena Scientific, Belmon, Massachusetts, U.S.A.

[3] Bezáková, I., Dani, V., 2005. Allocating indivisible goods, ACM SIGecom Exchanges5, 11–18.

[4] Brams, S.J., Edelman, P.H., Fishburn, P.C., 2001. Paradoxes of Fair Division, Journal of Philosophy 98, 300–314.

[5] Brams, S.J., Edelman, P.H., Fishburn, P.C., 2003. Fair division of indivisble items, Theory and Decision 55, 147–180.

[6] Brams, S.J., Fishburn, P.C., 2000. Fair division of indivisble items between two people with identical preferences: Envy-freeness, Pareto-optimality, and equity, Social Choice and Welfare 17, 247–267.

[7] Brams, S.J., Jones, M.A., Klamler, C., 2006. Better ways to cut a cake, Notices of the American Mathematical Society 53, 1314–1321.

[8] Brams, S.J., King, D.L., 2005. Efficient Fair Division: Help the Worst Off or Avoid Envy?, Rationalitiy and Society 17, 387–421.

[9] Brams, S.J., Taylor, A.D., 1996. Fair Division: from Cake-cutting to Dispute Resolution, Cambridge University Press.

[10] Brams, S.J., Taylor, A.D., 1999. The Win-win Solution, Guaranteeing Fair Shares to Everybody, W.W.Norton.

[11] Cormen, T.H., Leiserson, C.H., Rivest, R.L., Stein, C., 2001. Introduction to Algorithms, The MIT Press.

[12] Demko,S., Hill, T.P., 1988. Equitable distribution of indivisible objects, Mathematical Social Sciences 16, 145–158.

[13] Herreiner, D., Puppe, C., 2002. A simple procedure for finding equitable allocations of indivisible goods, Social Choice and Welfare 19, 415–430.

[14] Jones, M.A., 2002. Equitable, envy-free, and efficient cake cutting for two people and its applications to divisible goods, Mathematics Magazine 75, 275–283.

[15] Kuhn, H.W., 1967, On games of fair division, In: Shubik M.(Ed.), Essays in Mathematical Economics in Honor of Oskar Morgenstern. Princeton University Press.

[16] Legut, J., Wilczyński, M., 1988. Optimal partitioning of a measurable space, Proceedings of the American Mathematical Society 104, 262–264.

[17] Papadimitriou, C.H., Steiglitz, K., 1982. Combinatorial Optimization: Algorithms and Complexity, Prentice & Hall.

[18] Wolsey, L.A., Nemhauser, G.L., 1988. Integer and Combinatorial Optimization, Wiley-Interscience.

[19] Vazirani, V.V., 2001. Approximation Algorithms, Springer-Verlag.

Social Choice and the Logic of Simple Games

Tijmen R. Daniëls

Abstract

From the perspective of the minimal majority logic proposed by Pauly [12], we investigate the relation between axiomatic social choice theory, the logic of simple games, and neighbourhood semantics. We discuss the importance of the Rudin-Keisler ordering in this context and provide a simple characterisation of the monotonic modal fragment that corresponds to the logic of simple games based on this ordering. Finally we discuss its relevance for axiomatic social choice theory.

1 Introduction

Social choice theory is concerned with questions on how a group of agents can decide as a collective in a way that reflects the individual opinions of those involved. The rich history of the subject can be traced back more than two centuries, to eighteenth century thinkers such as Jeremy Bentham, Jean-Charles de Borda, and especially the Marquis de Condorcet. For many years the work done by these thinkers laid dormant. Then social choice suddenly picked up steam in the 1950's, when the economist Kenneth Arrow used observations originally made by Condorcet to prove a striking result, *viz.*, that it is impossible to aggregate rational preference relations into a collective (or social) rational preference relation by a mathematical procedure that satisfies certain natural axioms, or 'democratic' desiderata [1]. Many similar results followed in its wake.

Some recent work on social choice has revolved around **judgement aggregation** (a non-exhaustive list includes [3], [8], [10], [13]). This work is concerned with the question of aggregating a collection of sentences in of a formal logical language, in a logically consistent way, and by a method reflecting the individual views of a group of agents as much as possible. In some sense the story of judgement aggregation appears as a case of history repeating. Judgement aggregation can superficially be regarded as a generalisation of preference aggregation—it is by now well established that virtually all results on preference aggregation have their counterparts in this newer context. And indeed, the interest in judgement aggregation was spawned initially by the discovery of an Arrow style impossibility result (List and Pettit, [8]).

In our view, however, there are at least two merits of judgement aggregation over preference aggregation that warrant the renewed interest. First, by investigating the boundaries of collective reasoning from a purely logical stance, judgement aggregation elevates the theory of social choice to a higher level of abstraction as well as to a broader, and perhaps more natural, conceptualisation of the "rationality of the collective" than is provided by the focus on preference

relations stemming from economics. Second, judgement aggregation very explicitly brings out the connection between logic and social choice theory. A link between social choice and logic has always been present—on occasions Kenneth Arrow has recounted that his interest in applying axiomatic methods to social choice had sprung from exposure to the mathematics of Gödel and Tarski. But the new context has inspired logicians to investigate higher-order questions about social choice using formal methods. One promising way to go about is to define a formal language which can formalise certain behavioural properties of aggregation procedures. Recently, Pauly [12] has provided a modal-flavoured logic of collective decision making that does just that.

This paper is in this more recent logical tradition. It is not so much concerned with impossibility results *per se*, but rather with placing social choice in the context of methods familiar to logicians. We will be working with a formal language of collective decision making in the tradition of Pauly [12], defined in section 2. Instead of studying the language in isolation, we will make use of the artillery provided by monotonic modal logic and simple games. The importance of the latter to understanding social choice has been stressed by e.g. Monjardet [9]. Section 3 discusses such simple games in some depth; we present a generalisation of Monjardet's results to the logic of collective decision making and look at the Rudin-Keisler ordering on simple games. In section 4 we relate this perspective to monotonic modal logic. We work towards a simple characterisation that shows how the logic of collective decision procedures fits into the larger modal picture. We conclude with some implications for the axiomatic method: application of standard methods gives insight into what classes of social aggregation procedures can be defined in simple modal languages.

1.1 Preliminaries

We will define a basic language $\mathscr{L}_c$ that is just classical propositional logic. Thus, formulae in the language $\mathscr{L}_c$ are constructed from a set of sentence letters $q_1, q_2, \ldots$, and the logical connectives $\wedge, \neg$. Throughout the text we follow the standard conventions for bracketing and use the abbrevations $\rightarrow$, $\leftrightarrow$, $\vee$. By $\models$ we denote the standard (semantic) entailment relationship; $\models \varphi$ means φ is a tautology; $\varphi \models \psi$ means ψ follows from φ.

For the purpose of this paper we fix a finite number of sentence letters $\mathsf{Q} := \{q_1, \ldots, q_h\}$. N is the set of agents—whenever we assume N finite we will state this explicitly. A **choice function** is a function $\pi : N \rightarrow \mathscr{P}(\mathsf{Q})$; intuitively $\pi(i)$ provides the information on the choices of agent i. Π is the set of all such functions. Given $Q \subseteq \mathsf{Q}$, φ_Q is the formula:

$$\varphi_Q := \bigwedge_{q_i \in Q} q_i \ \wedge \bigwedge_{q_i \in (\mathsf{Q}-Q)} \neg q_i$$

If $\varphi_{\pi(i)} \models \psi$ then we say that "agent i accepts ψ". The set of all agents that accept $q_j \in \mathsf{Q}$, that is $\{i \in N \mid q_j \in \pi(i)\}$, is denoted by $[\![q_j]\!]_\pi$. More generally, for $\psi \in \mathscr{L}_c$, $[\![\psi]\!]_\pi := \{i \in N \mid \varphi_{\pi(i)} \models \psi\}$.

A **social aggregation function (SAF)** is a (possibly partial) function $F : \Pi \to \mathscr{P}(\mathscr{L}_c)$; $F(\pi)$ denotes the socially accepted sentences of $\mathscr{L}_c$ given π. The following terminology is standard:

Definition 1 *Let $\pi, \pi' \in \Pi$, $\varphi, \psi \in \mathscr{L}_c$ be arbitrary. A SAF is said to satisfy:*
universal domain (UD) *iff the domain of F is Π;*
monotonicity (M) *iff whenever $[\![\varphi]\!]_\pi \subseteq [\![\varphi]\!]_{\pi'}$ then $\varphi \in F(\pi) \implies \varphi \in F(\pi')$;*
neutrality (N) *iff whenever $[\![\varphi]\!]_\pi = [\![\psi]\!]_{\pi'}$ then $\varphi \in F(\pi) \iff \psi \in F(\pi')$.*

2 Semantics Based on SAFs

Our point of departure will be the following language whose semantic interpretation will be defined in terms of SAFs. This language $\mathscr{L}_\Box$ is grammatically generated by:

$$\psi ::= \Box\alpha \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \bot \qquad \text{with each } \alpha \in \mathscr{L}_c.$$

The interpretation of $\Box\psi$ is that "ψ is collectively accepted". The proposed interpretation of the $\Box$ operator leads us to consider the following natural semantics for the language $\mathscr{L}_\Box$: we interpret the formulae using SAFs and choice functions. The $\Box$ serves to shield the logic of group decisions from the (possibly logically inconsistent) outcome of the aggregation procedure. This gives the language distinct modal flavour, however there are no (iterated) modalities and also no boxless formulae. The origin of these ideas is Pauly [12], but readers familiar with that paper should be warned that the present semantics differ in details: Pauly's models assign truth values directly to the formulae of $\mathscr{L}_\Box$.

Definition 2 *Let F be a SAF, and π a choice function in the domain of F. The pair (F, π) is called a* ***model****. Let $\psi, \psi_1, \psi_2 \in \mathscr{L}_\Box$ and $\Psi \subseteq \mathscr{L}_\Box$. We write:*

	$(F,\pi) \Vdash \Box\varphi$	*iff $\varphi \in \mathscr{L}_c$ and $\varphi \in F(\pi)$;*
	$(F,\pi) \Vdash \psi_1 \wedge \psi_2$	*iff $(F,\pi) \Vdash \psi_1$ and $(F,\pi) \Vdash \psi_2$;*
	$(F,\pi) \Vdash \neg\psi$	*iff $(F,\pi) \nVdash \psi$;*
	$(F,\pi) \Vdash \bot$	*never,*
and:	$F \Vdash \psi$	*iff for all $\pi \in dom(F), (F,\pi) \Vdash \psi$,*
and finally:	$F \Vdash \Psi$	*iff for all $\psi \in \Psi, F \Vdash \psi$.*

Now consider:

$$RE := \{\Box\varphi \leftrightarrow \Box\psi \mid \models \varphi \leftrightarrow \psi\} \tag{RE}$$
$$RM := \{\Box\varphi \to \Box\psi \mid \models \varphi \to \psi\} \tag{RM}$$

It may be verified that the following holds (see also Pauly [12]):

Lemma 3 *If F satisfies* **N**, *then $F \Vdash RE$. If F satisfies* **N** *and* **M**, *then $F \Vdash RE \cup RM$.*

In the balance of this paper, we will be concerned with the logic of SAFs that are monotonic and neutral and satisfy universal domain. Models based on such SAFs will be called **simple models**.

3 Simple Games

Perhaps the most familiar and natural aggregation procedure is simple majority voting. Simple games provide a generalised interpretation of the notion of a "majority". Certainly, if some subset A of the collective of agents, N, constitutes a majority of N, then any other subset B of N that properly contains A will also be a majority. This is the basic intuition underlying simple games, formulated by Von Neumann and Morgenstern [11], and formalised as follows. Let $W \subseteq \mathscr{P}(N)$ be the collection of subsets of N that we think of as the majorities of N (or, in game theoretic parlance, the **winning coalitions** of N). Then W is closed under supersets:

$$\text{if } A \in W \text{ and } A \subseteq B, \text{ then } B \in W. \tag{M1}$$

Hence by a **simple game** we mean a pair (N, W), where N is a nonempty set of agents and $W \subseteq \mathscr{P}(N)$ satisfies condition (M1). A simple game is **finite** if N is a finite set. A simple game is called **proper** if it satisfies:

$$A \in W \text{ implies } N - A \notin W. \tag{M2a}$$

A simple game is called **strong** if it satisfies:

$$A \notin W \text{ implies } N - A \in W. \tag{M2b}$$

If W satisfies (M2a), then A is a majority of N only if its complement isn't; that is to say, all majorities are **strict**. On the other hand (M2b) expresses that A is a majority whenever its complement isn't. In a historically important paper by G. Th. Guilbaud [5], the proper strong simple games were called **families of majorities**, and we will stick to this terminology below.[1]

A player $i \in N$ is called a **dummy player** of (N, W) if:

$$\text{for all } X \in \mathscr{P}(N),\ X \in W \iff X \cup \{i\}. \in W$$

Generalising this notion to sets, a set $A \subseteq N$ is called a **set of dummy players** if:

$$\text{for all } X \in \mathscr{P}(N), \text{ and any } B \subseteq A,\ X \in W \iff X \cup B \in W.$$

Given $\Omega = (N, W)$, denote the set of its dummy players by $\mathscr{D}(\Omega)$.

[1] In fact, the simple games envisioned by Von Neumann and Morgernstern were both proper and strong. They investigated various properties of such games, including issues of computational complexity.

3.1 Passing from Simple Games to Social Choice and Vice-Versa

In this subsection we narrow down the relation between simple games and monotonic, neutral and universal domain SAFs to a 1-1 correspondence. These results expand on observations made by Monjardet [9] on preference aggregation. Let $\Omega = (N, W)$ be a simple game. Define:

$$F_\Omega(\pi) := \{\psi \in \mathscr{L}_c \mid \exists A \in W \; \forall i \in A \; \varphi_{\pi(i)} \models \psi\},$$

In words, $\psi \in F(\pi)$ if there is some winning coalition A of Ω such that every agent $i \in A$ accepts ψ. Clearly F_Ω is a SAF that satisfies M, N, and UD. Some properties of simple games pass at once to the resulting aggregation function.

Lemma 4 *Let $\psi \in \mathscr{L}_c$ and $\Omega = (N, W)$ a simple game.*
(a). If Ω is proper, then $F_\Omega \models \Box\psi \to \neg\Box\neg\psi$;
(b). If Ω is strong, then $F_\Omega \models \neg\Box\neg\psi \to \Box\psi$.

Proof. Let π be an arbitrary choice function. (a). Let $\psi \in F_\Omega(\pi)$. Then there is $A \in W$ such that every agent $i \in A$ accepts ψ. So $A \subseteq [\![\psi]\!]_\pi$. By (M1), $[\![\psi]\!]_\pi \in W$. As $\varphi_Q \models \psi \iff \varphi_Q \not\models \neg\psi$, we have $N - [\![\psi]\!]_\pi = [\![\neg\psi]\!]_\pi$. By (M2a), $N - [\![\psi]\!]_\pi \notin W$. Suppose towards a contradiction that $\neg\psi \in F_\Omega(\pi)$. Then there is $B \in W$ such that every agent $i \in B$ accepts ψ. Clearly $B \subseteq [\![\neg\psi]\!]_\pi$, so by (M1) $[\![\neg\psi]\!]_\pi = N - [\![\psi]\!]_\pi \in W$, a contradiction. Hence $(F, \pi) \Vdash \neg\Box\neg\psi$. (b). Suppose $\neg\psi \notin F_\Omega(\pi)$. Then there is no $A \in W$ such that every agent $i \in A$ accepts $\neg\psi$. In particular $[\![\neg\psi]\!]_\pi \notin W$. But then by (M2b), $N - [\![\neg\psi]\!]_\pi = [\![\neg\neg\psi]\!]_\pi = [\![\psi]\!]_\pi \in W$, and thus $(F, \pi) \Vdash \Box\psi$. ■

One interpretation of the above result is that it shows the important rôle of families of majorities as simple games that are neither too conservative nor too resolute. Intuitively, if Ω is a family of majorities, then F_Ω selects either ψ or $\neg\psi$, and never both. These two horns are expressed by the following schemes:

$$D := \{\Box\varphi \to \neg\Box\neg\varphi \mid \varphi \in \mathscr{L}_c\} \qquad \text{(D)}$$

$$Dc := \{\neg\Box\neg\varphi \to \Box\varphi \mid \varphi \in \mathscr{L}_c\} \qquad \text{(Dc)}$$

We say that a SAF F is **generated by a simple game** if there is a simple game Ω such that $F = F_\Omega$.

Proposition 5 *Fix a set of agents N. The following are equivalent.*
(a) F is a SAF satisfying M, N, UD;
(b) F is a SAF generated by a simple game Ω.
Moreover, $F \Vdash D$ iff Ω is proper, and $F \Vdash Dc$ iff Ω is strong.

Proof. (a $\implies$ b). Call a set A ψ-decisive iff $\psi \in F(\psi)$ whenever $[\![\psi]\!]_\pi = A$. If F is neutral, then A is ψ-decisive if and only if there exists $\pi \in \Pi$ such that $[\![\psi]\!]_\pi = A$ and $\psi \in F(\pi)$. Let $W(\psi)$ the family of ψ-decisive sets. By monotony,

if A contains a ψ-decisive set, then A is a ψ-decisive set, so W satisfies (M1). By neutrality, for all $\psi, \varphi \in \mathscr{L}_c$, $W(\varphi) = W(\psi) =: W$. So (N, W) is a simple game, and it is straightforward to verify F is generated by (N, W).

If $F \Vdash D$ then Ω is proper: Suppose $F \Vdash D$, and that $A \in W$. Let π be any choice function such that $[\![q_1]\!]_\pi = A$, and so $(F, \pi) \Vdash \Box q_1$. Clearly $[\![\neg q_1]\!] = N - A$. By D, $F \models \neg\Box\neg q_1$, and thus $N - A \notin W$. If $F \Vdash Dc$ then Ω is strong: Suppose $F \Vdash Dc$. Suppose $A \notin W$. Let π be any choice function such that $[\![\neg q_1]\!]_\pi = A$ and $\neg q_1 \notin F(\pi)$. Then $(F, \pi) \Vdash \neg\Box\neg q_1$. By Dc, $F \models \Box q_1$. So $[\![q_1]\!]_\pi = N - A \in W$.

The other halves of the claims follow by lemma 4. ■

3.2 The Rudin-Keisler Ordering

The **Rudin-Keisler (RK) ordering** was introduced by Rudin as an ordering of ultrafilters (see Jech [7]). Taylor and Zwicker [14] observe that this ordering has a natural interpretation when applied to simple games.[2] Formally, if N and M are two sets of agents, and $\Omega = (N, W)$ is a simple game and f is a map from N to M, $f_*(W)$ is the subset of $\mathscr{P}(M)$ given by:

$$A \in f_*(W) \iff f^{-1}[A] \in W,$$

where $f^{-1}[A]$ is the preimage of A (that is: $\{i \in N \mid f(i) \in A\}$).

RK-*ordering and bloc formation.* When applied to simple games, the game $(M, f_*(W))$ is obtained intuitively by considering the players of Ω identified by f to vote as a bloc. The upshot of this is that any outcome arrived at in $(M, f_*(W))$ can be arrived at in Ω by letting these players vote *en bloc* in this manner.

The following definition of the Rudin-Keisler ordering differs from the one given by Taylor and Zwicker [14] and from the one familiar from the literature on ultrafilters in that we do not require f to be a surjection.

Definition 6 *We say that $\Omega = (N, W)$ is **RK-below** $\Omega' = (N', W')$, iff there exists a map f such that $W = f_*(W')$; in this case we write $\Omega \leq_{\mathrm{RK}} \Omega'$. Games Ω and Ω' are called **isomorphic** if $\Omega \leq_{\mathrm{RK}} \Omega' \leq_{\mathrm{RK}} \Omega$. We will write $\Omega \leq^{\mathrm{SUR}}_{\mathrm{RK}} \Omega'$ if there exists a* surjection *f such that $\Omega = f_*(\Omega')$. Finally, we say an* RK-*projection is **finite** if both N and N' are finite sets.*

If $\Omega \leq_{\mathrm{RK}} \Omega'$ then Ω is called an RK-projection of Ω'. It is not hard to see that the relations $\leq_{\mathrm{RK}}$ and $\leq^{\mathrm{SUR}}_{\mathrm{RK}}$ are transitive and reflexive (and hence pre-orderings) and that $\leq^{\mathrm{SUR}}_{\mathrm{RK}} \subset \leq_{\mathrm{RK}}$. Properties preserved by RK-projection include monotony, properness, and strongness.

Lemma 7 *If $\Omega \subseteq \mathscr{P}(N \cup A)$ is obtained from $\Omega' \subseteq \mathscr{P}(N)$ by adding a set of dummy players A, then Ω is isomorphic to Ω'.*

[2] In fact one does not even need to demand the monotony condition (M1) of the families of sets under consideration—the ordering also makes sense for arbitrary subsets of $\mathscr{P}(N)$.

The analogous claim for $\leq_{\text{RK}}^{\text{SUR}}$ quite obviously fails; which explains our choice of $\leq_{\text{RK}}$ as the default.[3] In fact, it is quite easy to see that if the projection function $f : N \to M$ isn't surjective, then the set $M - \text{ran}(f)$ will consist of dummy players. Hence any RK-projection may be decomposed in a surjective projection and an operation that adds dummies.

4 Majority Logic

We are now ready to begin a more systematic study of the language of group decisions, or majority logic, that was defined in section 2. It was alluded to above that the language $\mathscr{L}_{\Box}$ has a distinct modal flavour. In fact, we will take a look at SAFs as close cousins of the modal notion of a "frame". This way of looking at things is justified, at least for M-N-UD-SAFs that concern us in this text, by proposition 5. For instance, observe that simple games allow us to refine the first line in the truth conditions stated in definition 2:

$$(F_{(N,W)}, \pi) \Vdash \Box\varphi \quad \text{iff} \quad [\![\varphi]\!]_\pi \in W \tag{1}$$

The aim is to investigate the expressive power of $\mathscr{L}_{\Box}$. The next subsection looks at invariance results for the language, and we shall see that RK-projection plays a prominent rôle as a morphism between simple models. Thereafter, we expand our view and show how $\mathscr{L}_{\Box}$ fits into the richer modal logic. Finally, we apply tools from modal logic to arrive at some definability results.

4.1 Invariance Results

In this section we define two ways of creating new simple models out of old that preserve the truth of $\mathscr{L}_{\Box}$ formulae. The first two of these constructions stem from the game-theoretical literature on simple games and thus have a natural interpretation outside the logical framework considered in this text [14].

Definition 8 *Let $\Omega = (N, W)$ and $\Omega' = (N', W')$. The **product game** $\Omega \otimes \Omega$ is given by:*

$$(N \cup N', \{X \subseteq \mathscr{P}(N \cup N') \mid X \cap N \in W \text{ and } X \cap N' \in W'\})$$

*The **bicameral meet** $\Omega \sqcap \Omega'$ is the special case where N and N' are disjoint sets.*

The terminology "bicameral meet" comes from the idea that N and N' are two distinct "chambers", and a proposal has to pass both of these chambers to become accepted [14].

[3] Taylor and Zwicker [14] point out the possibility of dropping the surjectivity condition on f in this context.

Lemma 9 *Suppose $\Omega = (N, W)$ and $\Omega' = (N', W')$ are simple games such that N and N' are disjoint. Let π and π' be choice functions such that $dom(\pi) = N$ and $dom(\pi') = N'$, and let π'' be the choice function such that $dom(\pi'') = N \cup N'$, and $\pi''(i) = \pi(i)$ if $i \in N$, and $\pi''(i) = \pi'(i)$ if $i \in N'$. Let $\varphi \in \mathscr{L}_\Box$ and suppose $F_\Omega, \pi \Vdash \varphi$ and $F_{\Omega'}, \pi' \Vdash \varphi$. Then $F_{\Omega \sqcap \Omega'}, \pi'' \Vdash \varphi$.*

Proof. By induction on the complexity of φ. ■

RK-projection of simple games was already introduced in the previous section.

Definition 10 ***RK-projection of simple models.*** *The relation $\leq_{\mathrm{RK}}$ can be extended to simple models as follows. Let $(F_{(N,W)}, \pi)$ and $(F_{(N',W')}, \pi')$ be models. Define the relation $\leq^{\mathrm{M}}_{\mathrm{RK}}$ by:*

$$(F_{(N,W)}, \pi) \leq^{\mathrm{M}}_{\mathrm{RK}} (F_{(N',W')}, \pi') \text{ if and only if there is } f \text{ s.t.}$$
$$W = f_*(W') \text{ and for all } i \notin \mathscr{D}((N, W)), \pi(f(i)) = \pi(i).$$

It turns out that this notion of RK-projection is the most natural notion of morphism for simple models. From the perspective of modal logic this does not come as a great surprise, since the construction is akin to the familiar notion of bounded morphism [2]. (Note however that the dummy clause allows one to "throw away" information about certain players, and this has some subtle consequences.) $\mathscr{L}_\Box$-truths are invariant under RK-projection:

Lemma 11 *Let $(F_{(N,W)}, \pi) \leq^{\mathrm{M}}_{\mathrm{RK}} (F_{(N',W')}, \pi')$ and f such that $W = f_*(W')$ and for all $i \notin \mathscr{D}((N, W)), \pi(f(i)) = \pi(i)$. Then for all $\varphi \in \mathscr{L}_\Box$, $(F_{(N,W)}, \pi) \Vdash \varphi \iff (F_{(N',W')}, \pi') \Vdash \varphi$.*

Proof. By induction on the complexity of φ. ■

We will say that two simple models (F, π) and (F', π') are **isomorphic** if $(F, \pi) \leq^{\mathrm{M}}_{\mathrm{RK}} (F', \pi') \leq^{\mathrm{M}}_{\mathrm{RK}} (F, \pi)$. Clearly, if simple models are isomorphic, they make the same $\mathscr{L}_\Box$-formulae true. The converse, however, is false.

The final construction introduced here is inspired by the notion of ultraproducts known from modal logic, rather than by game theory. Let $\{(N_i, W_i)\}_{i \in I}$ be a family of simple games such that the sets N_i are disjoint. Let U be an ultrafilter over I; U may be thought of as the collection of "large subsets" of I.

Definition 12 ***Generalised Meet.*** *$\bigsqcap_U (N_i, W_i)$ is the simple game (N, W) such that:*

$$N = \bigcup_{i \in I} N_i, \qquad \text{and} \qquad X \in W \iff \{i \in I \mid X \cap N_i \in W_i\} \in U.$$

A $\varphi \in \mathscr{L}_\Box$ is true in $\bigsqcap_U (N_i, W_i)$ iff it is in a "large set" of underlying models:

Lemma 13 *Let $\{\pi_i : N_i \to \mathscr{P}(\mathsf{Q})\}_{i\in I}$ be a collection of choice functions, and let $\pi : \bigcup_{i\in I} N_i \to \mathscr{P}(\mathsf{Q})$ be the choice function such that $\pi(j)$ is just $\pi_i(j)$. For all $\varphi \in \mathscr{L}_{\Box}$, $\prod_U \Omega_i \Vdash \varphi \iff \{i \in I | \Omega_i \Vdash \varphi\} \in U$.*

Proof. By induction on the complexity of φ. ■

4.2 Majority Logic as a Fragment of Modal Logic

The language $\mathscr{L}_{\Box}$ is quite plainly a fragment of modal logic, $\mathscr{L}_{\Box\Box}$, which makes use of the grammar:

$$\psi ::= q \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \Box\psi \mid \bot \qquad \text{with each } q \in \mathsf{Q}$$

At the same time, the semantics provided by simple models can be seen as a fragment of the standard semantics for monotonic modal logic. Hence we obtain a relation between modal logic and majority logic at the semantic and the syntactic level. This relation is the subject of this subsection. Some familiarity with monotonic modal logic is assumed, refer to Hansen [6] for a thorough introduction. As a brief reminder, in monotonic modal logic formulae are interpreted using neighbourhood semantics:

Definition 14 *A (monotonic) **neighbourhood frame (n.f.)** is a pair (S, ν), S is a nonempty set of states, $\nu : S \to \mathscr{P}(\mathscr{P}(S))$ is the neighbourhood function; for each $s \in S$, $\nu(s)$ satisfies* (M1)*. A **neighbourhood model (n.m.)**, $\mathfrak{M} = (S, \nu, V)$, is a n.f. paired with a valuation $V : W \to \mathscr{P}(\mathsf{Q})$.*

Formulae of $\mathscr{L}_{\Box\Box}$ are interpreted relative to states, and the semantics of monotonic modal logic will be clear to anyone familiar with normal modal logic, with the possible exception of the modal clause:

$$\mathfrak{M}, s \Vdash \Box\psi \quad \text{iff} \quad \{s \in S \mid \mathfrak{M}, s \Vdash \psi\} \in \nu(w). \tag{2}$$

If a formula ψ is true globally (that is, at all states of a n.m.), we write $\mathfrak{M} \Vdash \psi$. If a formula is valid on a n.f. (i.e., true under all valuations) we write $(S, \nu) \Vdash \psi$.

Note that expression (2) contains essentially the same thought as (1) above. A simple model based on a simple game $\Omega = (W, N)$ and choice function π can be viewed as a n.m. where $\nu(i) = W$ and $V(i) = \pi(i)$ for all $i \in N$. For this reason (admittedly with *abuse de langage*) we will denote the corresponding n.m. (or n.f.) simply by (F, π) (or F), and use $\Vdash$ for the truth conditions of both $\mathscr{L}_{\Box}$ and $\mathscr{L}_{\Box\Box}$. Also from this perspective, an easy induction shows that formulae of $\mathscr{L}_{\Box}$ have the distinct property that if they are true at *some* state (or agent) in a simple model (F, π), they are true at *all* states.

The language $\mathscr{L}_{\Box\Box}$ can be used to express additional properties of SAFs.

Example 15 *Let $\Omega = (N, \{N\})$. F_Ω is the **consensus-SAF**. It can be shown that $F = F_\Omega$ if and only if F satisfies M, N, and UD and $F \Vdash \Box p \to p$.*

Hence among M-N-UD-SAFs, $\Box p \to p$ defines consensus; however, consensus is not expressible by majority logic, since this property is not invariant under adding dummies to Ω, and thus not invariant under RK-projection. We will show next that this is exactly the idea needed to separate $\mathscr{L}_{\Box}$ from $\mathscr{L}_{\Box\Box}$.

Definition 16 ***RK-Invariance.*** *Let (F,π) and (F,π') be simple models. A formula $\varphi \in \mathscr{L}_{\Box\Box}$ is* RK*-invariant iff whenever $(F,\pi), i \Vdash \varphi$ and $(F,\pi) \leq^{\mathrm{M}}_{\mathrm{RK}} (F',\pi')$, then there is a state (or agent) i' in the model (F',π') such that $(F',\pi'), i' \Vdash \varphi$. In words, satisfaction of φ is preserved under* RK*-projection.*

Proposition 17 *Let $\varphi \in \mathscr{L}_{\Box\Box}$. Then φ is equivalent to a formula $\psi \in \mathscr{L}_{\Box}$ on all simple models if and only if φ is* RK*-invariant.*

Possibly the proposition can be proved in a syntactic way, e.g. by using reductions to modal normal forms (*à la* Fine [4]). In this text our focus has been firmly on the semantic perspective, and we will seek a proof along the lines of the Van Benthem characterisation result, a corner stone of normal modal logic (see [2]). We need an auxiliary definition and result.

Definition 18 ***Monotonic bisimulation*** *[6, 4.10]. Suppose $\mathfrak{M} = (S,\nu,V)$ and $\mathfrak{M}' = (S',\nu V)$. Let $Z \subseteq S \times S'$ a nonempty relation. Z is a* ***bisimulation*** *between $\mathfrak{M}$ and $\mathfrak{M}'$ if the following three conditions hold: (Prop). If sZs', then s and s' satisfy the same sentence letters; (Forth). If sZs' and $X \in \nu(s)$, then there is $X' \subseteq S'$ such that $X' \in \nu'(s')$ and for all $s' \in X'$, there is $s \in X$ s.t. sZs'; (Back). If sZs' and $X' \in \nu'(s')$, then there is $X \subseteq S$ such that $X \in \nu(s)$ and for all $s \in X$, there is $s' \in X'$ s.t. sZs'.*

If Z is a bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ and sZs', then $\mathfrak{M}, s \Vdash \varphi$ if and only if $\mathfrak{M}', s' \Vdash \varphi$, for all φ in the modal language $\mathscr{L}_{\Box\Box}$ (and thus in $\mathscr{L}_{\Box}$).

Let us write $\mathfrak{M} \equiv \mathfrak{M}'$ just in case for all $\varphi \in \mathscr{L}_{\Box}$, for all states s of $\mathfrak{M}$, and for all states s' of $\mathfrak{M}'$ we have $\mathfrak{M}, s \Vdash \varphi \iff \mathfrak{M}', s' \Vdash \varphi$.

Lemma 19 ***Collapse of Bisimulation.*** *Suppose $\mathfrak{M} \equiv \mathfrak{M}'$. Let Z be the relation where sZs' if and only if s and s' satisfy the same sentence letters. Then Z is a bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Proof. The proof uses ideas from Hansen [6], Proposition 4.31. Let $\mathfrak{M} = (S,\nu,V)$ and $\mathfrak{M}' = (S',\nu',V')$. (Prop). is clear. (Forth). Suppose sZs' and take $X \in \nu(s)$. We would like to find $X' \in \nu'(s')$ such that $\forall s' \in X'$, there is $s \in X$ for which xZx' holds.

Now towards a contradiction suppose there is no such X'. Then for every $Y \in \nu'(s')$, there is an y_i such that for all $x_j \in X$, it is not true that $x_j Z y_i$. This means y_i and x_j differ in their sentence letters, and there must be literals witnessing this; for instance: $y_i \Vdash \neg q$ and $x_i \nVdash \neg q$. Pick one and denote the literal true at y_i but not at x_j by φ_{ij}. Let Δ_i be the set: $\{\varphi_{i'j} \mid i = i'\}$. By

construction, for each y_i we have $\mathfrak{M}, y_i \Vdash \bigwedge \Delta_i$. Note that Δ_i is a finite set, since there are only finitely many literals given our assumption on Q. Hence:

$$\mathfrak{M}', s' \Vdash \neg\Box\neg \bigvee_i \bigwedge \Delta_i, \tag{3}$$

$$\text{however, } \mathfrak{M}, s \nVdash \neg\Box\neg \bigvee_i \bigwedge \Delta_i. \tag{4}$$

Since there are only finitely many literals, there can be only finitely many different sets Δ_i. Hence without loss of generality, we may assume any disjunction over a conjunction of them is finite; and thus $\neg \bigvee_i \bigwedge \Delta_i \in \mathscr{L}_c$ since it is a finite formula build from propositions, $\wedge$, $\vee$, $\neg$. Hence $\neg\Box\neg \bigvee_i \bigwedge \Delta_i \in \mathscr{L}_{\Box}$. Clearly, the discrepancy between (3) and (4) contradicts the fact that $\mathfrak{M} \equiv \mathfrak{M}'$. The (Back) clause can be proved in similar fashion. ■

Proof of proposition 17. Left-to-right follows from the invariance results above. As for the other direction, we will make use of the fact that the standard translation for monotonic neighbourhood semantics allows us to pass between first order logic and $\mathscr{L}_{\Box\Box}$, see again [6] for details. The standard translation of an $\mathscr{L}_{\Box\Box}$-formula χ is denoted $ST_s(\chi)$ (s is the state it is evaluated at). We use $\models$ for the first order entailment relation, for the purpose of this proof.

Assume φ is RK-invariant. Let C be a first order formula expressing that ν is a constant function. Define the set of $\mathscr{L}_{\Box}$-consequences of φ:

$$\mathrm{MLC}(\varphi) := \{ST_s(\chi) \mid \chi \in \mathscr{L}_{\Box} \text{ and } ST_s(\varphi) \cup \{C\} \models ST_s(\chi)\}.$$

If $\{C\} \cup \mathrm{MLC}(\varphi) \models ST_x(\varphi)$, then by compactness φ is equivalent to a formula $\psi \in \mathscr{L}_{\Box}$ on models satisfying C, hence on simple models. Therefore we will show $\{C\} \cup \mathrm{MLC}(\varphi) \models ST_x(\varphi)$. Assume that $\mathfrak{M} \models \{C\} \cup \mathrm{MLC}(\varphi)[s]$. We can view $\mathfrak{M}$ as some simple model (F, π). Say $F = F_\Omega$, $\Omega = (N, W)$.

Let $T = \{\forall x ST_x(\xi) \mid F, s \models \xi, \text{ and } \varphi \in \mathscr{L}_{\Box}\}$; $\mathfrak{M} \models T$. We claim $T \cup ST_y(\varphi)$ is consistent. For suppose not, then by compactness some finite subset T_0 of T is inconsistent with $ST_y(\varphi)$, and we have $ST_y(\varphi) \rightarrow \neg \bigwedge T_0$. Hence $ST_y(\varphi) \rightarrow \{\exists x \neg ST_x(\xi_1) \vee \cdots \vee \exists x \neg ST_x(\xi_k)\}$. But then $C \cup ST_y(\varphi) \models \forall x \neg ST_x(\xi_1) \vee \cdots \vee \forall x \neg ST_x(\xi_k)\}$ (using the fact that C forces a constant neighbourhood function). Hence it must be that $\bigvee_{j \in \{1,\ldots,k\}} \neg ST_s(\xi_k) \in \mathrm{MLC}(\varphi)$. But this contradicts $T_0 \subseteq T$. So $T \cup ST_y(\varphi)$ is consistent, and hence can be satisfied in some model, say $\mathfrak{N} = (S, \nu, V)$, at some state s^*. Since $\mathfrak{N} \models T$, we know $\mathfrak{N}$ makes exactly the same $\mathscr{L}_{\Box}$-formulae true as F, and thus $\mathfrak{N} \equiv (F, \pi)$. Now let:

$$D := \{V(s) \mid s \in S \text{ and there is no } i \in N, \pi(i) = V(s)\}.$$

We can add dummies to Ω to account for these all 'missing valuations', and obtain a simple model (F', π'); $(F, \pi) \leq^{\mathrm{M}}_{\mathrm{RK}} F', \pi')$. Suppose $F = F_{N', W'}$. Let $Z \subseteq S \times N'$ be the relation where sZi if and only if s and i satisfy the same sentence letters. By the previous lemma Z is a bisimulation. Moreover, there is a state i^* such that $s^* Z i^*$. Hence $F' \models ST_y(\varphi)[i^*]$. By our invariance assumption, there is $j \in N$, such that $F \models ST_y(\varphi)[j]$—as required. ■

4.3 Axiomatic Social Choice and N.f. Definability

Consider again example 15 above. It illustrates an important conceptual point. In social choice theory, axioms are used to pick out certain classes of social aggregation functions. In modal logic, frame validity gives a handle on the definability of frame classes. The $\mathscr{L}_{\Box\Box}$ formula $\Box p \to p$ picks out the simple game $(N, \{N\})$, which is identified with the consensus-SAF. Thus "frame definability", or in the present framework rather "simple game" definability, is the natural logical counterpart to the axiomatic approach to social choice. Modal-like languages gives us a precise logical tool to formulate certain kinds of axioms studied in social choice and it is then natural to ask about the expressive strengths of logical languages: are there limits on their expressive power? How does $\mathscr{L}_{\Box}$ sit inside $\mathscr{L}_{\Box\Box}$? The next two results provides partial answers to some of such questions. They also underline once more the fundamental importance of the notion of RK-projection.

Definition 20 *Let* K *a class of M-N-UD-SAFs. We say* K *is closed under* RK*-projection if the set* $\{\Omega \mid F_\Omega \in \mathsf{K}\}$ *is closed under* RK*-projection. Similarly for bicameral meet, etc.*

Proposition 21 *Let* K *a class of M-N-UD-SAFs.* K *is definable by a set of* $\mathscr{L}_{\Box}$*-formulae only if it is closed under* RK*-projections and bicameral meet.*

Proof. This follows from the invariance results stated in subsection 4.1. ■

Proposition 22 *Let* K *be a class of M-N-UD-SAFs that is definable by a set of* $\mathscr{L}_{\Box\Box}$ *formulae and that is closed under generalised meet. Then* K *is definable by a set of* $\mathscr{L}_{\Box}$ *formulae if and only if it is closed under* RK*-projection.*

Proof. Suppose K is definable by an $\mathscr{L}_{\Box\Box}$ theory S. We will show the $\mathscr{L}_{\Box}$ theory of K, $\Lambda^{\mathsf{K}}_{\Box}$, defines K, along the lines of a fairly standard argument from modal logic [2]. Suppose the contrary. Then there exists a simple model $\mathfrak{M}$, whose underlying simple game isn't in K, such that $\mathfrak{M} \Vdash \Lambda^{\mathsf{K}}_{\Box}$ but for some state s, $\mathfrak{M}, s \Vdash \neg\psi$, where $\psi \in S$. Let $\Lambda^{\mathsf{M}}_{\Box}$ be the $\mathscr{L}_{\Box}$ theory of $\mathfrak{M}$. Every finite subset of $\Lambda^{\mathsf{M}}_{\Box}$ is satisfiable in some model (Ω, π) in K—for suppose not, then there is a finite subset $F \subseteq \Lambda^{\mathsf{M}}_{\Box}$, $\neg \bigwedge F \in \Lambda^{\mathsf{K}}_{\Box}$, but this contradicts $\mathfrak{M} \Vdash \Lambda^{\mathsf{K}}_{\Box}$.

Define an index set I such that $I = \{F \subseteq \Lambda^{\mathsf{M}}_{\Box} \mid F \text{ is finite}\}$. For each $i \in I$ there is a simple model Ω_i, π_i such that $\Omega_i \Vdash i$. Because K is closed under RK-projection, we may take these models disjoint. For each $\varphi \in S$, let $\hat{\varphi}$ the set of all $i \in I$ that contain $\hat{\varphi}$. The set $\{\hat{\varphi} \mid \varphi \in \S\}$ has the finite intersection property and thus can be extended to an ultrafilter U. Now let $\pi : \bigcup_{i \in I} N_i \to \mathscr{P}(\mathsf{Q})$ be the choice function such that $\pi(j)$ is just $\pi_i(j)$. Then $\prod_U \Omega_i, \pi \Vdash \Lambda^{\mathsf{M}}_{\Box}$. This is true, since for each $i \in \hat{\varphi}$, we have $\varphi \in i$, and hence $\Omega_i, \pi_i \Vdash \varphi$. Therefore $\{i \in I \mid \Omega_i \Vdash \varphi\} \supseteq \hat{\varphi} \in U$ and thus by lemma 13, $\prod_U \Omega_i, \pi \Vdash \varphi$.

Given this model $\prod_U \Omega_i =: \mathfrak{M}^*$, by the closure conditions of K, $\Lambda^{\mathsf{M}}_{\Box}$ is satisfiable on a simple game in K. Now $\mathfrak{M}^* \equiv \mathfrak{M}$. Like in the proof of proposition 17,

we may add dummies to obtain a model $\mathfrak{M}^{**}$ and a state s^{**} bisimilar to the state s of $\mathfrak{M}$. It follows $\mathfrak{M}^{**}, s^{**} \Vdash \neg\varphi$ and hence $\mathfrak{M}^{**} \not\Vdash S$. But $\mathfrak{M}^{*} \leq^{\mathrm{M}}_{\mathrm{RK}} \mathfrak{M}^{**}$, and thus the underlying simple game is in K, a contradiction. ■

5 Concluding Remarks

The main theme of this text is that—from a logical point of view—the axiomatic approach to social choice (which should be distinguished from the *logical* idea of providing axiomatisations) quite naturally corresponds to the investigation of definability results in an appropriately chosen logical language. Let us conclude with two remarks that put this message into some wider perspective.

The first remark is illustrative in nature. While we have not been concerned explicitly with the impossibility results obtained in social choice theory, they emerge quite easily from our framework. To avoid that logical inconsistencies arise in the aggregation process, we would like a SAF to respect the rules of classical logic. In "axiomatic" terms, what we need is the SAF to validate the formula $\Box p \leftrightarrow \neg\Box\neg p$, and in addition we want $\Box$ to distributive: $\Box p \wedge \Box q \leftrightarrow \Box(p \wedge q)$. These $\mathscr{L}_{\Box}$-formulae force the underlying simple game to be strong and proper, and closed under finite intersections. It is well known that the only simple games that satisfy these properties correspond to the ultrafilters (indeed the formulae *define* this class of M-N-UD-SAFs); and hence the impossibility results emerge.

In addition, one might be interested in other behavioural properties of SAFs. A precise choice of logical language—majority logic in this text—allows us to get a firm logical grip on the axioms that can be formulated within a language, and then to compare the expressive power and relative complexity of different languages for the purpose of axiomatic social choice theory. To this end, one can apply tools from the logician's toolbox to study what properties of SAFs can and can't be defined in the language, and investigate the logical consequences. This has been the main subject matter of this text. It is worth to point out a related technical observation. We have argued that the semantics for the language $\mathscr{L}_{\Box}$ somehow sit as a fragment inside the more well-known neighbourhood semantics, and have made good use of this fact too. However, it turns out that the fragment is less well behaved than one might expect on an *a priori* basis. When comparing the structural truth preserving operations set out in section 4.1 with those familiar from modal logic, they appear closely related. What seems to be lacking, however, is an analogue for the disjoint union construction, which functions rather prominently in modal definability results. For future work it remains to be investigated which modal tools can, and which tools can't be applied under this limitation.

References

[1] Arrow, K. *Social Choice and Individual Values*, 2nd ed. No. 12 in Cowles

Foundation Monographs. New York etc.: Wiley, 1963.

[2] Blackburn, P., de Rijke, M., and Venema, Y. *Modal Logic*. Cambridge: Cambridge University Press, 2001.

[3] Dietrich, F., and List, C. Arrow's theorem in judgement aggregation. *Social Choice and Welfare* (forthcoming).

[4] Fine, K. Normal forms in modal logic. *Notre Dame journal of formal logic 16*, 2 (1975), 229–37.

[5] Guilbaud, G. Th. Les théories de l'intérêt général et le problème logique de l'agrégation. *Économie Appliquée 5*, 4 (1952), 501–84.

[6] Hansen, H. H. Monotonic modal logics. ILLC Preprint Series PP-2003-24, Institute of Logic, Language and Computation, Universiteit van Amsterdam, 2003.

[7] Jech, T. *Set Theory*, third millenion edition ed. Springer Monographs in Mathematics. Berlin etc.: Springer-Verlag, 2006.

[8] List, C., and Pettit, P. Aggregating sets of judgements: An impossibility result. *Economics and Philosophy 18* (2002), 89–110.

[9] Monjardet, B. An axiomatic theory of tournament aggregation. *Mathematics of Operations Research 3*, 4 (1978), 334–51.

[10] Nehring, K., and Puppe, C. Consistent judgement aggregation: A characterization. Tech. rep., University of Karlsruhe, 2005.

[11] Neumann, J. V., and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton: Princeton University Press, 1944.

[12] Pauly, M. Axiomatising judgement aggregation procedures in a minimal logical language. Unpublished version dated 30th March 2005.

[13] Pauly, M., and van Hees, M. Logical constraints on judgement aggregation. *Journal of Philosophical Logic* (forthcoming).

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

Tijmen R. Daniëls
Amsterdam School of Economics, and Tinbergen Institute,
Universiteit van Amsterdam
Roetersstraat 11
1018 WB Amsterdam
The Netherlands
Email: tijmen.daniels@uva.nl

Judgment aggregation without full rationality: a summary

Franz Dietrich and Christian List[1]

Abstract

Several recent results on the aggregation of judgments over logically connected propositions show that, under certain conditions, dictatorships are the only independent (i.e., propositionwise) aggregation rules generating fully rational (i.e., complete and consistent) collective judgments. A frequently mentioned route to avoid dictatorships is to allow incomplete collective judgments. We show that this route does not lead very far: we obtain (strong) oligarchies rather than dictatorships if instead of full rationality we merely require that collective judgments be deductively closed, arguably a minimal condition of rationality (compatible even with empty judgment sets). We derive several characterizations of oligarchies and provide illustrative applications to Arrowian preference aggregation and Kasher and Rubinstein's group identification problem.

1 Introduction

Sparked by the "discursive paradox", the problem of "judgment aggregation" has recently received much attention. The "discursive paradox", of which Condorcet's famous paradox is a special case, consists in the fact that, if a group of individuals takes majority votes on some logically connected propositions, the resulting collective judgments may be inconsistent, even if all group members' judgments are individually consistent (Pettit 2001, extending Kornhauser and Sager 1986; List and Pettit 2004). A simple example is given in Table 1.

	a	b	$a \wedge b$
Individual 1	True	True	True
Individual 2	True	False	False
Individual 3	False	True	False
Majority	True	True	False

Table 1: A discursive paradox

Several subsequent impossibility results have shown that majority voting is not alone in its failure to ensure rational (i.e., complete and consistent) collective judgments when propositions are interconnected (List and Pettit 2002, Pauly and van Hees 2006, Dietrich 2006, Gärdenfors 2006, Nehring and Puppe 2002, 2005, van Hees forthcoming, Dietrich forthcoming, Mongin 2005, Dokow

[1]Franz Dietrich: Maastricht University; Christian List: London School of Economics

and Holzman 2005, Dietrich and List forthcoming-a). The generic finding is that, under the requirement of proposition-by-proposition aggregation (independence), dictatorships are the only aggregation rules generating complete and consistent collective judgments and satisfying some other conditions (which differ from result to result). This generic finding is broadly analogous to Arrow's theorem for preference aggregation. (Precursors to this recent literature are Wilson's 1975 and Rubinstein and Fishburn's 1986 contributions on abstract aggregation theory.)

A frequently mentioned escape route from this impossibility is to drop the requirement of complete collective judgments and thus to allow the group to make no judgment on some propositions. Examples of aggregation rules that ensure consistency at the expense of incompleteness are unanimity and certain supermajority rules (List and Pettit 2002, List 2004, Dietrich and List forthcoming-b).

The most forceful critique of the completeness requirement has been made by Gärdenfors (2006), in line with his influential theory of belief revision (e.g., Alchourron, Gärdenfors and Makinson 1985). Describing completeness as a "strong and unnatural assumption", Gärdenfors has argued that neither individuals nor a group need to hold complete judgments and that, in his opinion, "the [existing] impossibility results are consequences of an unnaturally strong restriction on the outcomes of a voting function". Gärdenfors has also proved the first and so far only impossibility result on judgment aggregation without completeness, showing that, under certain conditions, any aggregation rule generating consistent and deductively closed (but not necessarily complete) collective judgments, while not necessarily dictatorial, is weakly oligarchic.

In this paper, we continue this line of research and investigate judgment aggregation without the completeness requirement. We drop this requirement, first at the collective level and later at the individual level, replacing it with the weaker requirement of merely deductively closed judgments. Our results do not need the requirement of collective consistency. Under standard conditions on aggregation rules and the weakest possible assumptions about the agenda of propositions under consideration, we provide the first characterizations of (strong) oligarchies (without a default)[2] and the first characterization of the unanimity rule[3] (the only anonymous oligarchy). As corollaries, we also obtain new variants of several characterizations of dictatorships in the literature (using no consistency condition).

Our results strengthen Gärdenfors's oligarchy results in three respects. First, they impose weaker conditions on aggregation rules. Second, they show that strong and not merely weak oligarchies are implied by these conditions and fully

[2]For truth-functional agendas, Nehring and Puppe (2005) have characterized *oligarchies with a default*, which are distinct from the *(strong or weak) oligarchies* considered by Gärdenfors (2006) and in this paper. Oligarchies with a default by definition generate complete collective judgments.

[3]Again without a default, thus with possibly incomplete outcomes.

characterize strong oligarchies. Third, they do not require the logically rich and infinite agenda of propositions Gärdenfors assumes. They reinforce Gärdenfors's arguments, however, in showing that, under surprisingly mild conditions, we are restricted to oligarchic aggregation rules.

In judgment aggregation, one can distinguish between *impossibility results* (like Gärdenfors's results) and *characterizations of impossibility agendas* (like the present results and the results cited below). The former show that, for certain agendas of propositions, aggregation in accordance with certain conditions is impossible or severely restricted (e.g., to dictatorships or oligarchies). The latter characterize the precise class of agendas for which such an impossibility or restriction arises (and hence the class of agendas for which it does not arise). Characterizations of impossibility agendas have the merit of identifying precisely which kinds of decision problems are subject to the impossibility results in question and which are free from them. (Notoriously, preference aggregation problems are subject to most such impossibility results.) There has been much recent progress on such characterizations. Nehring and Puppe (2002) were the first to prove such results. Subsequent results have been derived by Dokow and Holzman (2005), Dietrich (forthcoming) and Dietrich and List (forthcoming-a). But so far all characterizations of impossibility agendas assume fully rational collective judgments. We here give the first characterizations of impossibility agendas without requiring complete (nor even consistent) collective judgments.

All proofs are given in Dietrich and List (2006).

2 The model

Consider a set of individuals $N = \{1, 2, \ldots, n\}$ ($n \geq 2$) seeking to make collective judgments on some logically connected propositions. To represent propositions, we introduce a logic, using Dietrich's (forthcoming) general logics framework (generalizing List and Pettit 2002, 2004). A *logic (with negation symbol* $\neg$*)* is a pair $(\mathbf{L}, \vDash)$ such that

(i) $\mathbf{L}$ is a non-empty set of formal expressions (*propositions*) closed under negation (i.e., $p \in \mathbf{L}$ implies $\neg p \in \mathbf{L}$), and

(ii) $\vDash$ is a binary (*entailment*) relation ($\subseteq \mathcal{P}(\mathbf{L}) \times \mathbf{L}$), where, for each $A \subseteq \mathbf{L}$ and each $p \in \mathbf{L}$, $A \vDash p$ is read as "A entails p".

A set $A \subseteq \mathbf{L}$ is *inconsistent* if $A \vDash p$ and $A \vDash \neg p$ for some $p \in \mathbf{L}$, and *consistent* otherwise. Our results hold for any logic $(\mathbf{L}, \vDash)$ satisfying four minimal conditions;[4] this includes standard propositional, predicate, modal and

[4] L1 (self-entailment): For any $p \in \mathbf{L}$, $\{p\} \vDash p$. L2 (monotonicity): For any $p \in \mathbf{L}$ and any $A \subseteq B \subseteq \mathbf{L}$, if $A \vDash p$ then $B \vDash p$. L3 (completability): $\emptyset$ is consistent, and each consistent set $A \subseteq \mathbf{L}$ has a consistent superset $B \subseteq \mathbf{L}$ containing a member of each pair $p, \neg p \in \mathbf{L}$. L4 (non-paraconsistency): For any $A \subseteq \mathbf{L}$ and any $p \in \mathbf{L}$, if $A \cup \{\neg p\}$ is inconsistent then $A \vDash p$. In L4, the converse implication also holds given L1-L3. See Dietrich (forthcoming, Section 4) for the main properties of entailment and inconsistency under L1-L4.

conditional logics. For example, in standard propositional logic, $\mathbf{L}$ contains propositions such as a, b, $a \wedge b$, $a \vee b$, $\neg(a \rightarrow b)$, and $\vDash$ satisfies $\{a, a \rightarrow b\} \vDash b$, $\{a\} \vDash a \vee b$, but not $a \vDash a \wedge b$.

A proposition $p \in \mathbf{L}$ is a *tautology* if $\{\neg p\}$ is inconsistent, and a *contradiction* if $\{p\}$ is inconsistent. A proposition $p \in \mathbf{L}$ is *contingent* if it is neither a tautology nor a contradiction. A set $A \subseteq \mathbf{L}$ is *minimal inconsistent* if it is inconsistent and every proper subset $B \subsetneq A$ is consistent.

The *agenda* is a non-empty subset $X \subseteq \mathbf{L}$, interpreted as the set of propositions on which judgments are to be made, where X can be written as $\{p, \neg p : p \in X^*\}$ for a set $X^* \subseteq \mathbf{L}$ of unnegated propositions. For notational simplicity, double negations within the agenda cancel each other out, i.e., $\neg\neg p$ stands for p.[5] In the example above, the agenda is $X = \{a, \neg a, b, \neg b, a \wedge b, \neg(a \wedge b)\}$ in standard propositional logic. Informally, an agenda captures a particular decision problem.

An *(individual or collective) judgment set* is a subset $A \subseteq X$, where $p \in A$ means that proposition p is accepted (by the individual or group). Different interpretations of "acceptance" can be given. On the standard interpretation, to accept a proposition means to believe it, so that judgment aggregation is the aggregation of (binary) belief sets. On an entirely different interpretation, to accept a proposition means to desire it, so that judgment aggregation is the aggregation of (binary) desire sets.

A judgment set $A \subseteq X$ is

(i) *consistent* if it is a consistent set in $\mathbf{L}$,
(ii) *complete* if, for every proposition $p \in X$, $p \in A$ or $\neg p \in A$,
(iii) *deductively closed* if, for every proposition $p \in X$, if $A \vDash p$ then $p \in A$.

Note that the conjunction of consistency and completeness implies deductive closure, while the converse does not hold (Dietrich forthcoming, List 2004). Deductive closure can be met by "small", even empty, judgment sets $A \subseteq X$. Hence deductive closure is a much weaker requirement than "full rationality" (the conjunction of consistency and completeness). Let $\mathcal{C}$ be the set of all complete and consistent (and hence also deductively closed) judgment sets $A \subseteq X$. A *profile* is an n-tuple $(A_1, \ldots, A_n)$ of individual judgment sets.

A *(judgment) aggregation rule* is a function F that assigns to each admissible profile $(A_1, \ldots, A_n)$ a collective judgment set $F(A_1, \ldots, A_n) = A \subseteq X$. The set of admissible profiles is denoted $Domain(F)$.

Call F *universal* if $Domain(F) = \mathcal{C}^n$; call it *consistent*, *complete*, or *deductively closed* if it generates a consistent, complete, or deductively closed collective judgment set $A = F(A_1, \ldots, A_n)$ for every profile $(A_1, \ldots, A_n) \in Domain(F)$; call it *unanimity-respecting* if $F(A, ..., A) = A$ for all unanimous profiles $(A, ..., A) \in Domain(F)$; and call it *anonymous* if, for any profiles $(A_1, \ldots, A_n)$, $(A_1^*, \ldots, A_n^*) \in Domain(F)$ that are permutations of each other,

[5] To be precise, when we use the negation symbol $\neg$ hereafter, we mean a modified negation symbol $\sim$, where $\sim p := \neg p$ if p is unnegated and $\sim p := q$ if $p = \neg q$ for some q.

$F(A_1, \ldots, A_n) = F(A_1^*, \ldots, A_n^*)$.

Examples of aggregation rules are *majority voting*, where, for each $(A_1, ..., A_n) \in \mathcal{C}^n$, $F(A_1, ..., A_n) = \{p \in X : |\{i \in N : p \in A_i\}| > |\{i \in N : p \notin A_i\}|\}$ and a *dictatorship* of some individual $i \in N$, where, for each $(A_1, ..., A_n) \in \mathcal{C}^n$, $F(A_1, ..., A_n) = A_i$. Majority voting and dictatorships are each universal and unanimity-respecting. Majority voting is anonymous while dictatorships are not. But, as the "discursive paradox" shows, majority voting is not consistent (or deductively closed) (and it is complete if and only if n is odd), while dictatorships are consistent, complete and deductively closed. For some agendas X, so-called premise-based and conclusion-based aggregation rules can be defined.

The model can represent various realistic decision problems, including Arrowian preference aggregation problems and Kasher and Rubinstein's group identification problem, as illustrated in Sections 4 and 5.

3 Characterization results

Are there any appealing aggregation rules F if we allow incomplete outcomes? Our results share with previous results the requirement of *propositionwise aggregation*: the group "votes" independently on each proposition, as captured by the following condition.

Independence. For any $p \in X$ and any $(A_1, \ldots, A_n)$, $(A_1^*, \ldots, A_n^*) \in Domain(F)$, if [for all $i \in N$, $p \in A_i \Leftrightarrow p \in A_i^*$] then $p \in F(A_1, \ldots, A_n) \Leftrightarrow p \in F(A_1^*, \ldots, A_n^*)$.

Interpretationally, independence requires the group judgment on any given proposition $p \in X$ to "supervene" on the individual judgments on p (List and Pettit forthcoming). This reflects a "local" notion of democracy, which could for instance be viewed as underlying direct democratic systems that are based on referenda on various propositions. If we require the group not only to vote independently on the propositions, but also to use the same voting method for each proposition (a neutrality condition), we obtain the following stronger condition.

Systematicity. For any $p, q \in X$ and any $(A_1, \ldots, A_n)$, $(A_1^*, \ldots, A_n^*) \in Domain(F)$, if [for all $i \in N$, $p \in A_i \Leftrightarrow q \in A_i^*$] then $p \in F(A_1, \ldots, A_n) \Leftrightarrow q \in F(A_1^*, \ldots, A_n^*)$.

Some of our results require systematicity (and not just independence), and some also require the following responsiveness property.

Monotonicity. For any $(A_1, ..., A_n) \in Domain(F)$, we have $F(A_1^*, ..., A_n^*) = F(A_1, ..., A_n)$ for all $(A_1^*, ..., A_n^*) \in Domain(F)$ arising from $(A_1, ..., A_n)$ by replacing one A_i by $F(A_1, ..., A_n)$.

Monotonicity states that changing one individual's judgment set towards the present outcome (collective judgment set) does not alter the outcome.[6]

We call an aggregation rule F a *(strong) oligarchy* (dropping "strong" whenever there is no ambiguity) if it is universal and given by

$$F(A_1, .., A_n) = \cap_{i \in M} A_i \text{ for all profiles } (A_1, ..., A_n) \in \mathcal{C}^n, \tag{1}$$

where $M \subseteq N$ is fixed non-empty set (of *oligarchs*). A *weak oligarchy* is a universal aggregation rule F such that there exists a smallest winning coalition, i.e., a smallest non-empty set $M \subseteq N$ that satisfies (1) with "=" replaced by "$\supseteq$".[7] An oligarchy (respectively, weak oligarchy) accepts all (respectively, at least all) propositions unanimously accepted by the oligarchs.

Interesting impossibility results on judgment aggregation never apply to all agendas X (decision problems). Typically, impossibilities using the (strong) systematicity condition apply to most relevant agendas, while impossibilities using the (weaker) independence condition apply to a class of agendas that both includes and excludes many relevant agendas. Our present results confirm this picture.

We here use two weak agenda conditions (for our systematicity results) and one much stronger one (for our independence results). For any sets $Z \subseteq Y \subseteq X$, let $Y_{\neg Z}$ denote the set $(Y \backslash Z) \cup \{\neg p : p \in Z\}$, which arises from Y by negating the propositions in Z. The two weak conditions are the following.

(α) There is an inconsistent set $Y \subseteq X$ with pairwise disjoint subsets $Z_1, Z_2, \{p\}$ such that $Y_{\neg Z_1}$, $Y_{\neg Z_2}$ and $Y_{\neg \{p\}}$ are consistent.
(β) There is an inconsistent set $Y \subseteq X$ with disjoint subsets $Z, \{p\}$ such that $Y_{\neg Z}$, $Y_{\neg \{p\}}$ and $Y_{\neg (Z \cup \{p\})}$ are consistent.

These conditions are not *ad hoc*. As shown later, they are the weakest possible conditions needed for our results. If X is finite or the logic compact, (α) and (β) become equivalent to, respectively, the following standard conditions (see Dietrich and List 2006).

(i) There is a minimal inconsistent set $Y \subseteq X$ with $|Y| \geq 3$.
(ii) There is a minimal inconsistent set $Y \subseteq X$ such that $Y_{\neg Z}$ is consistent for some subset $Z \subseteq Y$ of even size (the *even-number negation* condition

[6] This is a judgment-set-wise monotonicity condition, which differs from a proposition-wise one (e.g., Dietrich and List 2005). Similarly, our condition of unanimity-respectance is judgment-set-wise rather than proposition-wise. One may consider this as an advantage, since a flavour of independence is avoided, so that the conditions in the characterisation are in the inutitive sense "orthogonal" to each other.

[7] The term "oligarchy" (without further qualification) refers to a strong oligarchy, whereas in Gärdenfors (2006) it refers to a weak one. A distinct oligarchy notion is Nehring and Puppe's (2005) "oligarchy with a default", which always generates complete collective judgments by reverting to a default on each pair $p, \neg p \in X$ on which the oligarchs disagree.

in Dietrich (forthcoming) and Dietrich and List (forthcoming-a), which for finite X is equivalent to Dokow and Holzman's (2005) *non-affineness* condition).

These conditions hold for most standard examples of judgment aggregation agendas X. For instance, if X contains propositions $a, b, a \wedge b$ as in the example of Table 1, then in (i) and (ii) we can take $Y = \{a, b, \neg(a \wedge b)\}$, where in (ii) $Z = \{a, b\}$. If X contains propositions $a, a \rightarrow b, b$ ("$\rightarrow$" could be a subjunctive implication) then in (i) and (ii) we can take $Y = \{a, a \rightarrow b, \neg b\}$, where in (ii) $Z = \{a, \neg b\}$. In Sections 4 and 5, we show that the conditions also hold for agendas representing Arrowian preference aggregation or Kasher and Rubinstein's group identification problem.

The stronger agenda condition, used in Theorem 2, is that of *path-connectedness*, a variant of Nehring and Puppe's (2002) *total blockedness* condition. For any $p, q \in X$, we write $p \vDash^* q$ (p *conditionally entails* q) if $\{p\} \cup Y \vDash q$ for some $Y \subseteq X$ consistent with p and with $\neg q$. For instance, for the agenda $X = \{a, \neg a, b, \neg b, a \wedge b, \neg(a \wedge b)\}$, we have $a \wedge b \vDash^* a$ (take $Y = \emptyset$) and $a \vDash^* \neg b$ (take $Y = \{\neg(a \wedge b)\}$). An agenda X is *path-connected* if, for every contingent $p, q \in X$, there exist $p_1, p_2, ..., p_k \in X$ (with $p = p_1$ and $q = p_k$) such that $p_1 \vDash^* p_2$, $p_2 \vDash^* p_3$, ..., $p_{k-1} \vDash^* p_k$.

The agenda $X = \{a, \neg a, b, \neg b, a \wedge b, \neg(a \wedge b)\}$ is *not* path-connected: for a negated proposition ($\neg a$ or $\neg b$ or $\neg(a \wedge b)$), there is no path to a non-negated proposition. By contrast, as discussed in Sections 4 and 5, the agendas for representing Arrowian preference aggregation problems or Kasher and Rubinstein's group identification problem are path-connected.

Theorem 1 *Let the agenda X satisfy (α) and (β).*

(a) *The oligarchies are the only universal, deductively closed, unanimity-respecting and systematic aggregation rules.*
(b) *Part (a) continues to hold if the agenda condition (β) is dropped and the aggregation condition of monotonicity is added.*

Theorem 2 *Let the agenda X satisfy path-connectedness and (β).*

(a) *The oligarchies are the only universal, deductively closed, unanimity-respecting and independent aggregation rules.*
(b) *Part (a) continues to hold if the agenda condition (β) is dropped and the aggregation condition of monotonicity is added.*

Proofs are given in the Appendix. Theorems 1 and 2 provide four characterizations of oligarchies. They differ in the conditions imposed on aggregation rules and the agendas permitted. Part (a) of Theorem 2 is perhaps the most surprising result, as it characterizes oligarchies on the basis of the logically weakest set of conditions on aggregation rules. We later apply this result to Arrowian preference aggregation problems and Kasher and Rubinstein's group identification problem.

In each characterization, adding the condition of anonymity eliminates all oligarchies except the *unanimity rule* (i.e., the oligarchy with $M = N$), and adding the condition of completeness eliminates all oligarchies except dictatorships (as defined above). So we obtain characterizations of the unanimity rule and of dictatorships.

Corollary 1 (a) *In each part of Theorems 1 and 2, the unanimity rule is the only aggregation rule satisfying the specified conditions and anonymity.*
(b) *In each part of Theorems 1 and 2, dictatorships are the only aggregation rules satisfying the specified conditions and completeness.*

Note that none of the characterizations of oligarchic, dictatorial or unanimity rules uses a collective consistency condition: consistency follows from the other conditions, as is seen from the consistency of oligarchic, dictatorial or unanimity rules.

As mentioned in the introduction, our results are related to (and strengthen) Gärdenfors's (2006) oligarchy results. We discuss the exact relationship in Section 6, when we relax the requirement of completeness not only at the collective level but also at the individual one.

Part (b) of Corollary 1 is also related to the characterizations of dictatorships by Nehring and Puppe (2002), Dokow and Holzman (2005) and Dietrich and List (forthcoming-a). To be precise, the dictatorship corollaries derived from parts (a) of Theorems 1 and 2 are variants (without a collective consistency condition) of Dokow and Holzman's (2005) and Dietrich and List's (forthcoming-a) characterizations of dictatorships.[8] The dictatorship corollaries derived from parts (b) of Theorems 1 and 2 are variants (again without a collective consistency condition) of Nehring and Puppe's (2002) characterizations of dictatorships.

As announced in the introduction, we seek to characterize impossibility agendas. While Theorems 1 and 2 establish the sufficiency of our agenda conditions for the present oligarchy results, we also need to establish their necessity. This is done by the next result. The proof consists in the construction of appropriate non-oligarchic counterexamples, given in the Appendix.[9]

Theorem 3 *Suppose $n \geq 3$ (and X contains at least one contingent proposition).*
(a) *If the agenda condition (β) is violated, there is a non-oligarchic (in fact, non-monotonic) aggregation rule that is universal, deductively closed, unanimity-respecting and systematic.*

[8] Our agenda conditions are, in the general case, at least as strong as those of the mentioned other dictatorship characterizations; but they are equivalent to them if X is finite or belongs to a compact logic (because then (β) reduces to a standard condition; see Section 3).

[9] Part (c) still holds for $n = 2$. It also follows from a rule specified by Nehring and Puppe (2002); our proof uses a simpler (and non-complete) rule.

(b) *If the agenda condition (α) is violated, there is a non-oligarchic aggregation rule that is universal, deductively closed, unanimity-respecting, systematic and monotonic.*

(c) *If the agenda is not path-connected, and is finite or belongs to a compact logic, there is a non-oligarchic (in fact, non-systematic) aggregation rule that is universal, deductively closed, unanimity-respecting, independent and monotonic.*

4 Application I: preference aggregation

We apply Theorem 2 to the aggregation of (strict) preferences, specifically to the case where a profile of fully rational individual preference orderings is to be aggregated into a possibly partial collective preference ordering.

To represent this aggregation problem in the judgment aggregation model, consider the *preference agenda* (Dietrich and List forthcoming-a; see also List and Pettit 2004), defined as $X = \{xPy, \neg xPy \in \mathbf{L} : x, y \in K \text{ with } x \neq y\}$, where

(i) $\mathbf{L}$ is a simple predicate logic, with
- a two-place predicate P (representing strict preference), and
- a set of constants $K = \{x, y, z, ...\}$ (representing alternatives);

(ii) for each $S \subseteq \mathbf{L}$ and each $p \in \mathbf{L}$, $S \vDash p$ if and only if $S \cup Z$ entails p in the standard sense of predicate logic, with Z defined as the set of rationality conditions on strict preferences.[10]

We claim that strict preference orderings can be formally represented as judgments on the preference agenda. Call a binary preference relation $\succ$ on K a *strict partial ordering* if it is asymmetric and transitive, and call $\succ$ a *strict ordering* if it is in addition connected. Notice that (i) the mapping that assigns to each strict partial ordering $\succ$ the judgment set $A = \{xPy, \neg yPx \in X : x \succ_i y\} \subseteq X$ is a bijection between the set of all strict partial orderings and the set of all consistent and deductively closed (but not necessarily complete) judgment sets; and (ii) the restriction of this mapping to strict orderings is a bijection between the set of all strict orderings and the set of all consistent and complete (hence deductively closed) judgment sets.

To apply Theorem 2, we observe that the preference agenda for three or more alternatives satisfies the agenda conditions of Theorem 2.

Lemma 1 *If $|K| \geq 3$, the preference agenda satisfies path-connectedness and (β).*[11]

[10] Z contains $(\forall v_1)(\forall v_2)(v_1 P v_2 \rightarrow \neg v_2 P v_1)$ (asymmetry), $(\forall v_1)(\forall v_2)(\forall v_3)((v_1 P v_2 \wedge v_2 P v_3) \rightarrow v_1 P v_3)$ (transitivity), $(\forall v_1)(\forall v_2)(\neg\, v_1 = v_2 \rightarrow (v_1 P v_2 \vee v_2 P v_1))$ (connectedness) and, for each pair of distinct contants $x, y \in K$, $\neg\, x = y$.

[11] Nehring (2003) has proved the path-connectedness result for the (weak) preference agenda.

Corollary 2 *For a preference agenda with $|K| \geq 3$, the oligarchies are the only universal, deductively closed (and also consistent), unanimity-respecting and independent aggregation rules.*

We have bracketed consistency since the result does not need the condition, although the interpretation offered above assumes it. In the terminology of preference aggregation, Corollary 2 shows that the oligarchies are the only preference aggregation rules with universal domain (of strict orderings) generating strict partial orderings and satisfying the weak Pareto principle and independence of irrelevant alternatives. Here an *oligarchy* is a preference aggregation rule such that, for each profile of strict orderings $(\succ_1, ..., \succ_n)$, the collective strict partial ordering $\succ$ is defined as follows: for any alternatives $x, y \in K$, $x \succ y$ if and only if $x \succ_i y$ for all $i \in M$, where $M \subseteq N$ is an antecedently fixed non-empty set of *oligarchs*.

This corollary is closely related to Gibbard's (1969) classic result showing that, if the requirement of transitive social orderings in Arrow's framework is weakened to that of quasi-transitive ones (requiring transitivity only for the strong component of the social ordering, but not for the indifference component), then oligarchies (suitably defined for the case of weak preference orderings) are the only preference aggregation rules satisfying the remaining conditions of Arrow's theorem. The relationship to our result lies in the fact that the strong component of a quasi-transitive social ordering is a strict partial ordering, as defined above.

5 Application II: group identification

Here we apply Theorem 2 to Kasher and Rubinstein's (1997) problem of "group identification". A set $N = \{1, 2, ..., n\}$ of individuals (e.g., a population) each make a judgment $J_i \subseteq N$ on which individuals in that set belong to a particular social group, subject to the constraint that at least one individual belongs to the group but not all individuals do (formally, each J_i satisfies $\varnothing \subsetneq J_i \subsetneq N$). The individuals then seek to aggregate their judgments $(J_1, ..., J_n)$ on who belongs to the social group into a resulting collective judgment J, subject to the same constraint $(\varnothing \subsetneq J \subsetneq N)$. Thus Kasher and Rubinstein analyse the case in which the group membership status of all individuals must be settled definitively.

By contrast, we apply Theorem 2 to the case in which the membership status of individuals can be left undecided: i.e., some individuals are deemed members of the group in question, others are deemed non-members, and still others are left undecided with regard to group membership, subject to the very minimal "deductive closure" constraint that if all individuals except one are deemed non-members, then the remaining individual must be deemed a member, and if all individuals except one are deemed members, then the remaining individual must be deemed a non-member.

To represent this problem in our model (drawing on a construction in List 2006), consider the *group identification agenda*, defined as $X = \{a_1, \neg a_1, ..., a_n, \neg a_n\}$, where

(i) **L** is a simple propositional logic, with atomic propositions a_1, ..., a_n and the standard connectives $\neg$, $\wedge$, $\vee$;

(ii) for each $S \subseteq \mathbf{L}$ and each $p \in \mathbf{L}$, $S \vDash p$ if and only if $S \cup Z$ entails p in the the standard sense of propositional logic, where $Z = \{a_1 \vee ... \vee a_n, \neg(a_1 \wedge ... \wedge a_n)\}$.

Informally, a_j is the proposition that "individual j is a member of the social group", and $S \models p$ means that S implies p relative to the constraint that the disjunction of a_1, ..., a_n is true and their conjunction false. The mapping that assigns to each J (with $\varnothing \subsetneq J \subsetneq N$) the judgment set $A = \{a_j : j \in J\} \cup \{\neg a_j : j \notin J\} \subseteq X$ is a bijection between the set of all fully rational judgments in the Kasher and Rubinstein sense and the set of all consistent and complete judgment sets in our model. A merely deductively closed judgment set $A \subseteq X$ represents a judgment that possibly leaves the membership status of some individuals undecided, as outlined above and illustrated more precisely below.

To apply Theorem 2, we observe that the group identification agenda for three or more individuals ($n \geq 3$) satisfies the agenda conditions of Theorem 2.

Lemma 2 *If $n \geq 3$, the group identification agenda satisfies path-connectedness and (β).*

Corollary 3 *For a group identification agenda with $n \geq 3$, the oligarchies are the only universal, deductively closed (and consistent), unanimity-respecting and independent aggregation rules.*

In group identification terms, the oligarchies are the only group identification rules with universal domain generating possibly incomplete but deductively closed group membership judgments and satisfying unanimity and independence. Here an *oligarchy* is a group identification rule such that, for each profile $(J_1, ..., J_n)$ of fully rational individual judgments on group membership, the collective judgment is given as follows: the set of determinate group members is $\bigcap_{i \in M} J_i$, the set of determinate non-members is $\bigcap_{i \in M} (N \backslash J_i)$, and the set of individuals with undecided membership status is the complement of these two sets in N, where $M \subseteq N$ is an antecedently fixed non-empty set of *oligarchs*.[12]

[12] In fact, the set of individuals whose group membership status is to be decided need not coincide with the set of individuals who submit judgments on who is a member. More generally, the set N can make judgments on which individuals in another set K ($|K| \geq 3$) belong to a particular social group, subject to the constraint stated above. K can be infinite. Corollary 3 continues to hold since the corresponding group identification agenda (for a suitably adapted logic) still satisfies path-connectedness and (β). Interestingly, if K is infinite the agenda belongs to a non-compact logic.

6 The case of incomplete individual judgments

As argued by Gärdenfors (2006), it is natural to relax the requirement of completeness not only at the collective level, but also at the individual one. Do the above impossibilities disappear if individuals can withhold judgments on some or even all pairs $p, \neg p \in X$? Unfortunately, the answer to this question is negative, even if the conditions of independence or systematicity are weakened by allowing the collective judgment on a proposition $p \in X$ to depend not only on the individuals' judgments on p but also on those on $\neg p$. Such weaker independence or systematicity conditions are arguably more defensible than the standard conditions: $\neg p$ is intimately related to p, and thus individual judgments on $\neg p$ should be allowed to matter for group judgments on p. As the weakened conditions are equivalent to the standard ones under individual completeness, all the results in Section 3 continue to hold for the weakened independence and systematicity conditions.

Formally, let $\mathcal{C}^*$ be the set of all consistent and deductively closed (but not necessarily complete) judgment sets $A \subseteq X$, and call F *universal** if F has domain $(\mathcal{C}^*)^n$ (a superdomain of $\mathcal{C}^n$). An *oligarchy** is the universal* variant of an oligarchy as defined above.

Following Gärdenfors (2006), call F *weakly independent* if, for any $p \in X$ and any $(A_1, ..., A_n), (A_1^*, ..., A_n^*) \in Domain(F)$, if [for all $i \in N$, $p \in A_i \Leftrightarrow p \in A_i^*$ and $\neg p \in A_i \Leftrightarrow \neg p \in A_i^*$] then $p \in F(A_1, \ldots, A_n) \Leftrightarrow p \in F(A_1^*, \ldots, A_n^*)$. Likewise, call F *weakly systematic* if, for any $p, q \in X$ and any $(A_1, ..., A_n), (A_1^*, ..., A_n^*) \in Domain(F)$, if [for all $i \in N$, $p \in A_i \Leftrightarrow q \in A_i^*$ and $\neg p \in A_i \Leftrightarrow \neg q \in A_i^*$] then $p \in F(A_1, \ldots, A_n) \Leftrightarrow q \in F(A_1^*, \ldots, A_n^*)$.

We now give analogues of parts (a) of Theorems 1 and 2, proved in the Appendix.

Theorem 1* *Let the agenda X satisfy (α) and (β). The oligarchies* are the only universal*, deductively closed, unanimity-respecting and weakly systematic aggregation rules.*

Theorem 2* *Let the agenda X satisfy path-connectedness and (β). The oligarchies* are the only universal*, deductively closed, unanimity-respecting and weakly independent aggregation rules.*

In analogy with Theorems 1 and 2, these characterizations of oligarchies* do not contain a collective consistency condition (but require individual consistency). In each of Theorems 1* and 2*, adding the collective completeness requirement (respectively, anonymity) narrows down the class of aggregation rules to dictatorial ones (respectively, the unanimity rule), extended to the domain $(\mathcal{C}^*)^n$. So Theorems 1* and 2* imply characterizations of the latter rules on the domain $(\mathcal{C}^*)^n$. Note, further, that our applications of Theorem 2 to the

preference and group identification agendas in Sections 4 and 5 can accommodate the case of incomplete individual judgments by using Theorem 2* instead of Theorem 2.

We can finally revisit the relationship of our results with Gärdenfors's results. Theorem 2, Corollary 1 and Theorem 2* strengthen Gärdenfors's oligarchy results. First, they do not require Gärdenfors's "social consistency" condition.[13] Second, they show that the conditions on aggregation rules imply (and in fact fully characterize) strong and not merely weak oligarchies (respectively, oligarchies*). Third, they weaken Gärdenfors's assumption that the agenda has the structure of an atomless Boolean algebra, replacing it with the weakest possible agenda assumption under which the oligarchy result holds, i.e., path-connectedness and (β).

Our results show that allowing incomplete judgments while preserving deductive closure and (weak) independence does not lead very far into possibility terrain. To obtain genuine possibilities, deductive closure must be relaxed or – perhaps better – independence must be given up in favour of non-propositionwise aggregation rules.

7 References

Alchourron, C. E., Gärdenfors, P., Makinson, D. (1985) On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic* 50: 510-530

Dietrich, F. (2006) Judgment Aggregation: (Im)Possibility Theorems. *Journal of Economic Theory* 126(1): 286-298

Dietrich, F. (forthcoming) A generalised model of judgment aggregation. *Social Choice and Welfare*

Dietrich, F., List, C. (forthcoming-a) Arrow's theorem in judgment aggregation. *Social Choice and Welfare*

Dietrich, F., List, C. (forthcoming-b) Judgment aggregation by quota rules. *Journal of Theoretical Politics*

Dietrich, F., List, C. (2006) Judgment aggregation without full rationality, Working paper, Maastricht Universiy

Dokow, E., Holzman, R. (2005) Aggregation of binary evaluations, Working paper, Technion Israel Institute of Technology

Gärdenfors, P. (2006) An Arrow-like theorem for voting with logical consequences. *Economics and Philosophy* 22(2): 181-190

Kasher, A., Rubinstein, A. (1997) On the Question "Who is a J?": A Social Choice Approach. *Logique et Analyse* 160: 385-395

[13] Gärdenfors's "social logical closure" is equivalent to our "deductive closure", where *entailment* in Gärdenfors' Boolean algebra agenda X should be defined as follows: a set $A \subseteq X$ *entails* $p \in X$ if and only if $(\wedge_{q \in A_0} q) \wedge \neg p$ is the contradiction for some finite $A_0 \subseteq A$.

Kornhauser, L. A., Sager, L. G. (1986) Unpacking the Court. *Yale Law Journal* 96(1): 82-117

List, C. (2004) A Model of Path-Dependence in Decisions over Multiple Propositions. *American Political Science Review* 98(3): 495-513

List, C. (2006) Which worlds are possible? A judgment aggregation problem. Working paper, London School of Economics

List, C., Pettit, P. (2002) Aggregating Sets of Judgments: An Impossibility Result. *Economics and Philosophy* 18: 89-110

List, C., Pettit, P. (2004) Aggregating Sets of Judgments: Two Impossibility Results Compared. *Synthese* 140(1-2): 207-235

List, C., Pettit, P. (forthcoming) Group Agency and Supervenience, Southern Journal of Philosophy

Mongin, P. (2005) Factoring out the impossibility of logical aggregation. Working paper, CNRS, Paris

Nehring, K. (2003) Arrow's theorem as a corollary. *Economics Letters* 80(3): 379-382

Nehring, K., Puppe, C. (2002) Strategy-Proof Social Choice on Single-Peaked Domains: Possibility, Impossibility and the Space Between. Working paper, University of California at Davies

Nehring, K., Puppe, C. (2005) Consistent Judgment Aggregation: A Characterization. Working paper, University of Karlsruhe

Pauly, M., van Hees, M. (2006) Logical Constraints on Judgment Aggregation. *Journal of Philosophical Logic* 35: 569-585

Pettit, P. (2001) Deliberative Democracy and the Discursive Dilemma. *Philosophical Issues* 11: 268-299

Rubinstein, A., Fishburn, P. (1986) Algebraic Aggregation Theory. *Journal of Economic Theory* 38: 63-77

van Hees, M. (forthcoming) The limits of epistemic democracy. *Social Choice and Welfare*

Wilson, R. (1975) On the Theory of Aggregation. *Journal of Economic Theory* 10: 89-99

The Probability of Sen's Liberal Paradox

Keith L. Dougherty and Julian Edward

Abstract

This paper determines the probability of a conflict between acyclicity, weak Pareto, and minimal liberalism in a relatively unrestricted domain. It seems reasonable to hypothesize that the probability of a conflict between these three properties decreases as the number of individuals increases. If this were the case, Sen's Liberal Paradox would be of greater concern in small populations, such as committees, than in large populations, such as nation states. However, we conduct several numerical computations and draw the opposite conclusion. Increasing the number of individuals or the number of decisive alternative pairs increases the probability of a conflict between acyclicity, weak Pareto, and minimal liberalism, suggesting that the paradox forming preferences are not only possible in democracies, they may be probable.

1 Introduction

Sen's Liberal Paradox (1970) shows a fundamental conflict between liberty and democracy. Although it has been widely known that majorities can tyrannize minorities (Mill [1859]2006; Hamilton et al. [1788]1961), Sen's paradox shows that a social choice function cannot simultaneously satisfy minimal liberalism and weak Pareto over an unrestricted domain. This is akin to showing a conflict between an individual's ability to determine very limited outcomes for themselves (such as whether they sleep on their belly or back, everything else equal) and unanimous decision making – two properties that cannot come into direct conflict if liberal rights are properly assigned. The conflict emerges because the preservation of these two principles can lead to a violation of acyclicity (a necessary condition for a social choice function).

Previous studies have shown that strong restrictions on the domain of preferences (Blau 1975; Farrell 1976; Sen 1979) can eliminate the paradox. However, the frequency with which acyclicity, minimal liberalism, and weak Pareto come into conflict in a relatively unrestricted domain is an open question. If the probability of a conflict is small, then the implications of the liberal paradox are limited. The conflict between these conditions exist for some preferences, but they would not be frequent enough to cause alarm for democracy. If the probability of a conflict is large, then it would be difficult to promote specific types of liberal values and the weak Pareto criterion at the same time, as Sen suggests.

This paper attempts to determine the probability that a set of individual preferences will cause a conflict between acyclicity, minimal liberalism, and

weak Pareto over a finite set of alternatives and a finite number of individuals. In this sense, it captures how likely paradox creating preferences exist within the domain of all possible preferences. This question is similar to the question asked by Niemi & Weisburg (1968), Caplin & Nalebuff (1988), and Gehrlein (2002) about the probability of voting cycles. Unlike many of these studies, this paper estimates the probability of a conflict between three properties using simulations. These simulations are based on preferences drawn from a multidimensional spatial voting model. The advantage of a multidimensional spatial voting model is that it can be used to sample a large variety of preferences where non-acyclic social rankings are expected to occur (McKelvey 1976; Schofield 1978). The advantage of simulation is that it allows for the calculation of probabilities that may be mathematically intractable.

One might hypothesize that as the number of voters, or the number of alternatives, increases, the probability of a conflict between acyclicity, weak Pareto, and minimal liberalism decreases. As such Sen's Liberal Paradox would be of greater concern in small populations, such as committees, than in large populations, such as nation states. Using numerical computations, our preliminary results suggest the opposite. The properties of Sen's Liberal Paradox are often in conflict, suggesting that his conundrum has the potential of being pervasive.

2 Sen's Theorem

Following the notation used by Sen (1970, 1979), let the binary relation xPy indicate society's strict preference for x over y; xRy indicate that society prefers x at least as much as y; and xIy indicate that society is indifferent between the two alternatives. Similar relations can be defined for individuals using the subscript i.

A weak requirement in social choice theory is that social preference relations should generate a "choice set," that is, in every set of alternatives S, a subset of the full set of alternatives X, there must be a "best" alternative. A best alternative (there may be more than one) is an alternative that is at least as preferred as all other alternatives in that subset. The function that creates such a choice set is called a social decision function. Sen (1979) notes that if a preference relation is reflexive and complete,[1] then a necessary and sufficient condition for the existence of a finite choice set is acyclicity. This condition is central to the proof of his paradox.

> **Definition 1** Acyclicity ($\bar{A}$)*: A social ordering is acyclical over* X *if and only if:*
> $\forall x_1, \ldots, x_j \in X : [\{x_1 P x_2 \ \& \ x_2 P x_3 \ \& \ \ldots \ \& \ x_{j-1} P x_j\} \rightarrow x_1 R x_j]$.

[1] A preference relations is reflexive if and only if $\forall x \in S : xRx$. A preference relation is complete if and only if $\forall x, y \in S : (x \neq y) \rightarrow (xRy \text{ or } yRx)$.

Definition 2 Unrestricted Domain (U)*: Every logically possible set of individual orderings is included in the domain of the social decision function.*

Definition 3 Weak Pareto ($\bar{P}$)*: If every individual prefers alternative x to alternative y, then society must prefer x to y.*

Definition 4 Minimal Liberalism (L^*)*: There are at least two individuals such that for each of them there is at least one pair of alternatives over which he/she is decisive, that is there is a pair $\{x, y\}$ such that if he/she prefers x (respectively y) to y (respectively x), then society should prefer x (respectively y) to y (respectively x).*[2]

The purpose of the last condition is to assure that at least two individuals are able to make one social choice, such as determining whether their walls will be pink rather than white or whether they will sleep on their belly or their back, everything else equal. With these definitions, Sen shows the following impossibility theorem.

Theorem 1 *There is no social decision function that can simultaneously satisfy U, $\bar{P}$, and L^*.*

Sen completes his proof by showing that there exists a set of preferences which cause a contradiction between $\bar{A}$, $\bar{P}$, and L^*. This is done for two individuals with 1) non-overlapping pairs, 2) overlapping pairs with one element in common, and 3) overlapping pairs with two elements in common. The proof shows that conundrum causing preferences can occur within an unrestricted domain. Since two individuals with decisive rights over one pair of alternatives each are a subset of a larger population with more decisive rights, the theorem applies to any number of individuals and any number of alternatives.

However, the theorem does not guarantee that conundrum causing preferences will occur for all elements in the domain. As Sen writes, "The dilemma posed here may appear to be somewhat disturbing. It is, of course, not necessarily disturbing for every conceivable society, since the conflict arises with only particular configurations of individual preferences" (Sen 1970, 155). To see this, consider a society with two individuals, Muddy and Billie, who can choose over three alternatives x, y, and z. Muddy prefers xP_iyP_iz and Billie prefers yP_ixP_iz. If Muddy is decisive over $\{x, y\}$ and Billie is decisive over $\{y, z\}$, then xPy and yPz by L^*. Furthermore, xPz by $\bar{P}$. These social preferences maintain $\bar{A}$.

3 The Probability of Sen's Conundrum

After noting that paradox causing preferences occur for *only some* preferences in the domain, the natural question is how likely are such preferences? Is

[2] Sen introduces the stronger condition of Liberalism which requires that "every" individual be decisive over at least on pair of alternatives. Of course, the theorem can be shown with the stronger condition, as well.

the probability of a conflict between $\bar{A}$, $\bar{P}$, and L^* affected by the size of the population and the number of alternatives available?

One reason these paradoxes might be *less* likely in larger populations than in smaller ones is that the probability of a Pareto preferred alternative decreases as the size of the population increases (Dougherty & Edward 2005). Hence, the probability of a conflict may diminish because Pareto preferred alternatives are less likely to arise. Furthermore, Blau (1975) shows that for the case of two individuals and four alternatives, only 4 of the 75^2 possible configurations of preferences would cause a conflict. He conjectures that this probability will decrease as the number of individuals increases.

One reason these paradoxes might be *more* likely in larger populations than smaller ones is that the probability of intransitivity may increase as the size of the population increases (Niemi & Weisberg 1968).[3] Hence, we are more likely to violate acyclicity in larger populations than in smaller ones.

To determine the probability of a conflict between $\bar{A}$, $\bar{P}$, and L^*, we conduct a series of probability experiments using multidimensional spatial voting models and a self-written C program. In the first two sets of experiments, we assume there are N individuals choosing among A alternatives in a multidimensional outcome space. Each individual has an ideal point I_i with Euclidean preferences. This implies that each individual prefers alternatives closer to their ideal point more than alternatives farther away. Although these assumptions do not allow for all possible combinations of preferences, they are sufficiently general to make non-acyclic social preferences likely (McKelvey 1976; Schofield 1978). Furthermore, single peaked and symmetric preferences are common in the political science literature (Poole 2005; Tsebelis 2002; Stewart 2001). Hence, they presumably model populations that researches believe occur.

3.1 Two Dimensions, Continuous Outcome Space

In our first probability experiment, the simulation proceeds as follows. For each trial, the program randomly draws N ideal points and A alternatives from a compact unit square. As our baseline study, we assume that both N and A are uniformly distributed. This helps to create something similar to Gehrlein's (2002) impartial culture condition, which makes all orderings equiprobable. We have also considered other distributions,[4] which will be more fully considered in future research. The program then determines individual preferences for each alternative based on which alternative is closest to the individual's ideal point and ascertains whether any alternative is Pareto preferred to another. If a Pareto preferred alterantive exists, the pair-wise preference is recorded in an $A \times A$ matrix of social preferences to indicate that the alternative numbered

[3] See Gehrlein (2002) for conclusions to the contrary.

[4] Drawing ideal points from two normally distributed clusters with means of $(.25, .25)$ and $(.75, .75)$ and a standard deviation of .10 produced smaller probabilities of a conflict than those in Table 1. However, the probability of a contradiction remained roughly 1.0 for $da = 20$ and $N \geq 41$. Presumably, the homogeneity of ideal points explains the smaller probability of a contradiction.

the same as the row index is preferred to the alternative numbered the same as the column index.

In the next phase of the trial, the program randomly assigns decisive rights, without replacement, to N individuals and updates the social ranking based on those rights.[5] The number of decisive alternative pairs, da, assigned to each of the N individuals is fixed as an input. If $N = 3$ and $da = 10$, then 30 pairs of alternatives are randomly assigned to the social order by one of the individual's decisive rights. In this experiment, no more than one individual is allowed to have decisive rights over the same pair of alternatives, even though Sen allowed such cases in the proof of his theorem.[6] Overlapping decisive rights are allowed. For example, Muddy and Billie cannot both be decisive over the pair $\{x, y\}$, but Muddy can be decisive over $\{x, y\}$, while Billie is decisive over $\{y, z\}$.

In the final phase of the trial, we test the strict transitivity of the social preferences and fill in preferences that can be deduced by transitivity.[7] Although testing for acyclicity directly may be more appropriate, the computational problems associated with considering triples, quadruples, quintuples, etc. for the antecedent of definition 1, makes the use of acyclicity computationally inefficient. Instead, strict transitivity is used as a rough approximation for acyclicity. This works because the probability of indifference is almost zero for a uniform distribution of alternatives in spatial models that do not allow thick indifference curves. Computations confirm this.

The program tests the strict transitivity of the social ranking and updates the social ranking as needed. If strict transitivity is violated, the trial is terminated and the contradiction is noted. If there is no contradiction, the program continues to update social preferences and tests for transitivity until their is a contradiction or it is clear that the social preference ranking does not violate transitivity. To assure that all of the deductive steps are incorporated in the test for transitivity, the evaluation is conducted once more after all transitive relationships have been updated.

Such trials are repeated to determine the relative frequency that $\bar{A}$, $\bar{P}$, and L^* are in contradiction for a specific N and A. In a large number of trials, this frequency should approximate the true probability in the population.[8].

The results of this probability experiment for $A = 50$ are presented in Table 1. As the table indicates, the probability of a contradiction between $\bar{A}$, $\bar{P}$, and L^* is large, even for small populations. This suggests that Sen's paradox

[5] Condition L^* requires at least two individuals to be decisive over at least one pair of alternatives. We assign N individuals decisive rights over at least one pair because we believe few would find solice in the more restricted set of rights permitted by L^* and because Sen considers the case of N individuals having decisive rights over at least one pair of alternatives in his condition L (Sen 1970).

[6] Of course, if individuals are allowed to have decisive rights over the same pair of alternatives, then the conundrum is even more likely than the results shown here.

[7] Strict transitivity: $\forall x_1, x_2, x_3 \in X : (x_1 P x_2 \ \& \ x_2 P x_3\ ) \rightarrow x_1 P x_3]$.

[8] With 1/2 million trials, we are 95% confident that the true probability is within 0.0015 of the relative frequencies reported. This statement is based on the standard deviation of a univariate proportion, $\sqrt{\frac{\pi(1-\pi)}{T}}$, where T is the number of trials.

Table 1: Probability of a Contradiction in Two Dimensions

	da		
N	1	10	20
3	0.015	0.676	0.947
41	0.274	1.000	1.000
60	0.703	1.000	1.000

Note: $A = 50$; hence, $\binom{A}{2} = 1,225$. Rounded figures indicate the probability of a contradiction between $\bar{A}$, $\bar{P}$, and L^*. Trials $= 500,000$.

may not be an abberation. Not only do the contradictions exist, they are fairly common in the domain of possible preferences. Furthermore, as N increases, or da increases, the probability of a contradiction increases as well. This implies that the paradox is more likely to occur in populations the size of a small town with a large number of alternatives and a great number of decisive rights than in committees with alternatives limited to say a short menu of exogenously formulated items.

Dougherty & Edward (2005) claim that the probability of a Pareto preferred alternative decreases as N increases. Diagnostics confirm this and suggest that the decrease occurs fairly rapidly. For example, there is an average of roughly 432 Pareto preferred comparisons for $N = 3$ and $A = 50$ (Recall, for $A = 50$ there are $\binom{A}{2} = 1,225$ of possible pairs of alternatives). However, there is an average of only 3.7 Pareto preferred alternatives for $N = 41$ and $A = 50$. This means that the increased probability of a conflict between $\bar{A}$, $\bar{P}$, and L^* for larger N is primarily due to a direct conflict between between $\bar{A}$, and L^*. This explains the larger N results in Table 1. For $N = 41$ and $da = 10$, 33% of the possible pairs are dictated by decisive rights $(= 410/1,225)$. For $N = 60$ and $da = 20$, this shoots up to 98% of the possible pairs dictated by such rights $(= 1,200/1,225)$. In such cases, a small amount of preference heterogeneity can lead to conflicts.

This highlights one of the fundamental issues in determining the probability of a conflict between $\bar{A}$, $\bar{P}$, and L^*. The proper ratio of decisive alternative pairs to all possible pairs appears to affect the result. However, before the reader concludes that the ratio of $Nda/\binom{A}{2}$ is the full explanation for these results, note that $A = 100$ is not sufficiently large to make the probability of a conflict between $\bar{A}$, $\bar{P}$, and L^* decrease from 1. In this case, the decisive alternative pairs represent only 8% of the total number of alternative pairs possible.[9] Finding a ratio that is both desirable and feasible may be critical to determining whether Sen's Liberal Paradox should be considered pervasive.

[9] Rounding at 10^{-6}.

In addition to more investigating the proper relationship between $N(da)$ and A, there are two natural extensions to these results worthy of further investigation. One extension is to try to reduce the structure of preferences imposed by a two dimensional spatial voting model. Instead, an attempt should be made to consider all possible cases in an almost completely unrestricted domain. The other is to restrict the domain in a way that more accurately models the type of preferences which might arise in an actual liberal choice situation. These ideas will be briefly addressed in the next two sections, respectively.

3.2 Twenty Dimensions, Continuous Outcome Space

In an attempt to reduce the structure imposed on the domain by a two dimensional spatial model, we extended the analysis to twenty dimensions. Our experiment for greater dimensions were conducted similar to the experiment described in the previous section. In each trial, the program randomly draws N ideal points and A alternatives from a uniform distribution on an n-dimensional hyper-cube. Individual $i's$ preference for each pair of alternatives is determined based on the shorter of the two distances between $i's$ ideal point and the two alternatives. Pareto comparisons, the assignment of decisive rights, and tests of transitivity are conducted as done before. The only difference is that the program keeps track of a greater number of dimensions.

Table 2: Probability of a Contradiction in Twenty Dimensions

	da		
N	1	10	20
3	0.001	0.637	0.971
41	0.236	1.000	1.000
60	x.xxx	x.xxx	1.000

Note: $A = 50$; hence, $\binom{A}{2} = 1,225$. Rounded figures indicate the probability of a contradiction between $\bar{A}$, $\bar{P}$, and L^*. Trials $= 500,000$.

Results of this experiment for 20 dimensional space appear in Table 2. Notice that the probability of a contradiction between $\bar{A}$, $\bar{P}$, and L^* for 20 dimensions is not greater than it is for 2 dimensions in almost every case. The one exception is $N = 3$ and $da = 20$. Part of the reasons why the figures in Table 2 are typically smaller than the figures reported in Table 1 is that the probability of a Pareto preferred alternative typically decreases as the number of dimensions increases. In the exceptional case of $N = 3$, $da = 20$, the probability increases with the number of dimensions. This conclusion is based on monitoring the average number of Pareto preferred alternatives per trial.

To confirm that the change in the number of Pareto preferred cases was

explaining the differences between the $2D$ and $20D$ results, we re-configured the program so that it made no Pareto comparisons. This left the program testing the probability of a conflict between $\bar{A}$ and L^*. We found that with Pareto removed, the probability of a conflict between $\bar{A}$ and L^* is roughly the same regardless of the number of dimensions (dimensions 3 through 6 were also explored). This suggests that the differences between comparable figures in Table 1 and Table 2 can be explained by the effects of $\bar{P}$ on the social preferences.[10]

Even though the probability of a conflict between $\bar{A}$, $\bar{P}$, and L^* decreases with increasing dimensions, the probabilities are still very large for fairly small populations. Again, this suggests that Sen's paradox may be probably in medium and large populations.

3.3 Decisive Dimensions, Dichotomous Alternatives

As Sen and others have pointed out, sufficient restriction on the domain of alternatives can lead to an avoidance of his paradox. In an attempt to more accurately model decisive choices that Sen may have envisioned, we now consider a very different model. In this model, individual i will be decisive over a pair of alternatives if and only if the difference between the two alternatives is a an attribute that individual i is supposed to be decisive over. An example may illustrate the point.

Imagine that there are two individuals: Muddy and Billie, each of which want to decide whether they will sleep on their belly or their back. With two attributes and two individuals, there are 2^N possible states:
$s_1 = \{$Muddy belly, Billie belly$\}$
$s_2 = \{$Muddy belly, Billie back$\}$
$s_3 = \{$Muddy back, Billie belly$\}$
$s_4 = \{$Muddy back, Billie back$\}$

Muddy is decisive over $\{s_1, s_3\}$ and $\{s_2, s_4\}$. Billie is decisive over $\{s_1, s_2\}$ and $\{s_3, s_4\}$. Pareto comparisons can be made among all the alternatives. The best way to assure that each individual has decisive rights over the alternatives that differ only in terms of their decisive attributes, is to create a decisive dimension for each individual. One dimension reflects Muddy's choice to sleep

[10] At the time of submission, we have no good explanation for why the probability of a Pareto preferred alternative decreases as the number of dimensions increases. We only note that for any alternative x *not* in the convex hull, the set of points Pareto preferred to x is *no greater than* those circumscribed by the indifference curve centered on I^*, where I^* is the ideal point closest to x. For dimensions greater than one, the set of alternatives Pareto preferred to x are often smaller. Now consider $|x - I^*| = 0.1$ fixed across dimensions. For one dimension, the area within the indifference curve for I^* is just less than 0.2, which is just less than 0.2 of the total interval. For two dimensions, the area within the indifference curve for I^* is .031 $(= \pi(.1)^2)$, which is .031 of the area of the unit square. For three dimensions, the area (or volume) within the indifference sphere for I^* is .004 $(= 4/3\pi(.1)^3)$ If the probability of $|x - I^*| = 0.1$ is roughly the same across dimensions, this would suggest that the potential for Pareto improvements would be smaller as the number of dimensions increases.

on his belly or back. The other dimension reflects Billie's choice to sleep on her belly or back. Each individual is decisive along this dimension *only if* the alternatives are identical on every other dimension. This set-up implies that the number of dimensions, and the number of alternatives, should be a function of the number of individuals, N, and the number of decisive attributes, r, assigned to each individual. For example if Muddy is decisive over sleeping on his belly or his back and whether to read *Lady Chatterly's Lover* or not, then $r = 2$. If Billie gets the same rights, then the number of dimensions is $Nr = 4$, and the appropriate number of alternatives is $2^{Nr} = 16$. For the case at hand, $r = 1$, the appropriate number of dimensions is 2, and $A = 4$.

Our third probability experiment tests this type of decision making over r dichotomous attributes and 2^{Nr} alternatives. Each alternative is assigned to a corner in the n-dimensional, unit hyper-cube. For $N = 2$ and $r = 1$, this hypercube is a unit square in two dimensional space, with four corresponding alternatives: $s_1 = (0, 0)$; $s_2 = (0, 1)$; $s_3 = (1, 0)$; and $s_4 = (1, 1)$. During each trial of the experiment, the program randomly assigns preferences to each individual, presuming that each of the $A!$ possible orders are equally likely. This is done by randomly drawing one of the alternatives as the individual's most preferred alternative without replacement. The individual's second most preferred alternative is then randomly drawn from the remaining alternatives. Care is taken to assure that this is drawn with equal probability among the remaining alternatives, again without replacement. The process continues until each individual is assigned a strict order over the A alternatives.

After the preferences are determined, the program assigns social preferences based on a Pareto, similar to the process described before. For example, if Muddy and Billie both prefer s_2 to s_3, then society $s_2 P s_3$.

For the minimal liberalism routine, the program compares all alternatives pairwise. It then determines whether $x_j = x_k$ on every dimension except the one representing individual $i's$ decisiveness. In such a case, the program assigns social preferences over that pair based on the preferences of individual i on that pair. Such an assignment of social preferences occurs only if x_j and x_k are identical on every dimension except the dimension of some individual i. The transitivity routine then works as described previously.

Due to limited time, we ran this program only for the case of $N = 3$ and $r = 1$. In this case, $A = 8$, with 3 dimensions. Computational results suggest that the probability of a contradiction between $\bar{A}$, $\bar{P}$, and L^* is 0.666. On average, 0.25 of the possible alternative pairs could be decided by Pareto.[11] Although there are a number of differences between the simulations described in this section and section 3.2, this probability can be compared to results in section 3.2 for $N = 3$, $A = 8$, $da = 4$ and 3 dimensions. In that particular case, the probability of a contradiction is 0.580, which is smaller than the probability reported here. This might loosely suggest that structuring the preferences more closely to some of the liberal examples presented by Sen may have little effect

[11] There were roughly 7 Pareto preferred alternatives per round. Hence, $7/\binom{8}{2} = .25$.

on reducing the probability of his paradox.[12]

4 Conclusion

A few lessons seem obvious. The Pareto principle does, of course, conflict with minimal liberalism and social decision making over certain sets of individual preferences. Sen's theorem shows this must be the case. Furthermore, the patterns which cause such conflicts appear to be quite common. Hence, what Sen introduced as a conundrum of potential conflict appears to be a conundrum of highly probable conflict. Although the probability of these conflicts can be limited by restricting the number of decisive alternative pairs, or the number of decisive individuals, we believe that few would take solace in such restrictions. The condition of minimal liberalism is quite weak and limiting liberal values to say one pair of alternatives seems to be a fairly strong limitation on liberty. If society cannot allow the preponderance of its members to be free to read what they like, sleep the way they prefer, and paint their walls their favorite color, irrespective of the preferences of others in the community, then it is not clear how society can be fully committed to liberal values and the Pareto criterion simultaneously. One of the conditions must go or the notion of a consistent social decision must be re-evaluated.

Future research will evaluate the probability of these conflicts for larger values of da relative to A and for different values of N. We will also consider various distributions of individual preferences and extend our analysis of the methods described in section 3.3, particularly to higher dimensions. Hopefully, such extensions will create a path for making social decisions that is consistent with liberalism, Pareto, and the possibility that a variety of preference patterns may occur.

References

Blau, J.H. 1975. "Liberal Values and Independence." *Review of Economic Studies* 42:395–402.

Caplin, A. & B. Nalebuff. 1988. "On 64%-Majority Rule." *Econometrica* 56(4):787–814.

Dougherty, K.L. & J. Edward. 2005. "Simple vs Absolute Majority Rule." University of Georgia.

Farrell, M.J. 1976. "Liberalism in the Theory of Social Choice." *Review of Economic Studies* 43:3–10.

[12] The average number of Pareto preferred alternatives per round were roughly 9.6 for the case associated with section 3.2. This suggests that the structure of decisive choices, or the structure of alternatives, explains the difference between the two results.

Gehrlein, William V. 2002. "Condorcet's Paradox and the Likelihood of its Occurrence: Different Perspectives on Balanced Preferences." *Theory and Decision* 52:171–199.

Hamilton, Alexander, Madison James & John Jay. [1788] 1961. *The Federalist Papers*. New York, Mentor.

List, Christian. 2004. "The Impossibility of a Paretian Republican? Some comments on Pettit and Sen." *Economics and Philosophy* 20:65–87.

McKelvey, Richard. 1976. "Intransitivities in mulitdimensional voting models and some implications for agenda control." *Journal of Economic Theory* 12:472–82.

Mill, John S. [1859] 2006. *On Liberty*. New York: Filiquarian Publishing, LLC.

Niemi, R.G.. & H. Weisberg. 1968. "A mathematical solution for the probability of the paradox of voting." *Behavioral Science* 13:317–23.

Poole, Keith. 2005. *Spatial Models of Parliamentary Voting*. Nwe York, Cambridge University Press.

Schofield, Norman J. 1978. "Instability of simple dynamic games." *Review of Economic Studies* 45:575–94.

Sen, Amartya K. 1970. "The Impossibility of a Paretian Liberal." *The Journal of Political Economy* 78(1):152–157.

Sen, Amartya K. 1979. *Collective Choice and Social Welfare*. New York, North-Holland.

Stewart, Charles. 2001. *Analyzing Congress*. New York, W.W. Norton.

Tsebelis, George. 2002. *Veto Players: how political institutions work*. New York: Russell Sage Foundation.

Keith L. Dougherty
Department of Political Science
University of Georgia
Athens, GA 30602, U.S.A.
Email: `dougherk@uga.edu`

Julian Edward
Department of Mathematics
Florida International University
Miami, FL 33199, U.S.A.
Email: `edwardj@fiu.edu`

The Discursive Dilemma as a Lottery Paradox

Igor Douven and Jan-Willem Romeijn[1]

Abstract

List and Pettit have stated an impossibility theorem about the aggregation of individual opinion states. Building on recent work on the lottery paradox, this paper offers a variation of that result. The present theorem places different constraints on the voting agenda and the domain of profiles, but it covers a larger class of voting rules, for which votes on separate propositions need not be independent.

The discursive dilemma concerns the question of how to determine the opinion state of a collective on the basis of the opinion states of its members. List and Pettit [2002] have stated an impossibility theorem about voting rules, that is, rules which are meant to answer the aforementioned question. Building on recent work on the lottery paradox, we show that their result persists if certain assumptions are added while the arguably most problematic condition of their theorem is relaxed. Specifically, we employ a voting agenda with richer logical structure, and focus only on certain voting profiles, but in exchange for that we can considerably weaken the requirements on the aggregation rule. Thus we arrive at a different trade-off between restrictions on the agenda and the generality of the voting rule.

We start by rehearsing the discursive dilemma and List and Pettit's impossibility theorem, then report a generalization of the lottery paradox, exhibit an important structural similarity between the discursive dilemma and the generalized version of the lottery paradox, and finally use this similarity to generalize List and Pettit's theorem. We also explain briefly how our result relates to another impossibility theorem by Pauly and van Hees [2006].

1. Consider a parliament whose members each have individual opinions on some designated set of propositions, and imagine that the parliament must come to a collective opinion on this set. To this aim the parliament may employ some voting rule, which transforms the individual opinions regarding the propositions into an opinion for the parliament as a whole. A standard rule is majority voting, but many other voting rules are possible. Now, if the members of the parliament all have consistent opinion states, one would expect that there

[1]Earlier versions of this paper were presented at a meeting of the PPM group at the University of Constance, at the 2006 conference of the Dutch–Flemish Society for Analytic Philosophy held in Amsterdam, and at the University of Kent. We are grateful to the audiences on those occasions for helpful questions and remarks. Thanks also to Franz Dietrich, Christian List, Marc Pauly, Martin van Hees, and Christopher von Bülow for lucid comments, helpful suggestions, and stimulating conversations. Contact: Igor Douven, Instituut voor Wijsbegeerte, Universiteit Leuven, email: `igor.douven@hiw.kuleuven.be`; Jan-Willem Romeijn, Psychologie, Universiteit van Amsterdam, email: `j.w.romeijn@uva.nl`.

exist voting rules that guarantee that the parliament has a consistent collective opinion state, too. However, as List and Pettit [2002] have shown, if voting rules are required to satisfy certain minimal and prima facie plausible conditions, this is not so.

To make their result precise, we first need to settle some logical and notational issues. Let the voting agenda Φ be a set containing at least two propositions that are contingent and logically independent of each other, and be closed under the relation of standard logical consequence, meaning that any proposition logically entailed by Φ is also an element of it. A valuation $v\colon \Phi \to \{0,1\}$ is said to be consistent iff there is no $\Psi \subseteq \Phi$ such that $v(\psi) = 1$, for all $\psi \in \Psi$, and Ψ entails $\bot$, the inconsistent proposition; it is said to be complete iff $v(\varphi) = 1$ or $v(\neg\varphi) = 1$ for all $\varphi \in \Phi$; and it is said to be closed under logical consequence iff for all $\Psi \subseteq \Phi$ and all $\varphi \in \Phi$, if $v(\psi) = 1$ for all $\psi \in \Psi$ and Ψ logically entails φ, then $v(\varphi) = 1$. Let V be the set of all valuations on Φ, and $V_\star$ the set of consistent and complete valuations. Note that it follows from the definitions of consistency and completeness and the closure conditions on Φ that each $v \in V_\star$ is closed under logical consequence.

Further, let $M = \{m_1, \ldots, m_n\}$ be a parliament with members m_i and $n \geqslant 2$. Each member m_i is associated with a consistent and complete valuation $v_i \in V_M$, where v_i can be thought of as the member's individual opinion state (at least with respect to Φ; we take this relativization to be implied from now on) and $V_M \subseteq V_\star$ is the set of valuations the members of M are allowed to adopt as individual opinion states.[2] Let $V_0 \subseteq V$ be the set of allowed collective valuations; note that these valuations are not by definition consistent or complete. Finally, a voting rule for the parliament is defined to be a function $r\colon (V_M)^n \to V_0$. Recall that the valuations v_i with $i \geqslant 0$ are themselves functions over a set of propositions, $v_i\colon \Phi \to \{0,1\}$. Thus, a voting rule can be decomposed into—possibly partial—functions r_φ for all propositions $\varphi \in \Phi$ separately, according to $r_\varphi(v_1, \ldots, v_n) = \big(r(v_1, \ldots, v_n)\big)(\varphi)$ for all $\langle v_1, \ldots, v_n\rangle \in (V_M)^n$. Note also that, since a voting rule is a function, rules that render the collective opinion empty do not qualify.

With these preliminaries in place, we can state List and Pettit's [2002] impossibility result, as follows:

Proposition 1. *There is no voting rule that satisfies all of the following requirements:*

- Universal Domain. *Members of the parliament are allowed to adopt* any *consistent and complete valuation of* Φ *as their individual opinion state, that is,* $V_M = V_\star$.
- Consistent and Complete Range. *The range of the voting rule* r *is restricted to the set of* consistent *and* complete *valuations, that is,* $V_0 = V_\star$.

[2] We throughout speak of parliaments. However, this is no more than a stylistic choice. Everything to be said about parliaments applies equally well to any other kind of voting body whose members have complete, consistent, and deductively closed individual opinion states.

- Anonymity. *All members of the parliament have an equal say in the collective opinion, that is, for any permutation* $u\colon M \to M$ *of members we have* $r(v_1,\ldots,v_n) = r\big(u(v_1),\ldots,u(v_n)\big)$.

- Neutrality. *All propositions on the agenda are voted for in the same way, that is, for any permutation* $f\colon \Phi \to \Phi$ *of propositions and any pair of* n*-tuples of valuations* $\langle v_1,\ldots,v_n\rangle$ *and* $\langle v'_1,\ldots,v'_n\rangle$*, if for all* $\varphi \in \Phi$ *and all* $i \in \{1,\ldots,n\}$ *we have* $v_i(\varphi) = v'_i(f(\varphi))$*, then* $r_\varphi = r_{f(\varphi)}$.

- Independence. *The collective opinion on a proposition is a function strictly of the individual opinions on it, that is, for all* $\varphi \in \Phi$*, if* $v_i(\varphi) = v'_i(\varphi)$ *for all* $i \in \{1,\ldots,n\}$*, then* $r_\varphi(v_1,\ldots,v_n) = r_\varphi(v'_1,\ldots,v'_n)$.

List and Pettit [2002] specify the last two conditions as a conjunction under one label, *Systematicity*, but following Pauly and van Hees [2006] we have stated the conjuncts separately; this facilitates a comparison of Proposition 1 with our result to be presented later.[3]

Pauly and van Hees generalize Proposition 1 partly in ways other than we intend to pursue. One of their generalizations is that they allow valuations which can take on more than two values, so that members can for example abstain from voting. A further generalization is that they weaken Anonymity. They replace this condition with *Responsiveness* and *Non-Dictatorship*. Responsiveness says that, for at least two propositions, the collective opinion on them is not the same given any possible collection of individual opinion states, that is, there exist distinct propositions φ and ψ such that $r_\varphi(v_1,\ldots,v_n) \neq r_\varphi(v'_1,\ldots,v'_n)$ and $r_\psi(v_1,\ldots,v_n) \neq r_\psi(v'_1,\ldots,v'_n)$, for some $\langle v_1,\ldots,v_n\rangle, \langle v'_1,\ldots,v'_n\rangle \in (V_M)^n$. Non-Dictatorship says that the parliament must not be a dictatorship, meaning that the collective opinion state must not, as a rule, coincide with the opinion state of some designated individual. In itself, Non-Dictatorship is an elaboration and not a real weakening of the conditions of List and Pettit. To see this, consider the condition of *Unanimity*, which a voting rule is said to satisfy iff it includes in the collective opinion state only propositions on which the votes are unanimous. List and Pettit rule out Unanimity because it does not ensure the completeness of the collective opinion. But note that under the assumption of Anonymity, Dictatorship comes down to assuming Unanimity. So for List and Pettit, ruling out Unanimity automatically rules out Dictatorship.

In this paper, we focus on List and Pettit's condition of Systematicity. List and Pettit [2002:99] seem right that the other conditions mentioned in Proposition 1 are hardly contestable, but that Systematicity may be more controversial. In section 4 of their paper, they do consider the possibility of relaxing Systematicity, more in particular the component of Neutrality, which requires that for all propositions, inclusion (or otherwise) in the collective opinion state depends on the individual opinions in the same way. They argue that Neutrality is a

[3]Dietrich and List [2006a] show that similar results may be derived for an incomplete range, and thus for a weaker agenda than in the above.

plausible assumption, and that there is no obvious way to relax it.[4] However, they do not address the other component of Systematicity, namely Independence, according to which inclusion of a proposition in the collective opinion state should depend exclusively on the individual opinions on *that* proposition. And in our view this is an unreasonably strong requirement. Imagine a voting rule that accepts a proposition in the collective opinion state if a majority agrees with it, *provided* there do not exist majorities for other propositions that jointly undermine the former proposition, where "undermine" could be cashed out in various ways, for instance in terms of forming a coherent set of propositions on their own, but an incoherent one when conjoined with the proposition voted on.[5] While that rule may prove to be untenable on close scrutiny, one certainly would not want to reject it offhand.

However, the prospects for saying anything informative about voting rules might seem bleak once Independence is dropped. For surely there are indefinitely many ways already to amend the proviso of the previous example; and of course a voting rule need not even make majority agreement a requirement for acceptance. Nevertheless, a remarkably general result concerning voting rules can be obtained that also applies to ones that violate Independence, and it can be obtained almost for free. For it follows immediately from a recent result concerning the lottery paradox, once we have exhibited the structural similarity between that paradox and the discursive dilemma.[6] What the result shows is that a voting rule may let the collective verdict depend on the opinions on as many propositions as one likes, and in ways as complex as one likes; as long as this dependence is definable in formal terms (in a sense to be made precise), there still is no guarantee that application of the rule to consistent individual opinion states results in a consistent collective opinion state.

2. It has seemed plausible to many that high probability is sufficient for rational acceptability. However, Kyburg's [1961] so-called lottery paradox shows that, its plausibility notwithstanding, this idea cannot be maintained, at least not if we also want to maintain that rational acceptability is closed under conjunction (meaning that if two propositions are rationally acceptable then so is their conjunction). The argument goes as follows: Suppose you own a ticket in a large and fair lottery with exactly one winner. Then although it is highly unlikely that your ticket is the winner, this cannot make it rational to accept

[4] In their [2006], Dietrich and List considerably weaken Neutrality to the condition of Unbiasedness, which is the requirement that only the voting rules for a proposition and its negation must be identical.

[5] And where in turn the notion of coherence could be understood along the lines of one of the probabilistic theories of coherence that have been proposed of late.

[6] Incidentally, Bovens [2006] points to a structural similarity between the discursive dilemma and the mixed-motivation problem, which (roughly) is the problem that a group of voters one part of which is motivated by self-interest and the other part by conduciveness to the common good may come to take decisions that are neither in the self-interest of a majority of voters nor regarded as being conducive to the common good by a majority. We have not investigated the question whether any interesting lessons concerning the latter follow from the work on the lottery paradox we make use of in the present paper.

that your ticket won't win. If it did, then by the same token it should be rational to accept of each of the other tickets that they won't win, for all tickets have the same high probability of losing. And by conjunctive closure that would make it rational to accept that no ticket will win, contradicting our knowledge that the lottery has a winner.

In response to this, some philosophers have proposed to abandon the idea that rational acceptability is closed under conjunction. Arguably, however, this proposal has some quite unpalatable consequences (see Douven [2002, Sect. 2] for an overview; see also Douven and Williamson [2006]). On a more popular type of proposal, high probability *defeasibly* warrants rational acceptance, meaning that a proposition is rationally acceptable if it is highly probable, *unless* it satisfies some defeating condition D (see, for example, Pollock [1990] and Douven [2002]). However, so far attempts to specify a satisfactory defeater have been unsuccessful, and recently a result by Douven and Williamson [2006] showed that what prima facie had seemed the most attractive type of conditions—namely, those that are definable in formal terms—are unavailing, because they would trivialize the proposal.[7]

The following makes this precise. Let W be a set of worlds, and think of propositions as subsets of W. Further assume a probability distribution Pr on $\wp(W)$. Then a function f is said to be an *automorphism* of $\langle W, \wp(W), \Pr\rangle$ iff f is a $1:1$ function from $\wp(W)$ onto itself that satisfies these conditions:

1. $f(\varphi \wedge \psi) = f(\varphi) \wedge f(\psi)$,
2. $f(\neg\varphi) = \neg f(\varphi)$,
3. $\Pr(\varphi) = \Pr\big(f(\varphi)\big)$,

for all propositions $\varphi, \psi \in \wp(W)$. A *structural property* of propositions is any property P such that for any proposition φ and any automorphism f of propositions, φ has P iff $f(\varphi)$ has P. This definition can be extended to cover relations in the obvious way. A predicate is structural iff it denotes either a structural property or a structural relation. An *aggregative property* of propositions is any property such that whenever two propositions have it, their conjunction has it

[7] Another response to the lottery paradox, made by Harman [1986:71], is that if we always conditionalize our probabilities after accepting a proposition to the effect that a given ticket will lose, no contradiction will arise. For by repeating such conditionalization for "enough" tickets, we will come to the point were it will no longer be rational to accept of any of the remaining tickets that it will lose (because conditional on what we already accept, it will no longer be highly probable for any of the remaining ones that it will lose). A similar proposal in the case of the discursive dilemma would be this: vote sequentially on the propositions on the agenda, and include a proposition in the collective opinion state only if it is consistent with the deductive closure of the propositions that have already been accepted in the collective opinion state at that stage. However, Harman's proposal has been criticized for making what it is rational to accept dependent on the order in which we accept propositions (cf. Nelkin [2000], but also Douven [2007] for another view on the matter); it is obvious that a parallel critique would apply to the suggestion of sequential voting. One could try to prioritize the propositions on the agenda in some way, aiming thereby to avoid the arbitrariness, but, as List and Pettit [2002:104 f] point out, that strategy is hopeless.

too. Call a probability distribution Pr on a set W of worlds *equiprobable* iff $\Pr(\{w\}) = \Pr(\{w'\})$ for all $w, w' \in W$. Finally, a proposition φ is defined to be inconsistent iff $\varphi = \emptyset = \bot$.

Then Douven and Williamson prove the following:

Proposition 2. *Let W be finite and let* Pr *be an equiprobable distribution on $\wp(W)$. Further, let P be structural, Q aggregative, and P sufficient for Q. Then if some proposition φ such that $\Pr(\varphi) < 1$ has P, then $\bot$ has Q.*

It may be useful briefly to sketch the proof. Assume there is some proposition φ that has the property P and such that $\Pr(\varphi) < 1$. Because of the latter fact and the fact that Pr is equiprobable, there must be some $w \in W$ such that $w \notin \varphi$. Then consider all permutations on W that map some world in φ onto w and all other worlds onto themselves; it is easy to show that each such permutation defines an automorphism of propositions. So, since φ has P and P is structural, each image of φ under any of the thus-defined automorphisms has P, too, and since P is sufficient for Q, the proposition φ and its said images all have Q. Because of how the permutations were defined, there is no one world that is an element of all of these propositions, so their conjunction is inconsistent. But since Q is aggregative, that conjunction has Q. So the inconsistent proposition has Q.

As Douven and Williamson point out, this means that if rational acceptability is to be closed under conjunction, and thus an aggregative property, then if there is a sufficient condition for rational acceptability that is structural as well as non-trivial—in the sense that some proposition with probability less than 1 has it—then the inconsistent proposition is rationally acceptable: just let Q be the property of being rationally acceptable and P some structural and non-trivial condition sufficient for rational acceptability.

To appreciate the generality of this result, it suffices to check that what can reasonably be regarded as the primitive predicates from (meta-)logic, set theory and probability theory (and more generally measure theory) all define structural properties or relations. Proposition 2.3 of Douven and Williamson [2006] then does the rest, for it says that any predicate defined strictly in terms of structural predicates by means of the Boolean operators and quantification (of any order) is itself structural.

A last thing that merits remark before we return to the discursive dilemma is that the above result crucially hinges on the fact that the model that is assumed is a *finite* probability space. But surely there are infinitely many propositions expressible in our language, and thus also infinitely many propositions that might be (or fail to be) rationally acceptable. Douven and Williamson [2006, Sect. 5] offer various responses to this objection, but for present concerns the most relevant one is that we need not think of the worlds in W as being maximally specific. We can simply assume that W is a set of mutually exclusive and jointly exhaustive worlds that determine answers to all the questions that are relevant in some given context; the subsets of W then represent the contextually relevant propositions.

3. We are aiming to derive a generalization of List and Pettit's impossibility theorem (Proposition 1) from the above result concerning the lottery paradox. The strategy for doing this builds on the idea that possible worlds may be thought of as voters. In the present section we construct a particular parliament with a specific function defined on it, and an agenda, and show how together these yield a model that is isomorphic to the one assumed in Proposition 2; that suffices to make Proposition 2 apply to our construction. The next section then shows that Proposition 2 can be interpreted as offering an impossibility result more general than that of List and Pettit.

Let $W = \{w_1, \ldots, w_n\}$ be a set of mutually exclusive and jointly exhaustive worlds and let Pr be an equiprobable distribution defined on $\wp(W)$. Furthermore, let $M_W = \{m_1, \ldots, m_n\}$ be a specific parliament, where the opinion states of the members of this parliament are defined as follows. For all $\varphi \in \wp(W)$ and $i \in \{1, \ldots, n\}$, $v_i(\varphi) = 1$ iff $w_i \in \varphi$. Note that it follows automatically that each individual opinion state is complete, consistent, and deductively closed. Let the parliament's agenda consist of the elements of $\wp(W)$. It is obvious that this set is deductively closed too. Finally, define a function $g\colon \wp(M_W) \to [0,1]$ as follows: $g(M') = |M'|/n$, for all $M' \in \wp(M_W)$. We may think of g as measuring the weight a subset of M_W has in determining the collective opinion state, but the interpretation of g need not be pinned down. It is simply intended to provide us with a formal equivalent of the equiprobable distribution Pr.

To prove that $\langle W, \wp(W), \Pr\rangle$ and $\langle M_W, \wp(M_W), g\rangle$ are isomorphic structures, it suffices to show, first, that there is a bijection h from W to M_W, and second, that $\Pr(\{w \mid w \in \varphi\}) = g\big(\{h(w) \mid w \in \varphi\}\big)$ for all $\varphi \in \wp(W)$.[8] For the bijection, simply define $h(w_i) = m_i$ for all $i \in \{1, \ldots, n\}$. As to the second, note that since W is finite and Pr equiprobable, $\Pr(\varphi) = |\varphi| \,/\, |W|$ for all φ. We thus have for all φ, $\Pr(\{w \mid w \in \varphi\}) = |\{w \mid w \in \varphi\}|/n = |\{h(w) \mid w \in \varphi\}|/n = g\big(\{h(w) \mid w \in \varphi\}\big)$.

As a result, Proposition 2 applies not only to $\langle W, \wp(W), \Pr\rangle$ but, properly interpreted, to $\langle M_W, \wp(M_W), g\rangle$ as well. To be maximally clear about what it says about the latter, it may be helpful to say a few words about what the crucial terms occurring in Proposition 2 come to when they are interpreted in $\langle M_W, \wp(M_W), g\rangle$ (insofar as this is not completely evident).

Firstly, the term "proposition" now refers to elements of $\wp(M_W)$ instead of $\wp(W)$. But note that the above-defined bijection h yields a second bijection $h'\colon \wp(W) \to \wp(M_W)$ in the following obvious way: $h'(\varphi) = \{h(w) \mid w \in \varphi\}$, for all φ. Therefore, each proposition φ can be taken to be represented by the set of φ-voters in M_W as much as it can be taken to be represented by the set of φ-worlds in W. As suggested earlier, for the purposes of Douven and Williamson's paper the possible worlds may as well *be* the members of M_W as defined above. The set of propositions $\wp(M_W)$, or any subset of it that allows us to uniquely

[8] To state the following in a formally entirely precise fashion, one would have to make explicit that both our models also contain the rational interval $[0,1] \cap \mathbb{Q}$, being the range of Pr and g, respectively. But that would only make the proof more cumbersome to read while not adding anything that is not obvious anyway.

identify members of the parliament by their opinions on propositions in that subset, serves as the semantic equivalent of the voting agenda Φ referred to earlier. It will further be obvious that the voting agenda has the same logical properties whether we think of propositions as members of $\wp(W)$ or as members of $\wp(M_W)$.[9]

Secondly, when interpreted in $\langle M_W, \wp(M_W), g\rangle$ the term "Pr" is to be taken as referring to the function g, of course. From the isomorphism between the two models it follows that, formally speaking, g is a probability function on $\wp(M_W)$. Since, patently, $|\{m_i\}|/n = |\{m_j\}|/n$ for all $i, j \in \{1, \ldots, n\}$, it is an equiprobable one. Note that, again in virtue of the correspondence between sets of worlds and sets of voters in the models, the function g can be thought of as measuring the fraction of the parliament that supports a given proposition. The function g may play a part in, or even fully determine, the voting rule, as is the case in majority voting. And if g completely determines the voting rule, the fact that it is equiprobable means, in the terminology of List and Pettit, that g assumes anonymity of the members of the parliament. Furthermore, whatever its precise role in the voting rule, the fact that $g(\{m_i \mid v_i(\varphi) = 1\}) < 1$ can be interpreted as meaning that φ is not unanimously supported by the parliament. This latter fact is central to the result to be presented in the next section.

Lastly, recall that being structural is defined as invariance under automorphisms *of a given model*. Hence a property or relation (and, correspondingly, a predicate denoting that property or relation) which is structural with respect to one model need not be so with respect to another. However, again from the isomorphism between $\langle W, \wp(W), \Pr\rangle$ and $\langle M_W, \wp(M_W), g\rangle$ it follows that all properties and relations that are structural relative to the former are also structural relative to the latter.

4. We now come to our main result: the specific parliament constructed in the foregoing will be used in an impossibility theorem.

We first need to link the rational acceptability of a proposition with its equivalent in the discursive dilemma, namely, the inclusion of a proposition in the collective opinion state. Let us say that a proposition φ satisfies the property R iff $r_\varphi(v_1, \ldots, v_n) = 1$. So, having property R is a sufficient condition for a proposition to end up being accepted in the collective opinion state.

We can now use this property R in a first translation of Proposition 2. Given that the parliament M_W is finite and g is the weighting function on $\wp(M_W)$, and filling in property R for P and the property of being accepted in the collective opinion state for Q, this proposition says the following about $\langle M_W, \wp(M_W), g\rangle$: if R is a structural property and the property of being in the collective opinion state is aggregative, and if some proposition $\varphi \in \wp(M)$ such

[9] Douven and Williamson's response to the objection that their result requires a finite probability space in which only finitely many propositions can be represented applies, mutatis mutandis, here as well, or even with more right: voting bodies typically do not and, realistically speaking, cannot aim to decide about all propositions expressible in our language, but only on some subset of contextually relevant ones.

that $g(\{m_i \mid v_i(\varphi) = 1\}) < 1$ satisfies R, then $\perp$ is in the collective opinion state. If we note that, by the definition of R, demands placed on this property are in effect demands placed on the corresponding voting rule r, and we call r structural iff R is a structural property, then a second, more intuitive translation of Proposition 2 is this: given the parliament M_W, if r is structural and its range includes the collective opinion states that are aggregative, then r renders the collective opinion state inconsistent, unless it only includes propositions in that state that are unanimously supported by the members of M_W. We will say that a voting rule that is structural satisfies the condition of *Structuralness*.

This translation of Proposition 2 brings us close to our impossibility theorem. Before stating this in a form similar to List and Pettit's theorem, however, it is worth noticing that the foregoing hinges on a highly specific construction, namely, a parliament M_W in which for every two members there is at least one proposition about which they disagree (so that every member can be individuated by her opinions on the agenda). Call such a parliament *opinionated*. From Douven and Williamson's result about the lottery paradox it follows that if there is a sufficient condition for rational acceptability that is structural and does not require probability 1, then the inconsistent proposition will qualify as being rationally acceptable as soon as some proposition that has probability less than 1 qualifies as such. It does *not* follow from the above result about the discursive dilemma that if a voting rule is structural and does not require unanimous support, then it will lead to inclusion of the inconsistent proposition in the collective opinion state. Whether it does will depend on whether the parliament is opinionated. However, for the impossibility theorem to be stated below it is enough that an opinionated parliament M_W is *possible*.

The fact that we are working with a fixed valuation has some consequences for how we can define the conditions of the impossibility result. For one thing, we need to consider the voting agenda and its relation to the parliament. In all impossibility results in the literature, the agenda is independent of the size and composition of the parliament. Unfortunately, this is not so in the construction of the inconsistent parliament M_W. The agenda must be such that it allows for an opinionated parliament, which provides a lower bound to the size of the agenda for a given parliament. Specifically, for a parliament of size n we need an agenda that has at least $k \geqslant \log_2 n$ logically independent propositions. And with an agenda of that size, the agenda must further contain all propositions that can be constructed with these k propositions by means of conjunction and negation operations. But on the face of it, we do not find these requirements on the size and richness of the agenda unnatural. Surely in real life it may happen that a parliament is opinionated. It seems natural to require from a voting rule that it be capable of dealing with such eventualities.

Nevertheless, it is of interest to see whether we can arrive at an inconsistent parliament with agendas in which an opinionated parliament cannot be constructed directly, either because the agenda is too small for that or because it does not have enough logical structure. It can be noted immediately that if a parliament of n members can be divided into d equally large parties, $n = 0$

mod n/d, then we may build a similar construction by taking the parties as single voters. This would require a smaller number of logically independent propositions, namely, $k \geqslant \log_2 d$. The requirement that the agenda be rich enough to make the parliament opinionated can therefore be greatly relaxed to the requirement that the agenda be rich enough to make the parliament *party-wise opinionated*, that is, divide the parliament in equally large parties each two of which disagree about at least one proposition on the agenda.

Now let us concentrate on the conditions appearing in List and Pettit's impossibility theorem: Universal Domain, Consistent and Complete Range, Anonymity, Neutrality, and Independence. Firstly, the use of a fixed valuation entails that the condition of Universal Domain can be weakened. To allow for the kind of parliament and agenda structure that is isomorphic to the model used in the generalization of the lottery paradox described in section 2, we must suppose that there are profiles in the domain of the voting functions with regard to which the parliament is party-wise opinionated. A domain that is universal in the sense of the condition of Universal Domain clearly includes such party-wise opinionated profiles, but smaller domains may also include them.

Secondly, the condition of Consistent and Complete Range may be weakened. Note first that we need not require the completeness of the collective opinion state. It can very well be that neither φ nor $\neg\varphi$ satisfies R, so that neither φ nor its negation need be an element of the collective opinion. Since the property of being accepted in the collective opinion state is only supposed to be aggregative, apart from consistency we only need to assume that whenever two propositions are both in the collective opinion, so is their conjunction. Thus, if we call the set of valuations that satisfies this condition plus consistency $V_\wedge$, then the minimal requirement is that $V_0 = V_\wedge$; call this requirement *Consistent and Aggregative Range*.

Thirdly, let us consider Anonymity. Recall that this condition requires that the voting rule be invariant under a permutation of voters, which means that it must have the same value at those profiles in the domain that only differ in the order of voters. This requirement is defined by reference to the domain V_M of the voting rule. But notice that in the construction M_W, the behavior of the voting rule only matters at the party-wise opinionated profiles in the domain. At these profiles the collective opinion is at danger of being inconsistent, and if at these profiles we allow the voting rule to give a deciding vote to some designated subset of its members, then the inconsistency can be avoided. Thus, to arrive at an impossibility result, all we need to assume is the invariance of the voting rule in the subdomain where the parliament is party-wise opinionated. That is, only at these profiles must we assume that the rule satisfies Anonymity, and thus invariance under permutations of voters.

But this restricted form of Anonymity is of limited importance in the present context. Recall that the translated version of Proposition 2 demands that the voting rule r is structural. As said, we call a voting rule r structural iff it is invariant under specific transformations of propositions, so-called automorphisms. With the further fact that in a party-wise opinionated parliament propositions

are represented by subsets of voters/parties, we can spell out automorphisms as transformations of propositions effected by a permutation of the voters/parties. So the requirement that the voting rule be structural is equivalent to the requirement that it be invariant under such permutations of voters. The question may arise whether the condition that the voting rule satisfies Anonymity is equivalent to the condition that it is structural, because both concern permutations of voters. The answer is negative. The important observation here is that the two types of permutations are not the same: it is much less to require of a voting rule that its value for a specific proposition be invariant under different labellings of the voters simpliciter, without the transformation of the proposition induced by the permutation of voters.

On the other hand, if a voting rule violates Anonymity at party-wise opinionated profiles—so that it is not invariant under different labellings of voters at these profiles—then it is also not invariant over some set of propositions that is closed under automorphisms. In such a case it may happen that some proposition φ will be accepted in the collective opinion in virtue of the fact that a specific voter or party supports it, while the proposition ψ, the image of φ under the permutation of this voter, or party of voters, with a voter that does not support φ, will not be accepted in the collective opinion. As a result, a structural voting rule automatically satisfies Anonymity at all profiles in its domain where the parliament is party-wise opinionated. We may therefore subsume the condition of Anonymity at party-wise opinionated profiles under the requirement of Structuralness.

Finally there is the condition of Neutrality. Recall that the inclusion of a proposition in the collective opinion state by a voting rule r depends on whether a proposition satisfies the corresponding property R. This property is assumed to apply to all propositions, and in this sense our result assumes Neutrality. However, the only assumption we are making about the property is that it is structural. Because of this, it is possible to incorporate any structural difference between two propositions φ and ψ in the property R. In other words, the above result is left intact under any violation of Neutrality that concerns types of propositions—in the sense that for propositions of one type one rule might be appropriate, for propositions of a second type a second rule might be appropriate, and so on—provided the types can be individuated in structural terms. We may therefore replace the condition of Neutrality with the weaker condition of Neutrality for types of propositions of the aforementioned sort, and again subsume this weaker condition under the condition of Structuralness. Equivalently, in the formulation of Neutrality in Proposition 1 we may replace "for any permutation of propositions" by "for any permutation of propositions that corresponds to an automorphism of those propositions."

With these translated conditions of Proposition 2 in place, the isomorphism that we established in the previous section effectively proves

Proposition 3. *Consider a parliament and assume an agenda and a domain of individual opinion states which allow for the possibility that the parliament is party-wise opinionated. Then for all party-wise opinionated profiles of the par-*

liament there exists no voting rule that satisfies Structuralness and Consistent and Aggregative Range, unless it also satisfies Unanimity.

In other words, we have derived that in the wider class of voting rules for which $V_0 = V_\wedge$ there are none that are structural, with the exception of rules that require Unanimity whenever the parliament is party-wise opinionated. Here the fact that the voting rule is structural entails that it satisfies Anonymity at party-wise opinionated profiles in the domain of the voting rule, and satisfies Neutrality in the weak form stated in the previous paragraph. Most notably, the problematic condition of Independence is no longer needed.

Some remarks on this are in order. First, the property of voting rules with which we avoid inconsistent collective opinions is a rather weak one: the Unanimity of voting rules need only apply at party-wise opinionated profiles in the domain. Much of the discussion on the discursive dilemma is premised on the Universal Domain assumption, while the present result is based on a construction that only involves these specific profiles in the domain of the voting rule. This sets apart the present result from many if not all other impossibility results. The reason for this is simply that the parallel between the discursive dilemma and the lottery paradox can be drawn only at those specific elements of the domain of the aggregation function. One may argue that this limits the relevance of the result for the discussion on the discursive dilemma, but we think not. It is a real life possibility that a parliament is opinionated. And it seems rather awkward to adopt a voting rule that functions normally in case at least two members vote the same, but that reverts to Unanimity once members or equal-sized parties can be identified by their opinions. Having to assume this weakened version of Unanimity is almost as bad as having to assume it over the whole domain.

It might further be said that the condition of Structuralness hardly has a natural interpretation in the context of voting rules, and thus that the above result is of limited interest at best. First, at the risk of repeating ourselves, there *is* a natural interpretation of Structuralness: A structural voting rule is a rule that is blind to the meaning, the order, or the name tags of the propositions involved, so that it is, in a sense, a completely impartial procedure. However, it may be objected that under this interpretation, Structuralness is still an esoteric condition, and that there is no natural motivation for demanding it. But surely Structuralness is not an outlandish condition at all. For one thing, the rule of majority voting, which in practice is without any doubt more common than any other rule, satisfies Structuralness. It is not hard to think of more complicated but still intuitively reasonable rules that satisfy this condition too. One may think here of rules of the type hinted at towards the end of section 1, which brought in considerations on possible majorities undermining the proposition at issue. It is to such attempts at repairing voting rules that Proposition 3 applies. What our result shows, and what at least to our eyes came as a surprise, is that no matter how complicated we make such attempts at repairing the voting rule, as long as it is structural there is no guarantee that application of it will result in a consistent collective opinion state, even if all voters can be assumed to have

consistent opinion states.[10]

Further, Proposition 3 invites a comparison with Proposition 1 of List and Pettit, and with Pauly and van Hees's generalization of their theorem. Here we want to emphasise again that Proposition 3 is based on the construction M_W involving party-wise opinionated parliaments, whereas almost all other results employ the Universal Domain assumption. In this sense the present result is simply different. Having said that, let us turn to these other results. As for List and Pettit, note first that we must make rather different assumptions on the agenda. For some parliaments the agenda may be equally minimal, but the interdependence between agenda and parliament remains and will in some cases lead to rather rich agendas. On the other hand, the conditions of our impossibility result are weaker than theirs in a number of respects: our result does not assume Consistent and Complete Range, but only Consistent and Aggregative Range, and via Structuralness it only assumes restricted forms of Anonymity and Neutrality. Above all, our result does not require Independence.

In this latter respect our result is also stronger than the result of Pauly and van Hees. But this is not so for the other conditions, although the comparison is not entirely clear, because our conditions employ the fixed valuation of M_W. First, in the guise of Structuralness we assume Anonymity, but only at the party-wise opinionated profiles, while Pauly and van Hees assume only Responsiveness, but over the whole domain of consistent and complete valuations. So the generalizations are in a sense orthogonal. Further, our result leads to the requirement of Unanimity at party-wise opinionated profiles, while Pauly and van Hees require Non-Dictatorship at all profiles in the domain. So in this sense their result is stronger. Finally, Pauly and van Hees are also more general in that they drop the condition of Neutrality altogether, while the above result still assumes the weakened kind of Neutrality implicit in Structuralness. The complete absence of Neutrality in Pauly and van Hees's paper allows us to tell apart propositions on the basis of their non-formal (most likely, semantical) properties.

This relates to our next point, which is that our result may be less dramatic than the corresponding one about the lottery paradox. At least it is quite clear that many have hoped for a (non-trivial) formal solution to the lottery paradox, and even for a formal theory of rationality (which would seem to presuppose a formal solution to the lottery paradox). It is not so clear that something similar holds true for voting rules. Although, as we said above, the paradigmatic rule of majority voting *is* structural, and although many parliaments may very well be opinionated, it may be argued that in general voting rules should be sensitive

[10]Note that, while the condition of Structuralness is rather weak in that it includes all formal voting rules, it excludes voting rules that make the inclusion of a proposition in the collective opinion state depend on the propositions (if any) that have already been included, or more generally on the order of voting on the propositions in the agenda. Such rules violate the condition of Structuralness, because the position of propositions in the order of the voting agenda is not invariant under automorphisms. In other words, the Structuralness of the voting rule excludes Harman's response to the lottery paradox, as mentioned in note 7, when that response is translated for the discussion of the discursive dilemma.

to the semantic content of the various propositions that are on the agenda, already for reasons independent of our result. A voting rule might then set higher standards for acceptance for (say) propositions whose acceptance would lead to tax benefits for farmers than for (say) propositions whose acceptance would have the effect of lowering the emission of pollutants. Be that as it may, it will still be good to know that already for purely logical reasons voting rules will have to be cast, at least partly, in non-formal terms.

Finally, we would like to point to a possible avenue for further research. We established an isomorphism between a structure relevant to the lottery paradox and one relevant to the discursive dilemma. This allowed us to employ a theorem concerning the lottery paradox in the context of judgement aggregation. But the bridge we built between the two discussions can also be crossed in the other direction, of course. And given the liveliness of the debate on judgement aggregation, and the many new results that keep coming out of that, it is not unrealistic to expect that at least some theorems originally derived, or still to be derived, within that context can be applied fruitfully to the context of the lottery paradox, and will teach us something new, and hopefully also important, about this paradox.

References

Bovens, L. [2006] "The Doctrinal Paradox and the Mixed-Motivation Problem," *Analysis* 66:35–39.

Dietrich, F. and List, C. [2006] "The Impossibility of Unbiased Judgment Aggregation," manuscript.

Dietrich, F. and List, C. [2006a] "Judgement Aggregation without Full Rationality," manuscript.

Douven, I. [2002] "A New Solution to the Paradoxes of Rational Acceptability," *British Journal for the Philosophy of Science* 53:391–410.

Douven, I. [2007] "The Lottery Paradox and Our Epistemic Goal," *Pacific Philosophical Quarterly*, in press.

Douven, I. and Williamson, T. [2006] "Generalizing the Lottery Paradox," *British Journal for the Philosophy of Science*, in press.

Harman, G. [1986] *Change in View*, Cambridge MA: MIT Press.

Kyburg, H. [1961] *Probability and the Logic of Rational Belief*, Middletown CT: Wesleyan University Press.

List, C. and Pettit, P. [2002] "Aggregating Sets of Judgements: An Impossibility Result," *Economics and Philosophy* 18:89–110.

Nelkin, D. [2000] "The Lottery Paradox, Knowledge, and Rationality," *Philosophical Review* 109:373–409.

Pauly, M. and van Hees, M. [2006] "Logical Constraints on Judgement Aggregation," *Journal of Philosophical Logic*, in press.

Pollock, J. [1990] *Nomic Probability and the Foundations of Induction*, Oxford: Oxford University Press.

Hybrid Voting Protocols and Hardness of Manipulation

Edith Elkind and Helger Lipmaa

Abstract

This paper addresses the problem of constructing voting protocols that are hard to manipulate. We describe a general technique for obtaining a new protocol by combining two or more base protocols, and study the resulting class of (vote-once) hybrid voting protocols, which also includes most previously known manipulation-resistant protocols. We show that for many choices of underlying base protocols, including some that are easily manipulable, their hybrids are NP-hard to manipulate, and demonstrate that this method can be used to produce manipulation-resistant protocols with unique combinations of useful features.

1 Introduction

In multiagent systems, the participants frequently have to agree on a joint plan of action, even though their individual opinions about the available alternatives may vary. Voting is a general method of reconciling these differences, and having a better understanding of what constitutes a good voting mechanism is an important step in designing better decision-making procedures. In its most general form, a voting mechanism is a mapping from a set of votes (i.e., voters' valuations for all alternatives) to an ordering of the alternatives that best represents the collective preferences. In many cases, however, the attention can be restricted to mechanisms that interpret their inputs (votes) as total orderings of the alternatives/candidates and output a single winner. A classical example here is Plurality voting, where only the top vote of each voter is taken into account, and the candidate with the largest number of top votes wins (to specify the protocol completely, we also need a draw resolution rule for the case where more than one voter gets this number of votes).

A fundamental problem encountered by all voting mechanisms is *manipulation*, i.e., the situation when a strategizing voter misrepresents his preferences in order to obtain a more desirable outcome. One can expect that rational agents will engage in manipulation whenever it is profitable for them to do so; as a result, the output of the voting mechanism may grossly misrepresent the actual preferences of the agents and be detrimental to the system as a whole.

It is well-known [8, 11] that any nondictatorial voting mechanism for three or more candidates is susceptible to manipulation. However, while there is no information-theoretic solution to this problem, one can try to discourage potential manipulators by making manipulation infeasible. This approach is particularly attractive in multiagent setting, when decisions have to be made in real time, and whether an agent can find a beneficial manipulation quickly is more important than whether such a manipulation exists in principle. It turns out that some of the voting protocols that are used in

[1] An earlier version of this paper appeared in ISAAC'05

practice enjoy this property: it has been shown [1, 2] that second-order Copeland and Single Transferable Vote (STV) are NP-hard to manipulate. Furthermore, in a recent paper [4], Conitzer and Sandholm showed that several protocols, including Borda, STV, Maximin and Plurality, can be modified so that manipulating them becomes computationally hard. Their method involves prepending the original protocol by a pre-round in which candidates are divided into pairs and the voters' preferences are used to determine the winner of each pair; the winners of the pre-round participate in elections conducted according to the original protocol. Different methods for pairing up the candidates and eliciting the votes give rise to different levels of complexity, such as NP-hardness, #P-hardness, or PSPACE-hardness.

The advantage of this method of constructing manipulation-resistant protocols is in preserving some of the properties of the original protocol: for example, if the base protocol is Condorcet-consistent (see Section 6 for definition), then the modified protocol is Condorcet-consistent as well. However, for some other desirable features this is not true, and, generally, eliminating half of the candidates using a set of criteria that may be very different in spirit from those used by the original protocol, is likely to alter the outcome considerably, so that the desiderata that motivated the original protocol may no longer be attainable.

We build upon the ideas of [4] to construct a larger family of protocols that are hard to manipulate. We observe that their pre-round phase can be viewed as the first stage of the voting protocol known as Binary Cup (BC) (defined in Section 2). While this protocol itself is not hard to manipulate (at least, when the schedule is known in advance), the results of [4] can be interpreted as showing that combining BC with other protocols results in manipulation-resistant schemes. We generalize this idea by showing that this kind of hardness amplification is not unique to BC.

We define the class of *(vote-once) hybrid voting protocols* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$. In $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$, after the voters have expressed their preferences, k steps of protocol X are performed to eliminate some of the candidates, and then protocol Y is run on the rest of the candidates, reusing the votes as restricted to the remaining candidates. In practice, such a reuse of votes is important, since it allows voters to only express their preferences once; this feature is desirable both for actual elections, where it is difficult to get citizens to the voting booths more than once, and for artificial agents, where round complexity of a protocol may be an issue. Clearly, the protocols of [4] belong to this family, as does STV; therefore, our framework encompasses most of the known hard-to-manipulate voting mechanisms.

We show that many other hybrid protocols are NP-hard to manipulate as well. Specifically, we consider several well-known protocols, such as Plurality, Borda, STV, and Maximin, and prove that many hybrids of these protocols are manipulation-resistant. We do this by formulating some fairly general conditions on X and Y under which the protocols of the form $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Plurality})$, $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$, or $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ are NP-hard to manipulate. Additionally, we show that a hybrid of a protocol with itself may be different from the original protocol — and much harder to manipulate. We prove that this is, indeed, the case for Borda protocol: $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ is NP-hard to manipulate, while Borda itself is easily manipulable.

We define a generic closure operation on protocols that makes them closed un-

der hybridization. Interestingly, applying this operation to the easy-to-manipulate Plurality results in the hard-to-manipulate STV. We conjecture that for many other basic protocols, their closed versions are NP-hard to manipulate as well. Whenever this is the case, the closed protocols provide the most faithful manipulation-resistant approximation to the underlying protocols, which makes them compelling alternatives to the original protocols.

On the flip side, we demonstrate that hybridization does not always result in hard-to-manipulate protocols: in particular, the hybrid protocols that use Plurality as their first component, are almost as easy to manipulate as their second component. Finally, we demonstrate that our techniques extend to voting protocols that allow voters to rate the candidates rather than just order them.

The value of our results is not so much in constructing specific new manipulation-resistant protocols, but rather in providing a general method for doing that, which can be used with many basic schemes. Since a hybrid inherits some of the properties of its ingredients, we get hard-to-manipulate protocols with properties not shared by the schemes from [1, 2, 4]. For example, since BC is not Pareto-optimal, all protocols obtained by the method of [4] are not Pareto-optimal either, while our approach allows to construct hybrids that have this valuable feature (for definitions, see Section 6). It has already been argued in [4] that it is desirable to have manipulation-resistant protocols that can be used in different real-life situations; our method fits the bill.

The use of voting and voting-related techniques is not restricted to popular elections: the ideas from this domain have been applied in rank aggregation [5, 9], recommender systems [10], multiagent decision making in AI [7], etc. In many of these settings, the number of alternatives is large enough to make our results applicable, and, furthermore, the agents are both sufficiently sophisticated to attempt manipulation and may derive significant utility from doing so. Therefore, we feel that it is important to have a better understanding of what makes voting protocols hard to manipulate, as this will allow us to design more robust decision-making systems that use voting-like methods.

The rest of the paper is organized as follows. In Section 2 we introduce our notation, give a precise definition of what it means to manipulate an election, and describe some well-known voting schemes discussed in the paper. In Section 3, we define hybrid protocols and some related notions. In Section 4, we show that certain hybrid protocols are NP-hard to manipulate. In Section 5, we discuss hybrids obtained by combining a protocol with itself. In Section 6, we define some desirable properties of voting protocols, show that many of them are preserved under hybridization, and demonstrate that our protocols can provide useful combinations of these properties. In Section 7, we provide examples of hybrids that are easy to manipulate and discuss limitations and extensions of our approach. Finally, in Section 8, we present our conclusions and future research directions.

2 Preliminaries and Notation

We assume that there are n voters and m candidates and denote the set of all voters by $V = \{v_1, \ldots, v_n\}$ and the set of all candidates by $C = \{c_1, \ldots, c_m\}$. Most of our complexity results are in terms of m and n, i.e., unless specified otherwise,

'polynomial' always means 'polynomial in m and n'.

The set of all permutations of C is denoted by $\Pi(C)$; the preference of the ith voter is expressed by a list $\pi_i \in \Pi(C)$: the first element is the voter's most preferred candidate, etc. In particular, this means that within one voter's preference list, ties are not allowed. We write $(\dots, c_i, \dots, C_j, \dots)$ to denote that a voter prefers c_i to all candidates in C_j, without specifying the ordering of candidates within C_j. For any subset $C' \subseteq C$, let $\pi|_{C'}$ be the permutation π as restricted to C' (i.e., elements not from C' are omitted). Note that $\pi|_{C'}$ corresponds to a valid preference in an election that has the candidate set C'.

When describing the preferences of a single voter v, we write $c_i \succ_v c_j$ to denote that v prefers c_i to c_j. Similarly, we write $C_i \succ_v C_j$ to denote that v prefers all candidates in the set C_i to all candidates in the set C_j, without specifying the ordering of candidates within C_i and C_j. When the identity of the voter is clear from the context, we omit the subscript and write $\succ$ instead of $\succ_v$.

A *voting protocol* is a mapping $P : \Pi(C) \times \dots \times \Pi(C) \mapsto C$ that selects a winner $c \in C$ based on all voters' preference lists. In this paper, we consider the following common voting protocols (in all definitions that mention points, the candidate with the most points wins):

Plurality: A candidate receives 1 point for every voter that ranks it first.

Borda: For each voter, a candidate receives $m-1$ point if it is the voter's top choice, $m-2$ if it is the second choice, ..., 0 if it is the last.

Single Transferable Vote (STV): The winner determination process proceeds in rounds. In each round, a candidate's score is the number of voters that rank it highest among the remaining candidates, and the candidate with the lowest score drops out. The last remaining candidate wins. (A vote transfers from its top remaining candidate to the next highest remaining candidate when the former drops out.)

Maximin: A candidate's score in a pairwise election is the number of voters that prefer it over the opponent. A candidate's number of points is the lowest score it gets in any pairwise election.

Binary Cup (BC): The winner determination process consists of $\lceil \log m \rceil$ rounds. In each round, the candidates are paired; if there is an odd number of candidates, one of them gets a bye. The candidate that wins the pairwise election between the two (or got a bye) advances into the next round. The schedule of the cup (i.e., which candidates face each other in each round) may be known in advance (i.e., before the votes are elicited) or it may depend on the votes.

Voting Manipulation

We say that a voter v_j can *manipulate* a protocol P if there is a permutation $\pi'_j \in \Pi(C)$ such that for some values of $\pi_i \in \Pi(C)$, $i = 1, \dots, n$, we have (i) $P(\pi_1, \dots, \pi_n) = c$; (ii) $P(\pi_1, \dots, \pi_{j-1}, \pi'_j, \pi_{j+1}, \dots, \pi_n) = c' \neq c$; (iii) v_j ranks c' above c. We say that v_j manipulates P *constructively* if v_j ranks c' first and *destructively* otherwise. All

results in this paper are on constructive manipulation; in what follows, we omit the word 'constructive'. A voter v_j manipulates P *efficiently* if there is a polynomial time algorithm that given preference lists $\pi_1, \ldots, \pi_n$ for which such π'_j exists, can find one such π'_j.

3 Hybrid Protocols

In this section, we formally define *(vote-once) hybrid protocols*. Intuitively, a hybrid of two protocols X and Y executes several steps of X to eliminate some of the candidates, and then runs Y on the remaining set of candidates. To make this intuition precise, however, we have to define how to interpret the first protocol X as a sequence of steps. While there is no obvious way to do this for an arbitrary protocol, most well-known protocols, including the ones described in Section 2, admit such an interpretation. In particular, we suggest the following definitions:

- For STV, a *step* is a single stage of the protocol. That is, a step of STV consists of eliminating a candidate with the least number of first-place votes and transferring each vote for this candidate to the highest remaining candidate on that ballot.
- For Binary Cup (BC), a *step* is a single stage of the protocol as well, i.e., it consists of pairing up the candidates and eliminating the ones who lose in the pairwise comparison.
- For point-based protocols, such as Plurality, Borda, or Maximin, we first compute the scores of all candidates, order them by their scores from the lowest to the highest, and define a *step* to consist of eliminating the first (i.e., the lowest ranked) remaining candidate in this sequence. Note that the scores are not recomputed between the steps. (A similar approach can be applied to any voting protocol that can be extended to a preference aggregation rule, i.e., a function that maps votes to total orderings of the candidates. In this case, the order in which the candidates are eliminated is obtained by inverting the output of the preference aggregation rule.)

Definition 1. *A* hybrid protocol $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$ *consists of two* phases*. Suppose that the voters' preference lists are described by the n-tuple $(\pi_1, \ldots, \pi_n)$. In the first phase, the protocol executes k steps of $\mathsf{X}(\pi_1, \ldots, \pi_n)$; suppose that S is the set of candidates not eliminated in the first phase. In the second phase, the protocol applies Y to $(\pi_1|_S, \ldots, \pi_n|_S)$, i.e., the preference lists restricted to the remaining set S of candidates.*

It is easy to extend this definition to hybrids $\mathsf{Hyb}(\mathsf{X}^{(1)}_{k_1}, \mathsf{X}^{(2)}_{k_2}, \ldots, \mathsf{X}^{(t)}_{k_t}, \mathsf{Y})$ of three or more protocols.

4 Hardness Results

4.1 Hardness of STV-Based Hybrids

In this subsection, we show that hybrids $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ and $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$ are NP-hard to manipulate for many "reasonable" voting protocols X and Y, including the

cases $\mathsf{X}, \mathsf{Y} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}\}$.

Theorem 1. *A hybrid of the form* $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ *is NP-hard to manipulate as long as* Y *satisfies the following condition: Whenever there is a candidate* c *who receives* K *first-place votes and* $n - K$ *second-place votes, while all other candidates receive at most* $K - 1$ *first-place vote,* Y *declares* c *the winner.*

The proof appears in the full version of the paper.

Corollary 1. *The hybrids of the form* $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$*, where* $\mathsf{Y} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}, \mathsf{STV}\}$*, are NP-hard to manipulate.*

The proof of this corollary is straightforward since all these voting protocols satisfy the required property.

Theorem 2. *A hybrid of the form* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$ *is NP-hard to manipulate if* X *satisfies the following condition for some unbounded nondecreasing function* $f(\cdot)$ *and infinitely many* K*: Suppose that all but one voter rank some* K *candidates* $c_1, \ldots, c_K$ *after all other candidates, and all other candidates receive at least 2 first-place votes. Then after* $f(K)$ *steps of* X*, the set of eliminated candidates is a subset of* $\{c_1, \ldots, c_K\}$.

Sketch. Set $k = f(K)$. Denote the set of candidates in the construction of [2] by C'; let $C'' = \{c_1, \ldots, c_K\}$ and $C = C' \cup C''$. Modify the votes of all honest voters in that construction so that they rank C' above C''. The reduction of [2] has the property that each candidate in C' gets more than 2 first-place votes. Hence, the set of candidates eliminated in k rounds of X is a subset of C''; furthermore, the remaining candidates from C'' will be the first candidates eliminated by STV. Hence, no matter how the manipulator ranks the candidates in C'', it has no effect on the execution of the protocol. Therefore, his vote can be interpreted as a vote in the original STV and vice versa. □

Corollary 2. *The hybrids of the form* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$*, where* $\mathsf{X} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}\}$*, are NP-hard to manipulate.*

Proof. It is easy to see that $\mathsf{Plurality}$, $\mathsf{Maximin}$ and BC satisfy the condition of the theorem. For Borda, it is satisfied whenever the number of voters exceeds the number of candidates; in the construction of [2], the number of voters is larger than $3|C'|$, so we can set $K = |C'|$. □

Our proofs that hybrids using STV as their first or second component are NP-hard to manipulate rely on some specific properties of the reduction constructed in [2]. In the full version of the paper, we provide black-box constructions, i.e., ones that work with any NP-hardness proof.

4.2 Hybrids of the Form $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Plurality})$

In this subsection, we prove that $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Plurality})$ is hard to manipulate whenever X satisfies Property 1. While this property might seem artificial, we show that it is possessed by at least two well-known protocols, namely, Borda and $\mathsf{Maximin}$.

Property 1. *For any set $G = \{g_1, \ldots, g_N\}$, any collection $S = \{s_1, \ldots, s_M\}$ of subsets of G, and any $K \leq M$, there are some k', $k' \leq M$, and T, $T > 3N$, such that it is possible to construct in polynomial time a set of $T + N(T-2) + 3N$ votes over the set of candidates $C' \cup C'' \cup \{p\}$, where $C' = \{c'_1, \ldots, c'_N\}$, $C'' = \{c''_1, \ldots, c''_M\}$, so that*

- *there are T voters who rank p first;*
- *for each $i = 1, \ldots, N$, there are $T-2$ voters who rank c'_i first;*
- *for each $i = 1, \ldots, N$, there are 3 voters who rank all c''_j such that $g_i \in s_j$ above c'_i, and rank c'_i above all other candidates;*
- *for any additional vote π, when it is tallied with all other votes, the set of candidates eliminated in the first k' rounds is a subset of C'' of size $M - K$;*
- *for any subset $S' \subseteq S$, $|S'| = M - K$, one can design in polynomial time a vote $\pi_{S'}$ that, when tallied with other votes, guarantees that the set of candidates eliminated in the first k' rounds is exactly $\{c''_i \mid s_i \in S'\}$.*

Theorem 3. *A hybrid of the form* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Plurality})$ *is NP-hard to manipulate constructively whenever X satisfies Property 1.*

Proof. We give a reduction that is based on the NP-hard problem SET COVER. Recall that SET COVER can be stated as follows: Given a ground set $G = \{g_1, \ldots, g_N\}$, a collection $S = \{s_1, \ldots, s_M\}$ of subsets of G, and an integer K, does there exist a K-cover of G, i.e., a subset S' of S, $S' = \{s_1, \ldots, s_K\}$, such that for every $g_i \in G$ there is an $s_j \in S'$ such that $g_i \in s_j$?

Construct the set of votes based on G, S, and K so that it satisfies Property 1. Let $k = k'$, and let p be the manipulator's preferred candidate. We show that the manipulator can get p elected under $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Plurality})$ if and only if he can find a set cover for G. Indeed, after k rounds of X, all candidates in $C' \cup \{p\}$ survive, as well as exactly K candidates from C''. We show that p wins if and only if these K candidates correspond to a set cover of G. Observe that any surviving candidate from C'' has at most $3N < T$ first-place votes, so he cannot win in the last stage. Now, consider a candidate $c'_i \in C'$. Suppose that the corresponding element is not covered, i.e., all c''_j such that $g_i \in s_j$ are eliminated. Then after the end of the first phase, c'_i has $T + 1$ first-place vote, while p has T first-place votes, so in this case p cannot win.

On the other hand, suppose that for any $g_i \in G$ there is an $s_j \in S$ such that $g_i \in s_j$ and c''_j is not eliminated in the first phase. Then at the beginning of the second phase each $c'_i \in C'$ has $T-2$ first-place votes, while p has T first-place votes, so in this case p wins.

Hence, manipulating this protocol is equivalent to finding a set cover of size K. □

Corollary 3. *The protocols* $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Plurality})$ *and* $\mathsf{Hyb}(\mathsf{Maximin}_k, \mathsf{Plurality})$ *are NP-hard to manipulate.*

Proof. Let the voters who rank p first, rank the candidates in C' above those in C'', and the voters who rank c'_i first, rank the candidates in $p \cup C'$ above those in C''. For large enough T, this guarantees that both Borda and Maximin scores of the candidates in $C' \cup \{p\}$ are much higher than those of the candidates in C'', so none of the candidates in $C' \cup \{p\}$ can be eliminated in the first phase. On the other hand, we still have enough flexibility to ensure that all candidates in C'' have the same Borda (or Maximin) score with respect to the honest voters' preferences. Then, for both protocols, the manipulator can get any $M - K$ candidates from C'' eliminated by putting them on the bottom of his vote and ranking the remaining K candidates above the candidates in $C' \cup \{p\}$. Thus, both Borda and Maximin satisfy all conditions in the statement of Theorem 3. □

Together with our results on STV and the results of [4], the constructions of this section provide a wide choice of manipulation-resistant protocols. In the next section, we add to our repertoire two more protocols that are hard to manipulate, namely, $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ and $\mathsf{Hyb}(\mathsf{Maximin}_k, \mathsf{Borda})$.

5 Hybrid of a Protocol with Itself

We say that a protocol is *hybrid-proof* if a hybrid of several copies of this protocol has the same outcome as the original protocol. While some protocols, such as STV or Binary Cup, have this property, for many other protocols, especially score-based ones, this is not the case. To see this, note that in a hybrid protocol, the scores of all surviving candidates are recomputed in the beginning of the second phase, while in the original protocol they are computed only once. As a result, in a hybrid of, say, two copies of the Plurality protocol, one candidate may gain a lot of first-place votes from voters who rank him right after the candidates that were dropped in the first phase, while some other candidate may get no extra votes at all; a similar phenomenon happens in Borda and Maximin.

Nevertheless, any protocol can be modified to be hybrid-proof. For an arbitrary protocol X, define a *closed protocol* $\overline{\mathsf{X}}$ by $\overline{\mathsf{X}} = \mathsf{Hyb}(\mathsf{X}_1, \ldots, \mathsf{X}_1)$, where the number of copies of X_1 is such that $\overline{\mathsf{X}}$ selects a single winner; a step of $\overline{\mathsf{X}}$ corresponds to a single copy of X_1.

Proposition 1. *For any protocol* X*, the closed protocol* $\overline{\mathsf{X}}$ *is hybrid-proof.*

We omit the proof.

Interestingly, $\mathsf{Hyb}(\mathsf{Plurality}_1, \ldots, \mathsf{Plurality}_m) = \mathsf{STV}$: the vote transfer mechanism can be viewed as recomputing each candidate's Plurality score. Observe that while Plurality has particularly bad manipulation resistance properties (see, e.g., Section 7), STV is NP-hard to manipulate. This leads us to conjecture that for many other base protocols, the new protocols obtained in this manner are NP-hard to manipulate. Whenever this is the case, the closed protocols provide the most faithful manipulation-resistant approximation to the underlying protocols, which makes them compelling alternatives to the original protocols. This conjecture is supported by the fact that for some easy-to-manipulate protocols, a hybrid of just two copies of the protocol is NP-hard to manipulate; increasing the number of copies should make the manipulation

harder, not easier. As an illustration, we prove that a hybrid of two instances of Borda is NP-hard to manipulate.

Theorem 4. *The hybrid* $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ *is NP-hard to manipulate.*

Proof. We give a reduction from EXACT COVER BY 3-SETS, which is stated as follows: Given a ground set $G = \{g_1, \ldots, g_N\}$, $N = 3L$, and a collection $S = \{s_1, \ldots, s_M\}$ of 3-element subsets of G, does there exist an exact set cover of G, i.e., a subset S' of S, $S' = \{s_1, \ldots, s_{N/3}\}$ such that for every $g_i \in G$ there is a unique $s_j \in S'$ such that $g_i \in s_j$?

We construct two sets of voters V', $|V'| = 2N+2$, and V'', $|V''| = (M+1)(N+1)$ and define $V = V' \cup V''$. Let $C^g = \{c_1^g, \ldots, c_N^g\}$ and $C^s = \{c_1^s, \ldots, c_M^s\}$, and let the set of candidates be $C = C^g \cup C^s \cup \{c_0\} \cup p$, where p is the manipulator's preferred candidate.

For each $i = 1, \ldots, N-1$, there are 2 voters in V' who rank the candidates as

$$(c_{i+1}^g, c_{i+2}^g, \ldots, c_N^g, p, c_1^g, \ldots, c_{i-1}^g, C_i^s, c_i^g, C^s \setminus C_i^s, c_0) \ , \tag{1}$$

where $C_i^s = \{c_j^s \mid g_i \in s_j\}$. Also, there are 2 voters who rank the candidates as

$$(p, c_1^g, c_2^g, \ldots, c_{N-1}^g, C_N^s, c_N^g, C^s \setminus C_N^s, c_0) \ , \tag{2}$$

where $C_N^s = \{c_j^s \mid g_N \in s_j\}$, a voter who ranks the candidates as $(c_1^g, c_2^g, \ldots, c_N^g, c_0, p, C^s)$, and a voter who ranks the candidates as $(c_1^g, c_2^g, \ldots, c_N^g, p, c_0, C^s)$.

In V'', for each $i = 1, \ldots, N-1$, there are $M+1$ voters who rank the candidates as

$$(c_{i+1}^g, c_{i+2}^g, \ldots, c_N^g, p, c_1^g, \ldots, c_i^g, c_0, C^s) \ , \tag{3}$$

$M+1$ voters who rank the candidates as $(p, c_1^g, c_2^g, \ldots, c_N^g, c_0, C^s)$, and $M+1$ voters who rank the candidates as $(c_1^g, c_2^g, \ldots, c_N^g, p, c_0, C^s)$.

Set $k = M - N/3$. We can set the voters' preferences over the candidates in C^s so that everyone in C^s has the same Borda score, in which case the manipulator's vote will determine which k of them will be eliminated in the first phase.

Suppose that the manipulator votes so that the set of candidates from C^s who survive the first phase corresponds to an exact set cover of G. Then for each candidate c_i^g and any $j = 1, \ldots, N$, there are two voters in V' who rank him in the jth position and two voters in V' who rank him in the $(N+2)$nd position (these two voters prefer c_j^s to c_i^g, where s_j is the set in the set cover that contains g_i). Hence, the Borda score of each candidate in C^g with respect to V' is $\sum_{t=m-k-N}^{m-k-1} 2t + 2(m-k-N-2)$.

On the other hand, the Borda score of p with respect to V' is $\sum_{t=m-k-N}^{m-k-1} 2t + (m-k-N-1) + (m-k-N-2)$, and the score of c_0 is lower that the score of any candidate in $C^g \cup \{p\}$, so in this case p wins.

Conversely, suppose that the set of candidates from C^s who survive the first phase does not correspond to a set cover of G. Consider an element $g_i \in G$ that is not covered. All voters in V' prefer c_i^g to all surviving candidates in $C^s \cup \{c_0\}$, which means that his Borda score is higher than that of p. □

Using the same construction, one can show that $\mathsf{Hyb}(\mathsf{Maximin}_k, \mathsf{Borda})$ is NP-hard to manipulate for infinitely many values of k; we omit the details.

6 Properties of Voting Protocols

Voting protocols are evaluated based on various criteria, such as

(1) *Pareto-optimality*: a candidate who is ranked lower than some other candidate by every voter never wins;

(2) *Condorcet-consistency*: if there is a candidate who is preferred to every other candidate by a majority of voters, this candidate should be the winner of the election;

(3) *Monotonicity*: with the relative order of the other candidates unchanged, ranking a candidate higher should never cause the candidate to lose, nor should ranking a candidate lower ever cause the candidate to win.

In the context of this paper, a natural addition to this list is *hardness of manipulation*.

Most voting schemes based on pairwise comparisons, in particular, BC and Maximin, are Condorcet-consistent, while for STV, or positional methods, such as Plurality or Borda, this is not the case. One can prove that Plurality, Borda, Maximin, and BC are monotone, while STV is not. All basic voting protocols considered in this paper except BC are Pareto-optimal.

To analyze whether properties (1)–(3) are preserved under hybridization, we have to extend these definitions to multi-step protocols. We say that a multi-step protocol is *strongly Pareto-optimal* if whenever every voter ranks c_1 below c_2, c_1 is eliminated before c_2, and *strongly monotone* if ranking a candidate higher does not affect the relative order of elimination of other candidates and cannot result in him being eliminated at an earlier step; the definition of Condorcet consistency remains unchanged. It is easy to see that multi-step versions of Pareto-optimal protocols that we consider are strongly Pareto-optimal, at least for some draw resolution rules. However, not all monotone protocols are strongly monotone: for example, in Borda, moving a candidate several positions up changes other candidates' scores in a non-uniform way.

Proposition 2. *For any voting protocols* X *and* Y *and any* $k > 0$*, if both* X *and* Y *are Condorcet-consistent, so is* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$*; if* X *is strongly Pareto-optimal (strongly monotone) and* Y *is Pareto-optimal (monotone), then* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$ *is Pareto-optimal (monotone).*

We omit the proofs.

The construction proving that BC is not Pareto-optimal can be easily modified to show that any protocol of the form $\mathsf{Hyb}(\mathsf{BC}_k, \mathsf{Y})$ is not Pareto-optimal for some k, where $\mathsf{Y} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{STV}\}$. Hence, prior to this work, the only Pareto-optimal mechanisms that were known to be NP-hard to manipulate were STV and the variants of the Copeland protocol that were described in [1]. Our results imply that $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Plurality})$, $\mathsf{Hyb}(\mathsf{Maximin}_k, \mathsf{Plurality})$, and $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ also combine these two properties.

Furthermore, except for STV, all previous hard-to-manipulate protocols involved methods that use pairwise comparisons, and such methods have been criticized for relying too much on the number of victories rather than their magnitude. On the other hand, both $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Plurality})$ and $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ are based purely on positional methods, which do not suffer from this flaw, and Maximin (and hence, hybrids of Maximin with positional methods) also takes into account the magnitude of victories.

7 Limitations and Extensions

7.1 Hybrids That Are Easy to Manipulate

Unfortunately, our method of obtaining hard-to-manipulate protocols is not universal: if the protocol used in the first phase does not provide the manipulator with sufficiently many choices, the resulting hybrid protocol is almost as easy to manipulate as its second component. In particular, this applies to Plurality protocol.

Theorem 5. *Suppose that a protocol* Y *satisfies the following property for any candidate* c*: Given other voters' preference profiles, the manipulator can in polynomial time find a beneficial manipulation that ranks* c *first or infer that no such manipulation exists. Then there is a polynomial-time algorithm that can constructively manipulate the hybrid* $\mathsf{Hyb}(\mathsf{Plurality}_k, \mathsf{Y})$ *for any* k*.*

Proof. For the first phase of the protocol, the only choice that the manipulator has to make is which candidate to rank first; the rest of his vote will have no effect on the elimination process. Hence, he can try all m options. Suppose that when the manipulator ranks c_i first, the set of candidates that survive the first phase is C_i. The manipulator can deduce the honest voters' preferences over C_i. If $c_i \notin C_i$, he simply has to construct a beneficial manipulation $\pi|_{C_i}$ of Y and, in his vote, rank c_i first and order the candidates in C_i as suggested by $\pi|_{C_i}$. If $c_i \in C_i$, in constructing a beneficial manipulation of Y he is restricted to orderings that rank c_i first. By our assumptions, he can find a solution to this problem in polynomial time. □

Corollary 4. *There are polynomial-time algorithms that can constructively manipulate* $\mathsf{Hyb}(\mathsf{Plurality}_k, \mathsf{Y})$*, where* $\mathsf{Y} \in \{\mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}, \mathsf{Plurality}\}$ *for any* k*.*

The property of Plurality that makes it an unsuitable candidate for the first phase of a hybrid protocol is that by altering his vote, the manipulator can obtain at most m different outcomes of the first phase, so he can go over all of them and pick the one that produces best results. It is not clear whether any other protocol for which changing a single vote leads to polynomially many different outcomes is just as bad: each outcome imposes specific restrictions on the manipulator's vote in the second phase, and finding a manipulation that satisfies them may be harder than manipulating the original protocol.

7.2 Other Measures of Complexity

In their paper [4], Conitzer and Sandholm prove that under some pre-round scheduling algorithms, many protocols become #P-hard or PSPACE-hard to manipulate when preceded by a BC pre-round, and [6] shows that one can make manipulation as hard as

inverting one-way functions. However, since other protocols that we consider do not have the flexibility provided by the BC scheduling step, the problem of manipulating the hybrids whose first component is not BC, but some other protocol from our list, is inherently in NP. Consequently, a proof that these hybrids are #P-hard or PSPACE-hard to manipulate will lead to a collapse of the polynomial hierarchy, and hence is unlikely.

For the entire class of voting protocols considered in this paper, manipulation is easy when the number of candidates m is very small. This applies both to the standard protocols like STV and to the new hybrid protocols. Indeed, since there are only $m!$ possible ballots for the manipulator, he can go over all of them in order to determine which of them produces the best outcome.

7.3 Utility-Based Voting

In previous sections, we investigated voting schemes that required each voter to submit a total ordering of the candidates. However, in many settings a voter may be essentially indifferent between some of the alternatives, but have a strong opinion on the relative merit of other alternatives. In this case, his preference may be better reflected by a *utility vector* $\mathbf{u} = (u_1, \ldots, u_m)$, where $0 \leq u_j \leq 1$ is the *utility* that this voter assigns to candidate c_j. To guarantee fairness, the utility vectors are normalized, i.e., we require that either $u_j = 0$ for all j or $\sum_j u_j = 1$. In addition, we require that all u_j are rational numbers whose representation size is polynomial in n and m.

The definitions of a voting protocol and manipulation can be modified in a straightforward manner. A hybrid of two utility-based protocols is a protocol that performs k steps of the first protocol, re-normalizes the utility vectors (restricted to the surviving candidates) and executes the second protocol on the remaining candidates.

The most natural voting protocol for the utility-based framework is HighestScore, which computes the total score of each candidate, i.e., the sum of utilities assigned to this candidate by all voters, and selects the candidate with the highest total score. However, this protocol is not manipulation-resistant.

Proposition 3. *There is a polynomial-time algorithm that can manipulate* HighestScore.

Fortunately, it turns out that the techniques we use for ordering-based protocols are applicable in this setting, too.

A *step* of HighestScore is naturally defined as eliminating the candidate with the lowest score; consequently, the hybrid protocol $\mathsf{Hyb}(\mathsf{HighestScore}_k, \mathsf{HighestScore})$ consists of eliminating k candidates with the lowest score, renormalizing the utility vectors, and choosing the candidate with the highest score among the remaining candidates.

Theorem 6. $\mathsf{Hyb}(\mathsf{HighestScore}_k, \mathsf{HighestScore})$ *is NP-hard to manipulate.*

Another way to increase resistance to manipulation is to use the method of [4], i.e., prepend HighestScore with a pre-round. A technical difficulty that arises here is that in [4], the pre-round winners are determined on the basis of comparisons, while in our setting, this information may not be available (utility vectors allow for draws).

This can be resolved either by requiring the voters to submit an ordering together with their utility vector (clearly, the two should be consistent) or by determining the winner of each pre-round pair by comparing their scores. Both approaches result in hybrid protocols that are NP-hard to manipulate.

8 Conclusions and Future Work

Our work places the results of [3, 4] within a more general paradigm of hybrid voting schemes. The advantage of our approach is that it works for a wide range of protocols: while some voting procedures are inherently hard to manipulate, they may not satisfy the intuitive criteria of a given setting. On the other hand, a hybrid of two protocols retains many of their desirable properties, and sometimes may combine the best of both worlds. All of the voting protocols described in Section 2, as well as many others, are used in different contexts; while it would be unreasonable to expect that all of them will be replaced, say, by STV just because it is harder to manipulate, hybrids of these protocols with similar ones or even with themselves may be eventually preferred to the original protocols. Moreover, our results on utility-based voting suggest that our techniques can be useful for a wider class of problems and can be viewed as a contribution to the more general task of constructing computationally strategy-proof mechanisms.

While we proved that many specific hybrid protocols are hard to manipulate (though some are not), our goal is not to give a complete list of such protocols, or investigate all possible protocol combinations; indeed, given the variety of voting algorithms used in practice, this task seems infeasible. Rather, our work should be viewed as a step towards understanding what makes protocols hard to manipulate, and whether a protocol at hand can be modified to have this property. We believe that the conditions we suggest in our hardness reductions apply in many cases not mentioned in the paper; simplifying these conditions, or replacing them with necessary and sufficient criteria is an interesting open problem.

Another important issue not addressed in this paper is that of designing efficient protocols with high average-case manipulation complexity. However, even asking this question properly, i.e., coming up with a natural distribution of voter's preferences with respect to which the average-case hardness is computed is itself a difficult task: clearly, in most scenarios one cannot expect preferences to be uniformly distributed. Initial results in this direction can be found in [6]; however, this topic should be explored further.

Acknowledgements

Part of this work was done when the first author was visiting Helsinki University of Technology. The authors were partially supported by the Finnish Defence Forces Institute for Technological Research; also, the first author was supported by the EPSRC grant GR/T07343/01 "Algorithmics of Network-sharing Games".

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 and T. Sandholm. Complexity of Manipulating Elections with Few Candidates. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*, pages 314–319, Edmonton, Alberta, Canada, July 28 — August 1 2002. AAAI Press.

[4] V. Conitzer and T. Sandholm. Universal Voting Protocol Tweaks to Make Manipulation Hard. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 781–788, Acapulco, Mexico, August 9–15 2003.

[5] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, Rank aggregation methods for the Web, In *Proc. 10th International World-Wide Web Conference (WWW)*, pages 613–622.

[6] E. Elkind and H. Lipmaa. Small Coalitions Cannot Manipulate Voting. In *Proceedings of Financial Cryptography and Data Security - Ninth International Conference*, Roseau, The Commonwealth Of Dominica, February 28–March 3, 2005.

[7] E. Ephrati and J. S. Rosenschein. Multi-Agent Planning as a Dynamic Search for Social Consensus. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1993.

[8] A. F. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41:597–601, 1973.

[9] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, E. Vee. Comparing and Aggregating Rankings with Ties. In *Proceedings of 23rd ACM Symposium on Principles of Database Systems (PODS)*, pages 47–58.

[10] D. M. Pennock, E. Horvitz, and C. Lee Giles. Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, July 2000.

[11] M. A. Satterthwaite. *The Existence of Strategy-Proof Voting Procedures: A Topic in Social Choice Theory*. PhD thesis, University of Wisconsin, Madison, 1973.

Edith Elkind
Department of Computer Science
University of Liverpool
Liverpool, UK

Helger Lipmaa
UCL Department of Computer Science
UCL Adastral Park Campus
Ipswich, UK

The Complexity of Bribery in Elections[1]

Piotr Faliszewski, Edith Hemaspaandra, and
Lane A. Hemaspaandra

Abstract

We study the complexity of influencing elections through bribery: How computationally complex is it for an external actor to determine whether by a certain amount of bribing voters a specified candidate can be made the election's winner? We study this problem for election systems as varied as scoring protocols and Dodgson voting, and in a variety of settings regarding the nature of the voters, the size of the candidate set, and the specification of the input. We obtain both polynomial-time bribery algorithms and proofs of the intractability of bribery. Our results indicate that the complexity of bribery is extremely sensitive to the setting. For example, we find settings where bribing weighted voters is NP-complete in general but if weights are represented in unary then the bribery problem is in P. We provide a complete classification of the complexity of bribery for the broad class of elections (including plurality, Borda, k-approval, and veto) known as scoring protocols.

1 Introduction

This paper studies the complexity of bribery in elections, that is, the complexity of computing whether it is possible, by modifying the preferences of a given number of voters, to make some preferred candidate a winner. Recall that an election system provides a framework for aggregating voters' preferences—ideally (though there is no truly ideal voting system [DS00, Gib73, Sat75]) in a way that is satisfying, attractive, and natural. Societies use elections to select their leaders, establish their laws, and decide their policies. However, practical applications of elections are not restricted to people and politics. Many parallel algorithms start by electing leaders; multiagent systems sometimes use voting for the purpose of planning [ER93]; web search engines can aggregate results using methods based on elections [DKNS01].

With such a range of applications, it is not surprising that elections may have a wide range of voter-to-candidate proportions. For example, in typical presidential elections there are relatively few candidates but there may be millions of voters. In the context of the web, one may consider web pages as voting on other pages by linking to them, or may consider humans to be voting on pages at a site by the time they spend on each. In such a setting we may have both a large number of voters and a large number of candidates. On the other

[1] A preliminary version of this paper appeared in the Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06); a full version is also available [FHH06]. Supported in part by NSF grants CCR-0311021 and CCF-0426761, a Friedrich Wilhelm Bessel Research Award, and the Alexander von Humboldt Foundation's TransCoop program.

hand, Dwork et al. [DKNS01] suggest designing a meta search engine that treats other search engines as voters and web pages as candidates. This yields very few voters but many candidates.

Typically, we are used to the idea that each vote is equally important. However, all the above scenarios make just as much sense in a setting in which each voter has a different voting power. For example, U.S. presidential elections are in some sense weighted (different states have different voting powers in the Electoral College); shareholders in a company have votes weighted by the number of shares they own; and search engines in the above example could be weighted by their quality. Weighted voting is a natural choice in many other settings as well.

The importance of election systems naturally inspired questions regarding their resistance to abuse, and several potential dangers were identified and studied. For example, the organizers can make attempts to *control* the outcome of the elections by procedural tricks such as adding or deleting candidates or encouraging/discouraging people from voting. Classical social choice theory is concerned with the possibility or impossibility of such procedural control. However, recently it was realized that even if control is possible, it may still be difficult to find what actions are needed to effect control, e.g., because the computational problem is NP-complete. The complexity of controlling who wins the election was first studied by Bartholdi, Tovey, and Trick [BTT92].

Elections are endangered not only by the organizers but also by the voters (*manipulation*), who might be tempted to vote strategically (that is, not according to their true preferences) to obtain their preferred outcome. This is not desirable as it can skew the result of the elections in a way that is arguably not in the best interest of the society. The Gibbard–Satterthwaite/Duggan–Schwartz Theorems [Gib73, Sat75, DS00] show that essentially all election systems can be manipulated. So it is important to discover for which systems manipulation is *computationally difficult* to execute. This line of research was started by Bartholdi, Tovey, and Trick [BTT89a], and was continued by many researchers, e.g, [CS02a, CS02b, CS03, CLS03, EL05, HH05].

Surprisingly, nobody seems to have addressed the issue of (the complexity of) bribery, i.e., attacks where the person interested in the success of a particular candidate picks a group of voters and convinces them to vote as he or she says. Bribery seems strongly motivated from both real life and from computational agent-based settings, and shares some of the flavor of both manipulation (changing voters' (reported) preferences) and control (deciding which voters to influence). This paper initiates the study of the complexity of bribery in elections.

There are many different settings in which bribery can be studied. In the simplest one we are interested only in the least number of voters we need to bribe to make our favored candidate win. A natural extension is to consider prices for each voter. In this setting, voters are willing to change their true preferences to anything we say, but only if we can meet their price. In an even more complicated setting it is conceivable that voters would have different

prices depending on how we want to affect their vote (however, it is not clear how to succinctly encode a voter's price scheme). We study only the previous two scenarios.

We classify election systems with respect to bribery by in each case seeking to either prove the complexity is low by giving a polynomial-time algorithm or argue intractability via proving the NP-completeness of discovering whether bribery can affect a given case. We obtain a broad range of results showing that the complexity of bribery depends closely on the setting. For example, for weighted plurality elections, bribery is in P but jumps to being NP-complete if the weighted voters have price tags as well. As another example, for approval voting the manipulation problem is easily seen to be in P, but in contrast we prove that the bribery problem is NP-complete. Yet we also prove that when the bribery cost function is made more local the complexity of approval voting falls back to P. For scoring protocols we obtain a full classification of the complexity of bribery, for all the settings that we consider.

The paper is organized as follows. In the preliminary section we describe the election systems and bribery problems we are interested in. Then we provide a detailed study of bribery in plurality elections. After that we study connections between manipulation and bribery, and fully classify bribery under scoring protocols. Finally, we study the case of succinctly represented elections. Due to space limits, the proofs are omitted except for some brief sketches. All details can be found in the full version [FHH06].

2 Preliminaries

We can describe elections by providing a set $C = \{c_1, \ldots, c_m\}$ of candidates, a set V of voters specified by their preferences, and a rule for selecting winners. A voter v's preferences are represented as a list $c_{i_1} > c_{i_2} > \ldots > c_{i_m}$, $\{i_1, i_2, \ldots, i_m\} = \{1, 2, \ldots, m\}$, where c_{i_1} is the most preferred candidate and c_{i_m} is the most despised one. We assume that preferences are transitive, complete (for every two candidates each voter knows which one he or she prefers), and strict.

Let us briefly describe the election systems that we analyze in this paper.[2] Winners of *plurality* elections are the candidate(s) who are the top choice of the largest number of voters (of course, these will be different voters for different winners). In *approval* voting each voter selects candidates he or she approves of; the candidate(s) with the most approvals win.

A *scoring protocol* for m candidates is described by a vector $\alpha = (\alpha_1, \ldots, \alpha_m)$ of nonnegative integers such that $\alpha_1 \geq \alpha_2 \ldots \geq \alpha_m$. (We have not required

[2]In the social choice literature, often voting systems are assumed to have at least one winner, or exactly one winner, but at least in terms of the notion of voting system, we do not require such a restriction, since one can imagine wanting to study elections in which—perhaps due to tie effects or symmetry effects (or even due to having zero candidates)—there is not always exactly one winner. Indeed, in practice, in such elections as those on Hall of Fame induction worthiness or on who should be hired at a given academic department, it is quite possible that a real-world election system might give the answer "No one this year."

$\alpha_1 > \alpha_m$, as we wish to classify the broadest class of cases possible, including the usually easy boundary case when all α_i's are equal.) Each time a candidate appears in the i'th position of a voter's preference list, that candidate gets α_i points; the candidate(s) who receive the most points win. Well-known examples of scoring protocols include the Borda count, plurality, k-approval, and veto voting systems, where for m-candidate elections Borda count uses $\alpha = (m - 1, m - 2, \ldots, 0)$, plurality uses $\alpha = (1, 0, \ldots, 0, 0)$, k-approval uses $(1^k, 0^{m-k})$, and veto uses $\alpha = (1, 1, \ldots, 1, 0)$.

A Condorcet winner is a candidate who (strictly) beats all other candidates in pairwise contests, that is, a Condorcet winner beats everyone else in pairwise plurality elections. Clearly, there can be at most one Condorcet winner, but sometimes there are none. There are many voting systems that choose the Condorcet winner if one exists and use some compatible rule otherwise. One such system is that of Dodgson, where a winner is the person(s) who can become a Condorcet winner by a smallest number of switches in voters' preference lists. (A switch changes the order of two adjacent candidates on a list.) If a Condorcet winner exists, he or she is the unique winner in Dodgson's scheme. See Dodgson [Dod76] for details regarding Dodgson's voting rule, under which it is known that winner testing is complete for parallel access to NP [HHR97].

Now let us define the bribery problem for a given election system $\mathcal{E}$. All numbers are nonnegative integers and, unless otherwise specified, are represented in binary. $\mathcal{E}$-bribery is the following problem.

Given: A set C of candidates, a set V of voters specified via their preference lists, distinguished candidate p, and a nonnegative integer k.

Question: Is it possible to make p a winner of the $\mathcal{E}$ election by changing the preference lists of at most k voters?

Bribery problems come in several different flavors. In the unweighted case, the default for this paper, all voters are equal; in the weighted case each voter has a possibly different weight. In the $\mathcal{E}$-\$bribery family of problems we assume that each voter has a price for changing his or her preference list. In such a case we ask not whether we can bribe at most k people, but whether we can make p a winner by spending at most k dollars on bribing. Naturally, we also consider \$bribery problems with weighted voters.

Regarding the fact that in these models voters are assumed to vote as the bribes dictate, we stress that by using the term bribery, we do not intend to imply any moral failure on the part of bribe recipients: Bribes are simply payments.

Formally, our bribery problems speak of making the preferred candidate a winner rather than making him or her the unique winner. However, essentially all our results hold for the unique winner cases as well.

As always, we say $A \leq_m^p B$ (A many-one polynomial-time reduces to B) if there is a polynomial-time computable function f such that $x \in A \iff f(x) \in B$. We also use disjunctive truth-table reductions: $A \leq_{dtt}^p B$ (A disjunctively truth-table reduces to B) if there is a polynomial-time procedure that on input

x outputs a list of strings such that $x \in A$ if and only if at least one of those strings is in B. See, e.g., the work of Ladner, Lynch, and Selman [LLS75] for details regarding various reduction types. $\|S\|$ denotes the cardinality of set S.

3 Plurality

In this section we establish the complexity of bribery for plurality rule elections. The widespread use of plurality elections makes these results of particular relevance.

Not surprisingly, plurality-bribery is easy.

Theorem 3.1 plurality-bribery *is in P.*

To make sure our favorite candidate p wins, it is enough to keep on bribing voters of the currently most popular candidate to vote for p until p becomes a winner. However, bribery within the plurality system is not always easy.

Theorem 3.2 plurality-weighted-$bribery *is NP-complete, even for just two candidates.*

That is, bribery is easy in the simplest case, but if we allow voters to have prices and weights, then the problem becomes intractable. It is natural to ask which of the additional features (prices? weights?) is responsible for making the problem difficult. It turns out that neither of them is the sole reason and that only their combination yields enough power to make the problem NP-complete.

Theorem 3.3 *Both* plurality-$bribery *and* plurality-weighted-bribery *are in P.*

A direct greedy algorithm, like that underpinning Theorem 3.1, fails to prove Theorem 3.3. Rather we approach Theorem 3.3's proof as follows. Assume that p will be capable of getting at least r votes (or in the weighted case, r vote weight), where r is some number to be specified later. If this is to make p a winner, we need to make sure that everyone else gets at most r votes. Thus we carefully choose enough cheapest (heaviest) voters of candidates that defeat p and bribe those voters to vote for p. Then we simply have to make sure that p gets at least r votes by bribing the cheapest (the heaviest) of the remaining voters. If during this process p ever becomes a winner without exceeding the budget (the bribe limit) then we know that bribery is possible. How do we pick the value of r? In the case of plurality-$bribery, we can just run this procedure for all $\|V\|$ possible values, and accept exactly if it succeeds for at least one of them. For plurality-weighted-bribery a slightly trickier approach works. (Essentially, we need to try only $\|V\|$ values as well; we can start from $r = 1$ and always increase r so that minimally less people need to be bribed in the first part of the above algorithm.)

Pushing the approach outlined above even further it is in fact possible to get yet stronger results. In plurality-weighted-$bribery we assume that both

prices and weights are encoded in binary. However, if either the prices or the weights are encoded in unary, then the problem becomes easy.

Theorem 3.4 *Both* plurality-weighted-$bribery$_{\text{unary}}$ *and* plurality-weighted$_{\text{unary}}$-$bribery *are in P.*

The main idea of the proof of Theorem 3.4 is the same as that underpinning the proof of Theorem 3.3, but the details are more complicated. Theorem 3.4 is particularly interesting because it says that plurality-weighted-$bribery will be difficult only if we choose both weights and bribe prices to be high. But the prices are set by voters, and in many cases one could assume that there would be fairly low values for the bribe prices and so the problem would be easy.

We can look at the problem of bribery within plurality elections from a yet another perspective. Note that all of the above algorithms (in most cases, implicitly) assume that we bribe others to vote for p. This is a reasonable method of bribing if one wants p to become a winner, but it also has potential real-world downsides: The more people we bribe, the more likely it may be that the malicious attempts will be detected and will work against p. To minimize the chances of that happening we might instead bribe voters not to vote for p but for some other candidates. This way p does not get extra votes but might be able to take away enough voters from the most popular candidates to become a winner. We call this setting negative-bribery. Negative bribery draws a very sharp line between the complexity of bribing weighted and priced voters.

Theorem 3.5 plurality-weighted-negative-bribery *is NP-complete, but* plurality-negative-$bribery *is in P.*

4 Bribery versus Manipulation

The previous section provides a detailed discussion of the complexity of bribery for plurality voting. Its results are obtained by hand-crafting algorithms and reductions. It would be nicer if one could find tools that would let one inherit complexity results from the vast election systems literature. In this section we study relations between bribery and manipulation, and show how to obtain results using the relations we find. In the next section we will discuss another fairly general tool to study certain types of bribery and manipulation problems.

Manipulation is in flavor somewhat similar to bribery, with the difference that in manipulation the set of voters who may change their preference lists is specified by the input. Bribery can be viewed as manipulation where the set of manipulators is not fixed in advance and finding who to manipulate is part of the challenge. This might suggest that bribery problems should not be easier than analogous manipulation ones. In fact, there are election systems for which bribery is NP-complete but manipulation is easy.

Theorem 4.1 approval-bribery *is NP-complete, but* approval-manipulation *and* approval-weighted-manipulation *are both in P.*

Algorithms for approval-manipulation and approval-weighted-manipulation are trivial: The manipulating group approves of just the favorite candidate. The NP-completeness result follows from a reduction from the NP-complete Exact-Cover-by-3Sets problem.

Somewhat surprisingly, it is also possible that manipulation is NP-complete, while bribery is in P. We have designed an artificial election system where this is the case.

Theorem 4.2 *There exists a voting system $\mathcal{E}$ for which manipulation is NP-complete, but bribery is in P.*

We briefly sketch a proof of this theorem. Let A be an NP-complete set and let $B \in \mathrm{P}$ be such that

1. $A = \{x \in \Sigma^* \mid (\exists y \in \Sigma^*)[\langle x, y\rangle \in B]\}$, and
2. $(\forall x, y \in \Sigma^*)[\langle x, y\rangle \in B \Rightarrow |x| = |y|]$.

Such sets can easily be constructed from any NP-complete set by padding. The idea of the proof is to embed a verifier for A within the election rule $\mathcal{E}$. We do this in a way that forces manipulation to solve arbitrary A instances, while allowing bribery to still be easy.

First, we observe that preference lists can be used to encode arbitrary binary strings. We will use the following encoding. For C a set of candidates, let $c_1, c_2, \ldots, c_m$ be those candidates in lexicographical order. We will view the preference list

$$c_{i_1} > c_{i_2} > c_{i_3} > \cdots > c_{i_m}$$

as an encoding of the binary string $b_1 b_2 \cdots b_{\lfloor m/2 \rfloor}$, where for each j, $1 \leq j \leq \lfloor m/2 \rfloor$, if $i_{2j-1} > i_{2j}$ then $b_j = 0$ and otherwise $b_j = 1$.

In our reduction, binary strings starting with 1 will encode instances, and binary strings starting with 0 will encode witnesses. Given this setup, we can describe our election system $\mathcal{E}$. Let (C, V) be an election. For each $c \in C$, c is a winner of the election if and only if $\|V\| = 3$ and

Rule 1: all preference lists encode strings starting with 1 or all preference lists encode strings starting with 0, or

Rule 2: exactly one preference list encodes a string that starts with 1, say $1x$, and at least one other preference list encodes a string $0y$ such that $\langle x, y\rangle \in B$.

Thus, either all candidates are winners or none of them are winners. Note that testing whether a candidate c is a winner of an $\mathcal{E}$ election can easily be done in polynomial time. $\mathcal{E}$-bribery is in P because we are in one of the following three cases: all candidates are winners already, or no candidate can become a winner because $||V|| \neq 3$, or we can make each candidate a winner by bribing exactly one voter so that all voters' preference lists encode strings that start with the same symbol.

On the other hand, the ability to solve the manipulation problem for $\mathcal{E}$ implies the ability to solve A. We construct a reduction from A to $\mathcal{E}$-manipulation. Given a string $x \in \Sigma^*$, we first check whether $\langle x, 0^{|x|} \rangle \in B$. If so, then clearly $x \in A$ and we output some fixed member of $\mathcal{E}$-manipulation. Otherwise, we output a manipulation problem with candidates $\{1, 2, \ldots, 2(|x|+1)\}$ and three voters, v_0, v_1, and v_2, such that v_0's preference list encodes $1x$, v_1's preference list encodes $00^{|x|}$, and v_2 is the only manipulative voter. We claim that candidate 1 can be made a winner if and only if $x \in A$.

Since $\langle x, 0^{|x|} \rangle \notin B$, the only way in which v_2 can make 1 a winner is when v_2 encodes a string $0y$ such that $\langle x, y \rangle \in B$ in which case $x \in A$. For the converse, if $x \in A$, there exists a string $y \in \Sigma^{|x|}$ such that $\langle x, y \rangle \in B$. We can encode string $0y$ as a preference list over $\{1, 2, \ldots, 2(|x|+1)\}$, and let this be the preference list for v_2. This ensures that 1 is a winner of the election.

Since this reduction can be computed in polynomial time, and the $\mathcal{E}$-manipulation's membership in NP is clear, we have that $\mathcal{E}$-manipulation is NP-complete.

While the above election system is not natural, together with the results on approval voting it tells us that we cannot hope to get a result that says "manipulation always reduces to an analogous bribery problem," unless P = NP. Nonetheless, if instead of looking at all election systems and all versions of bribery and manipulation we somewhat restrict our focus, either to some subclass of election systems or to some subclass of bribery types, then we can still find interesting connections between the complexity of bribery and the complexity of manipulation.

First, let us observe that to check whether bribery can be successful on a given input we can simply try all possible manipulations by k voters, where k is the number of bribes we are willing to make. This way for a fixed k we can disjunctively truth-table reduce any bribery problem to the analogous manipulation problem. In the following meta-theorem, $\mathcal{B}$ means any of our bribery problems that does not involve prices, and $\mathcal{M}$ represents the analogous manipulation problem.

Theorem 4.3 *For each fixed k it holds that $\mathcal{B} \leq^p_{dtt} \mathcal{M}$, where the bribery problem $\mathcal{B}$ allows at most k bribes, and the manipulation problem $\mathcal{M}$ allows the manipulator set to contain any number of voters between 0 and k.*

While simple, this result is still powerful enough to inherit some results from previous papers. Bartholdi, Tovey, and Trick [BTT89a] discuss manipulations by single voters. Theorem 4.3 translates their results to the bribery case. In particular, this translation says that bribery for $k = 1$ is in P for plurality, Borda count and many other systems.

Instead of looking at bribery problems that restrict the number of voters we can affect, we can focus on a restricted set of election systems. Namely, we will concentrate on a very natural and broad family, the scoring protocols.

Hemaspaandra and Hemaspaandra [HH05] showed a dichotomy theorem that classifies weighted manipulation problems for all scoring protocols as either

being NP-complete or in P (see also [PR06] and the unpublished 2005 combined version of [CS02a, CLS03]). By reducing manipulation to bribery for scoring protocols (but also by employing several other insights) we have obtained an analogous classification for the case of bribery.

Theorem 4.4 *For each scoring protocol $\alpha = (\alpha_1, \ldots, \alpha_m)$ Table 1 shows the complexity of each of the five natural bribery problems for that scoring protocol.*

We will now briefly sketch how Table 1 was obtained. First, let us note that for each scoring protocol $\alpha = (\alpha_1, \ldots, \alpha_m)$ such that $\alpha_1 = \cdots = \alpha_m$ any bribery problem is in P; in this case each preference list has the same effect on elections. For $\alpha_1 > \alpha_2 = \cdots = \alpha_m$ it is clear that each appropriate bribery problem is a special case of an analogous bribery problem for plurality. (Only the NP-completeness result needs some care, but it can be translated to the scoring protocol world as well.) Thus, the interesting part of the table is that where it is not the case that $\alpha_2 = \cdots = \alpha_m$.

Each time we consider some scoring protocol $\alpha = (\alpha_1, \ldots, \alpha_m)$ we automatically limit ourselves to a setting with a fixed constant number of candidates, namely m. Thus, there are only $m!$ different preference orders each voter might have, and so it is possible to evaluate all possible ways of bribing unweighted voters. This gives us that both α-bribery and α-\$bribery are in P for any α. Using a similar in spirit, but somewhat more involved, dynamic-programming algorithm we can also show that α-weighted$_{\text{unary}}$-\$bribery is in P.

It remains to show that α-weighted-bribery and α-weighted-\$bribery are NP-complete when it is not the case that $\alpha_2 = \cdots = \alpha_m$. In the case of α-weighted-\$bribery this is fairly easy as we simply need to observe that manipulation is in fact just a special case of \$bribery (Theorem 4.5) and invoke the Hemaspaandra and Hemaspaandra dichotomy theorem.

Theorem 4.5 *Let $\mathcal{M}$ be some manipulation problem and let $\mathcal{B}$ be the analogous \$bribery problem (for the same election system). It holds that $\mathcal{M} \leq_m^p \mathcal{B}$.*

The reduction simply takes an instance of the manipulation problem and outputs a bribery problem that is identical to the manipulation one only that all the nonmanipulators have price 1, all manipulators have price 0 and (just for specificity) some fixed preferences, and the budget is set to 0.

It remains to show that for scoring protocols α such that it is not the case that $\alpha_2 = \cdots = \alpha_m$ the α-weighted-bribery is still NP-complete, even without the use of price tags. It would be nice to do so by reducing to our problem from the corresponding manipulation problems. This seems not to work, but we construct such a reduction that has the right properties whenever its inputs satisfy an additional condition (namely, that the weight of the lightest manipulating voter is at least double that of the heaviest nonmanipulator). So we would be done if this restriction of the manipulation problem were NP-hard. To show that, we by close examination of the dichotomy proof of Hemaspaandra and Hemaspaandra [HH05] prove that though the reduction from partition to that manipulation problem does not obey the desired condition in all its image

bribery problem	Scoring protocol $\alpha = (\alpha_1, \ldots, \alpha_m)$. $\alpha_1 = \cdots = \alpha_m$	$\alpha_1 > \alpha_2$ and $\alpha_2 = \cdots = \alpha_m$	not true that $\alpha_2 = \cdots = \alpha_m$
α-bribery	P	P	P
α-\$bribery	P	P	P
α-weighted$_{\text{unary}}$-\$bribery	P	P	P
α-weighted-bribery	P	P	NP-complete
α-weighted-\$bribery	P	NP-complete	NP-complete

Table 1: The complexity of bribery within scoring protocols.

elements, if we look at the image of only a certain restriction of the partition problem we can modify the thus-obtained elections to obey the desired conditions. Finally, we show that the restricted partition problem used above is NP-hard. This completes the sketch of the proof of Theorem 4.4.

In the beginning of this section we noted that approval voting is an example of an election system where bribery is NP-complete, whereas manipulation is easy. We mention that bribery in approval elections is actually very easy, provided that one looks at a slightly different model. Our bribery problems allow us to completely modify the approval vector of a voter. This may, however, be too demanding since a voter might be willing to change some of his or her approval vector's entries but not to completely change his or her approval vector. In particular, in the approval-bribery$'$ problem we will ask whether it is possible to make our favorite candidate p a winner by at most k entry changes in the approval vectors. We also define the weighted and priced versions of approval-bribery$'$ in the natural way.

Theorem 4.6 approval-bribery$'$, approval-\$bribery$'$, approval-weighted$_{\text{unary}}$-\$bribery$'$ *and* approval-weighted-\$bribery$'_{\text{unary}}$ *are in P.* approval-weighted-\$bribery$'$ *is NP-complete.*

Which of the bribery models for approval is more practical depends on the setting. For example, bribery$'$ seems more natural when we look at the web and treat web pages as voting by linking to other pages. It certainly is easier to ask a webmaster to add/remove a link than to completely redesign the page.

5 Succinct Elections

So far we have discussed only nonsuccinct elections—ones where voters with the same preference lists (and weights, if voters are weighted) are given by listing them one at a time (as if given a stack of ballots). It is also very natural to consider the case where each preference list has its frequency conveyed via a count (in binary), and we will refer to this as "succinct" input. Succinct in curly braces within a name of a bribery problem will describe the fact that it holds in both cases, e.g., if we say that plurality-{succinct}-bribery is in P, we mean that both plurality-bribery and plurality-succinct-bribery are in P. (By

the way, Theorem 3.1, by a similar but more careful algorithm than the one mentioned right after it, also holds for the succinct case.)

In this section we provide P membership results regarding succinctly represented elections with a fixed number of candidates. (Such results for the case of succinct representation immediately yield results for the nonsuccinct case.) The most useful tool here is Lenstra's [Len83] extremely powerful result that the integer programming feasibility problem is in P when the number of variables is bounded. Lenstra's algorithm has a very large constant factor in its running time, but what we are after are P-membership results and tools for obtaining them and not actual optimized algorithms.

Using the integer programming approach we obtain polynomial-time algorithms for bribery under scoring protocols in both the succinct and the nonsuccinct cases. The same approach yields a similar result for manipulation. (The nonsuccinct case for manipulation was already obtained by Conitzer and Sandholm [CS02a].)

Theorem 5.1 *For every scoring protocol* $\alpha = (\alpha_1, \ldots, \alpha_m)$*, both* α-{succinct}-bribery *and* α-{succinct}-manipulation *are in P.*

The power of the integer programming approach is not limited to the case of scoring protocols. In fact, the seminal paper of Bartholdi, Tovey, and Trick [BTT89b] shows that applying this method to computing the Dodgson score in nonsuccinct elections with a fixed number of candidates yields a polynomial-time score algorithm (and though they did not address the issue of succinct elections, one can see that there too this method works perfectly). Applying an integer programming attack for the case of bribery is a bit more complicated, since one has both the issue of the bribes and the issue of the exchanges involved in computing the Dodgson scores. But even in this setting one can represent the question as an integer programming feasibility problem, and thus via Lenstra's algorithm we have the following result.

Theorem 5.2 *For each fixed number of candidates,* DodgsonScore-{succinct}-bribery *is in P when restricted to that number of candidates.*

By this we mean that in polynomial time we can test if a given bribe suffices to obtain or beat a given Dodgson score for our favored candidate (the Dodgson score of candidate c is the number of switches needed to be done in voters' preference lists to make c the Condorcet winner). Using binary search we may compute the minimum bribe needed to make our favored candidate have a given Dodgson score.

In Young elections ([You77]; see also [RSV03], which proves that the winner problem in Young elections is complete for parallel access to NP) the score of a candidate is the number of voters that need to be removed to make that candidate a Condorcet winner.

Theorem 5.3 *For each fixed number of candidates,* YoungScore-{succinct}-bribery *is in P when restricted to that number of candidates.*

The issue of actually making a candidate p a winner (a unique winner, if we are studying the unique winner case) of Dodgson elections is, as already indicated, much more difficult and a direct attack using integer linear programming seems to fail. Nonetheless, combining the integer programming method with a brute-force algorithm resolves the issue for the nonsuccinct case.

Theorem 5.4 *For each fixed number of candidates,* Dodgson-bribery*,* Dodgson-$bribery*,* Young-bribery*, and* Young-$bribery *are all in P.*

On the other hand, using integer programming, we obtain polynomial-time algorithms for bribery in Kemeny elections in both the succinct and nonsuccinct cases.

Theorem 5.5 *For each fixed number of candidates,* Kemeny-{succinct}-bribery *is in P when restricted to that number of candidates.*

In brief, Kemeny's system elects each candidate who is most preferred in at least one preference order that maximizes the number of agreements with the voters' preferences, where for two candidates, a and b, two preference orders agree if they both place a ahead of b or both place b ahead of a.

6 Research Directions

This paper provides a detailed study of the complexity of bribery with respect to plurality rule and, more generally, scoring protocols. This paper also provides tools and results regarding many other election systems such as approval voting and Dodgson elections.

There are several directions in which further research on bribery might go. The most obvious one is to study the complexity of bribery for other election systems. Another very interesting route is to study approximation algorithms for $bribery problems. It would also be interesting to study the complexity of bribery in other settings, such as with incomplete information, multiple competing bribers, or more complicated bribe structures.

Acknowledgments: We are very grateful to Samir Khuller for helpful conversations about the Bartholdi, Tovey, and Trick integer programming attack on fixed-candidate Dodgson elections, and we thank the anonymous AAAI-06 and COMSOC-06 referees for helpful comments.

References

[BTT89a] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT89b] 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(2):157–165, 1989.

[BTT92] 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.

[CLS03] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 201–214. ACM Press, July 2003.

[CS02a] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 314–319. AAAI Press, July/August 2002.

[CS02b] V. Conitzer and T. Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 392–397. AAAI Press, July/August 2002.

[CS03] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 781–788. Morgan Kaufmann, August 2003.

[DKNS01] 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, March 2001.

[Dod76] C. Dodgson. A method of taking votes on more than two issues. Pamphlet printed by the Clarendon Press, Oxford, and headed "not yet published", 1876.

[DS00] J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

[EL05] E. Elkind and H. Lipmaa. Small coalitions cannot manipulate voting. In *Proceedings of the 9th International Conference on Financial Cryptography and Data Security*, pages 285–297. Springer-Verlag *Lecture Notes in Computer Science #3570*, 2005.

[ER93] E. Ephrati and J. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 423–429. Morgan Kaufmann, 1993.

[FHH06] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? Technical Report TR-895, Department of Computer Science, University of Rochester, Rochester, NY, April 2006. Revised, September 2006.

[Gib73] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[HH05] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. Technical Report TR-861, Department of Computer Science, University of Rochester, Rochester, NY, April 2005. Journal version to appear in *Journal of Computer and System Sciences*.

[HHR97] 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.

[Len83] H. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.

[PR06] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 497–504. ACM Press, May 2006.

[RSV03] 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.

[Sat75] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[You77] H. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16(2):335–353, 1977.

Piotr Faliszewski
Department of Computer Science, University of Rochester
Rochester, NY 14627 USA, www.cs.rochester.edu/u/pfali

Edith Hemaspaandra
Department of Computer Science, Rochester Institute of Technology
Rochester, NY 14623 USA, www.cs.rit.edu/∼eh

Lane A. Hemaspaandra
Department of Computer Science, University of Rochester
Rochester, NY 14627 USA, www.cs.rochester.edu/u/lane

Optimizing Streaming Applications with Self-Interested Users using M-DPOP

Boi Faltings*, David Parkes†, Adrian Petcu‡, Jeff Shneidman§

Abstract

In this paper we deal with the problem of optimally placing a set of query operators in an overlay network. Each user is interested in performing a query on streaming data and each query has an associated set of in-network operators that filter, aggregate and process the data in various ways. Each user has private information about the operators associated with a query and about the utility from different combinations of operator placements. Each server in the overlay network is able to perform some set of operators, and servers differ in their network and computational characteristics.

We model this problem as a Distributed Constraint Optimization Problem (DCOP), and apply the M-DPOP algorithm from Petcu et al. [19], executed here by clients associated with users and situated at nodes on the overlay network. M-DPOP makes truth-telling an *ex-post Nash equilibrium* and determines the social-welfare maximizing placement of operators to servers. No client can benefit by deviating from the M-DPOP algorithm and nodes need only communicate with other nodes that have an interest in placing an operator on the same server. The only central authority required is a bank that can extract payments from users. Preliminary results from simulation show that message size will be a bottleneck in applying M-DPOP to operator placement unless structure can be enforced and then exploited.

1 Motivation

Recently, there has been interest in building long-lived data streaming applications on the Internet. These applications typically involve querying, processing, and delivering real-time data from multiple distributed data sources, such as sensor networks, and making use of shared resources in the Internet to aggregate, filter, or multicast this data. Commonly-cited target applications include continuous monitoring of Internet paths and system loads [8], and querying geographically diverse data sources [20].

These streaming applications can run on overlay networks that consist of nodes that are capable of performing in-network processing. A few examples of such overlays are IrisNet [7], PIER [8], Borealis [1], and SBON [20]. In these networks, *queries* are submitted by users who wish to receive data from

*boi.faltings@epfl.ch, EPFL, CH-1015 Lausanne
†parkes@eecs.harvard.edu, Harvard University, DEAS, Cambridge MA 02138, USA
‡adrian.petcu@epfl.ch, EPFL, CH-1015 Lausanne
§jeffsh@eecs.harvard.edu, Harvard University, DEAS, Cambridge MA 02138, USA

producers via one or more in-network operators. Examples of in-network operators include database style "join" operators, or custom logic provided by an end user.

One of the fundamental questions in these overlay networks is how to perform *query placement*, which is the problem of mapping the *operators* in a particular query to a collection of nodes (the *servers*) that will run those operators. Placement can be formulated as a complex constrained optimization problem; placements must be done subject to load and bandwidth node limitations (where important), quality of service stream requirements, and done in a computationally efficient way.

In practice, users have varying levels of happiness, or *utility*, for how their queries are placed and executed. For example, a user with a jitter-sensitive query should avoid operator placement on nodes with high variance in communication latency. On the other hand, a user with a high volume query may not care about latency but wants operators placed on nodes with high data rates.

The goal of our research is to provide a distributed constrained optimization algorithm that allows users to influence where their query operators are placed, but seeks to maximize the aggregate utility across all users. This approach is unique in that it is the first overlay query placement algorithm that uses ideas from computational mechanism design to compute a value-maximizing placement in the presence of self-interested nodes. Our solution leverages M-DPOP [19], which provides a distributed algorithm for social choice problems, and can exploit problem structure to scale well to large problems. M-DPOP is a *faithful* distributed implementation [21], in the sense that even nodes controlled by users (and thus open to manipulation) will choose to follow the algorithm because this maximizes, in equilibrium, their individual self-interest. Preliminary results from simulation show that message size will be a bottleneck in applying M-DPOP to operator placement unless structure can be enforced and then exploited.

2 Problem Statement

Each user is associated with a query and has a client located at a particular node on the overlay network. Each query has an associated set of data producers, known to the user and located at nodes on the network. Each query also requires a set of operators, to be placed on (server) nodes between the producers and the user's node. Each user assigns values (or *utilities*) to various allocations of operators to servers. This preference information is private, and users are assumed self-interested and seek to maximize their individual utility.

We seek a distributed algorithm, to be executed by user clients situated on network nodes, that will determine the allocation and also payments to be made by each user for the outcome. The server nodes (i.e. the nodes that finally execute the operators and respond to queries) are assumed to "opt-in" in that

they will implement whatever allocation is determined by users. Constraints on server nodes, e.g. based on maximal load, are commonly known to users and thus server nodes are represented in the decision procedure. However, server nodes play no active role in the algorithm.

More formally, and adopting the term "agent" to represent a user and "utility" to describe user values, we have:

- agents $A = \{A_1, \ldots, A_n\}$, each with a query
- server nodes $B = \{B_1, \ldots, B_m\} \cup \phi$ where ϕ is the "null" node (corresponds to *operator not assigned*)
- each agent A_i has k_i operators $G_i = \{g_1, \ldots, g_{k_i}\}$ and each must be assigned to one node in the network (perhaps the null node, i.e. *not assigned*)
- each agent determines a possibility set $P(A_i, g) \subseteq B$ for each operator $g \in G_i$, which is the set of nodes for which the agent *could* have non-zero utility for placement (i.e. in some combination with other placements)
- each agent has a utility on allocations, with $u_i(x) \in \mathbb{R}$, where $x : \{G_1, \ldots, G_n\} \rightarrow B$ defines an allocation of operators to the network. (The utility is $-\infty$ if an operator is allocated to a node outside of the possibility set.)
- constraints to restrict the set of feasible allocations due to load and bandwidth considerations.

We assume there is no collusion between users and that each user controls only one client, namely its own client, and does not control any other functionality on any other nodes.

Each operator is associated with an input data stream and an output data stream. For instance, operators that process raw data received from one or more producers associated with a query define the set of producers. Thus, information about producers is implicit in the set of operators G_i associated with a query. Server nodes may be capable of running query operators on behalf of multiple concurrent queries, even allowing results of a particular operator to be re-used and shared across multiple relevant queries [20]; e.g. when the same aggregation from two producers is required by two different queries. Operator semantics must be rich enough to allow these synergies to be identified. Notice that the *null* node allows for the operator placement to decide to block one or more queries completely, or drop some of the operators in a query completely, when this is in the joint interest of all users.

Users can determine a *possibility set* of nodes for the placement of each operator. This excludes only those nodes for which the user can be absolutely certain that there is no value (to the user) for placing the operator on that

node.[5] We assume that user clients can communicate without interference with user clients representing queries that share a possible interest in a server. Each user has a utility function, which describes her value for different assignments of operators to nodes. We shall assume that a user's utility encodes bandwidth and latency considerations, and furthermore, a user is able to access enough information to determine this utility information. For instance, previous work has suggested the efficacy of various techniques to perform bandwidth and latency [22, 20] estimation between pairs of nodes [10].

The main constraint in placing the operators is that each node can only handle a limited number of operators, due to CPU and bandwidth limitations.

3 Background and Related Work

This work draws from 3 different research areas: distributed stream placement optimization, distributed constraint optimization, and faithful implementations of social choice functions.

3.1 Distributed Stream Placement Optimization

Distributed stream processing systems (DSPS) need to find a good placement for the in-network operators required by user queries. Some systems relegate this placement task back to the user. Others perform automated optimization for a hard-coded node or network metric. For instance, Borealis [1] and GATES [4] require the user to specify the initial operator locations. This forces the user to perform any optimization off-line before specifying locations. Other DSPSs, such as Medusa [3], place operators to improve application performance by balancing node load. Still other DSPs effectively randomize placement, as in PIER [8]. Neither placement strategy may be appropriate for jitter or latency-sensitive queries. SAND [2], an extension to Borealis, performs network-aware operator placement to minimize bandwidth of a query. SAND also allows applications to specify delay constraints on the query, which affects the placement decision. SBON [20] is also a system that performs network-aware operator placement, minimizing a network usage metric.

None of these previous works have taken a value-maximization (or preference-based) approach in choosing where to place operators. Rather, the system assumed that all queries were equally important, and in many cases, that queries were only concerned with latency or node load.

3.2 Distributed Constraint Optimization Framework

Distributed Constraint Optimization (DCOP), e.g. Modi et al. [11], can model social choice problems where a set of self interested agents with private utility

[5] A more advanced implementation would allow a user to state general utility functions for types of nodes – such as $u(bandwidth)$, $u(latency)$, and rely on the network to calculate the candidate set and derive the utility of each node.

functions have to agree on a set of decisions (see Petcu et al. [19]). Each decision is modeled as a variable that can take values in a well-defined domain, subject to side constraints. The goal is to maximize the total utility across all agents. Among many algorithms for this type of problems, we mention ADOPT ([11]) and DPOP ([15]). We define the general DCOP framework here. Later, in Section 4 we will instantiate operator placement in the DCOP formalism.

Definition 1 (DCOP) *A distributed constraint optimization problem (DCOP) is a tuple* $< \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{R} >$ *such that:*

$\mathcal{A} = \{A_1, ..., A_n\}$ *is a set of* **self-interested** *agents interested in the optimization problem;*

$\mathcal{X} = \{X_1, ..., X_m\}$ *is the set of* **public** *decision variables;* $X(A_i)$ *are the variables in which agent* A_i *is interested and* ***does*** *have relations. Agents* A_i *for which* $X_j \in X(A_i)$ *for some variable* X_j *form the* community *for variable* X_j*, denoted* $A(X_j) \subseteq \mathcal{A}$*.*

$\mathcal{D} = \{d_1, ..., d_m\}$ *is the set of finite* **public** *domains of the variables* $\mathcal{X}$*; each domain* d_i *is known to all interested agents (i.e. agents* $A_j s.t. X_i \in X(A_j)$*;*

$\mathcal{C} = \{c_1, ..., c_q\}$ *is a set of* **public** *constraints, where a constraint* c_i *is a function* $c_i : d_{i_1} \times .. \times d_{i_k} \rightarrow \{-\infty, 0\}$ *that returns 0 for all allowed combinations of values of the involved variables, and* $-\infty$ *for disallowed ones; these constraints are known and agreed upon by all agents involved in the respective communities;*

$\mathcal{R} = \{R_1, ..., R_n\}$ *is a set of* **private** *relations, where* R_i *is the set of relations specified by agent* A_i *and relation* $r_i^j \in R_i$ *is a function* $d_{j_1} \times .. \times d_{j_k} \rightarrow \mathbb{R}$ *specified by agent* A_i*, which denotes the utility* A_i *receives for all possible values on the involved variables* $\{j_1, \ldots, j_k\}$ *(negative values can be thought of as costs). An agent's utility for a complete assignment of values to variables is defined by the sum of its relations.*

The optimal solution is a complete instantiation X^* *of all variables in* $\mathcal{X}$*, s.t.* $X^* = argmax_{X \in \mathcal{D}}(\sum_{R_i \in \mathcal{R}} R_i(X) + \sum_{c_i \in \mathcal{C}} c_i(X))$,[6] *where* $R_i(X) = \sum_{r_i^j \in R_i} r_i^j(X)$ *is* A_i*'s utility for this solution.*

Later, we use DCOP($-A_i$) to denote the constraint optimization problem without agent A_i, and refer to this as the "marginal problem without agent A_i."

In addition to private relations on public variables, DCOP allows an agent to have *private variables* and arbitrary relations and constraints imposed on subsets of private variables and public variables. Decisions about private variables, as well as explicit information about these relations and constraints remain private to an agent.

In our context, variables will be associated with each instance of an operator, and domains with the servers that are of possible interest to the user associated

[6]Notice that the second sum is either $-\infty$ if X is an infeasible assignment, or 0 if it is feasible. Thus, optimal solution X^* will always satisfy all hard constraints when that is possible.

with the operator. Relations provide a method to express factored utilities, with the utility for an allocation decomposed into an aggregate over utilities for server assignments on groups of operators that are inputs and outputs to each other.

3.3 M-DPOP: faithful utilitarian social choice

Petcu et al. [19] have proposed *M-DPOP*, a distributed optimization protocol that *faithfully* implements (in the sense of Shneidman and Parkes [21]) the Vickrey-Clarke-Groves (VCG) mechanism ([6]) for the problem of utilitarian social choice. No agent can benefit by unilaterally deviating from any aspect of the protocol, neither information-revelation, computation, nor communication. Additionally, M-DPOP provides a faithful method to redistribute some of the VCG payments back to agents (weak budget-balance). The optimization algorithm itself is based on *DPOP* ([15]), which is a dynamic programming algorithm adapted for distributed constraint optimization problems. Agents need only communicate with other agents that have an interest in the same variable, and provided that DPOP scales then the entire method of M-DPOP scales.

Briefly, M-DPOP has the following phases:

1. *Community formation and DFS creation:* the agents interested in the value of a variable X_i organize themselves in the *community* $A(X_i)$ of that variable (a community can be physically implemented as a public medium like a bulletin board, a mailing list, etc.) All agents interested in X_i subscribe to X_i's community. Each agent $A_i \in A(X_i)$ then creates its own replica of X_i, and expresses its preferences on combinations of X_i and other variables as local relations on the local copies of these variables. By doing so, each agent creates its *local optimization problem*, denoted $COP(A_i)$. Copies of the private variables are synchronized among all interested agents using equality constraints. Once the constraint graph is established, a depth-first-search (DFS) traversal is constructed starting from a randomly chosen node. This defines the control logic.

2. *Solving the main problem:* DPOP is run on the previously established DFS structure, and the optimal solution for $DCOP(\mathcal{A})$ is obtained. DPOP involves a bottom-up propagation (and aggregation) of utility information followed by a top-down propagation of assignment information.

3. *Solving each marginal problem:* DPOP is then run in parallel on each marginal problem. Computation from the main problem (i.e. residual local state) is reused for solving each marginal problem, $DCOP(-A_i)$, in a way that prevents manipulation by A_i. Finally, the VCG taxes are then computed distributedly again in a non-manipulable fashion by all agents except the one whose tax is computed, and levied by a trusted bank.

M-DPOP's complexity in terms of number of messages is always linear in the number of variables in the optimization problem. In terms of message size, the largest message sent by any agent while executing M-DPOP is $O(exp(w))$, where w the induced width of the constraint graph ([5]). Roughly, a small induced width reflects problems with limited interconnectedness between decisions.

4 DCOP Models for Optimal Operator Placement

The problem structure of an instance of DCOP($\mathcal{A}$) can be represented as a *multigraph*, with the decision variables as nodes, and (possibly) multiple relations belonging to different agents that involve the same variables, and expressing their utilities. Figure 1(a) shows an example where the variables are associated with servers, domains are combinations of operators, and agents express preferences on combinations thereof, in the form of constraints and relations on those variables.

We adopt an alternate formulation, depicted in Figure 1(b)), in which the variables are associated with individual operators and the domains are servers of possible interest for an operator.[7] In order to allow multiple agents to express preferences on the same set of variables, we require distributed models where each agent can model its own interests as an internal optimization problem ($COP(A_i)$), and interactions between agents (agreement, feasibility constraints) are modeled as inter-agent constraints.[8] This is reflected in the model of operator placement.

4.1 DPOP model for Operator Placement

4.1.1 Local optimization problem

The local optimization problem $COP(A_i)$ of agent A_i models A_i's interests and is composed of private **variables** and **relations** (see Figure 1(b) for an example). Each agent A_i creates one variable $A_i g_j$ for each one of its operators g_j. The domain of a variable $A_i g_j$ is the possibility set $P(A_i, g_j)$ for the agent-operator (A_i, g_j) to whom the variable relates. These variables are private to agents and each agent has as many variables as it has operators to assign.

Each agent has *relations* on the values assigned to its variables (blue edges in Figure 1). These relations may be factored, e.g. perhaps one operator must be placed on any of some set of nodes with particular properties (Linux, high-bandwidth, etc.) while the other two operators should be within 3 hops of each other. Intra-agent constraints (i.e. private) may constrain combinations of operator positions that are not suitable to the agent. For example, in Figure 1),

[7] A subtle incentive problem exists with the servers-as-variables model which will be explained in a longer version of this paper.

[8] Local, private variables do not show up in inter-agent communication and agents typically need not solve the internal problem for all combinations of values of the public variables [23].

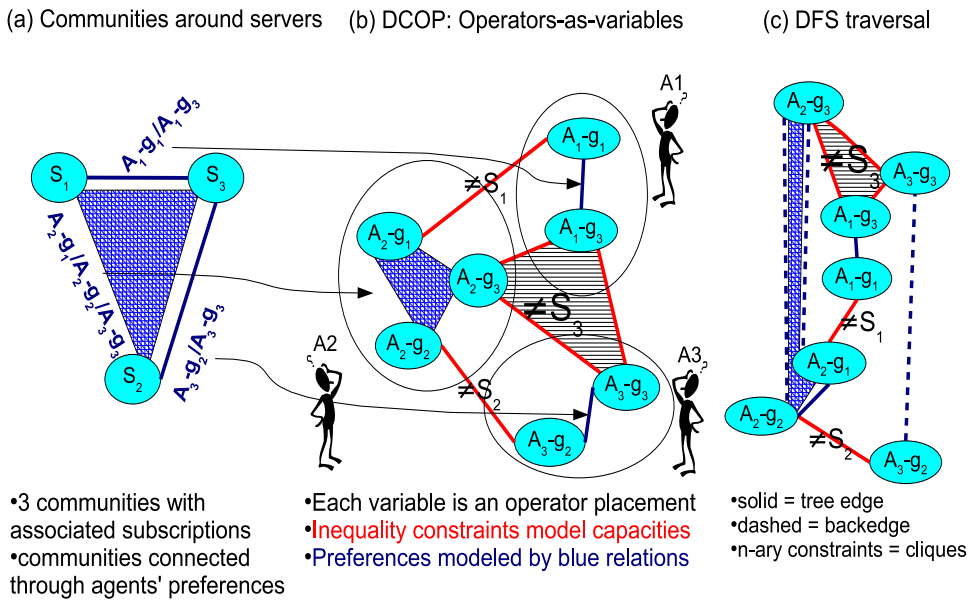


Figure 1: *An operator placement problem: (a) community formation , (b) DCOP model (operators-as-variables), and (c) DFS arrangement*

A_2 could express with its ternary relation (blue area connecting all its variables) that it prefers to have all its operators assigned on machines with the same OS, or that the sum of the bandwidths of the hosting servers must exceed a certain threshold, etc.

4.1.2 Interdependencies between local optimization problems

Local optimization problems are connected through *interagent* constraints. In this case, interagent constraints (commonly known to all agents) represent capacity constraints. For instance, consider a particular node h and let $X(h)$ denote the variables that include h in their domain. For each such h, there could be a constraint of the form "no more than C_h (some small integer) unique operators can be assigned to this node." For example, in Figure 1, agents A_1, A_2, and A_3 are each interested in placing an operator ($A_1\_g_3$, $A_2\_g_3$ and $A_3\_g_3$, respectively) on S_3. Since S_3 has limited capacity, all three variables are connected through a ternary capacity constraint. *Note that synergies can be found between operators in this formulation since agents 1 and 2 may have the same operator (e.g. apply an aggregation operator to readings from the same two producer nodes), and thus the coordinated decision by both agents to place this operator on the same node would only count once against the capacity of that node.* Agents must be able to identify this equivalence between operators.

4.2 Applying M-DPOP to Operator Placement

We describe in more detail the initial phase of M-DPOP whereby the communities are formed and the DFS structure is constructed. As user's variables in this model are initially private the process is slightly changed from that in Petcu et al. [19]:

1. Each agent A_i expresses internally its interests as an optimization problem $COP(A_i)$ (see 4.1.1)
2. Agents subscribe to the servers where they would like to place operators (see Figure 1). Each server S_j maintains a public subscriber list $A(S_j)$, and at the end of the subscription process, notifies all subscribers. Agents $A(S_j)$ are referred to as the *community* of server S_j. Each subscriber connects its corresponding variable to all other variables, thus forming a clique that corresponds to an n-ary capacity constraint.
3. Every agent can infer (or the server can specify) what combinations of operators observe the capacity of the server and capture this information via hard constraints.
4. Once the constraint graph is thus established, a depth-first-search traversal is constructed starting from a randomly chosen node (see Figure 1(c) for an example DFS)

Next, the bottom-to-top utility propagation and top-to-bottom decision propagation phases proceed as in M-DPOP. The capacity constraints from this model are the equivalent of the *hard-constraints* in M-DPOP parlance. As in M-DPOP, they are treated by the lowest agent in the DFS tree that has a variable involved in the constraint. For example, in Figure 1(c), we have the capacity constraint corresponding to S_3 as the shaded area involving $A_1\_g_3$, $A_2\_g_3$ and $A_3\_g_3$. This would be handled by A_1 when it sends its $UTIL$ message from $A_1\_g_3$ to $A_3\_g_3$. A_1 does this by assigning $-\infty$ to all value combinations of $A_1\_g_3$, $A_2\_g_3$ and $A_3\_g_3$ that violate the S_3 capacity constraint, and the normally computed valuations to the other combinations.

This ensures that throughout the whole propagation, all combinations of operator assignments that violate at least one capacity constraint will be assigned $-\infty$ utility and will therefore be avoided. On termination, once DCOP has been run for the main and each marginal problem the solution to the main problem is adopted by the server nodes and the bank collects VCG payments (defined in terms of the difference between main and marginal problem solutions and reported in a distributed manner.)

Applying M-DPOP to this domain provides a protocol that is *ex post Nash* faithful, meaning that if each client follows the M-DPOP algorithm then no single client can benefit by deviating from the algorithm (including in reporting untruthful information) *whatever* the private relations of the other agents.[9]

[9] In game-theoretic terms, the algorithm prescribes an *ex post* Nash equilibrium. The usual dominant-strategy equilibrium property that is achieved in VCG mechanisms is weakened to

5 Scalability of M-DPOP

This section presents a theoretical complexity analysis of our algorithm (Section 5.1), and an experimental evaluation on randomly generated problems (Section 5.2).

5.1 Theoretical Complexity

At the core of M-DPOP([19]), we use the DPOP([15]) algorithm for constraint optimization. Consequently, complexity-wise, M-DPOP can be seen as a sequence of DPOP runs: one for the main economy, and then another one for each marginal economy. M-DPOP has a mechanism for identifying parts of the computation from the main economy that can be safely (manipulation-free) reused in each marginal economy. Therefore, only in the worst case, when no effort can be reused, M-DPOP requires n+1 times the effort spent by DPOP. For more detail, please refer to Petcu et al. [19].

DPOP's complexity can be specified along two dimensions. First, in terms of number of messages, DPOP has the advantage that it requires only a linear number of messages. Second, in terms of the size of the messages, DPOP produces in the worst case messages whose size depends on the structure of the problem graph. This structure is captured with a parameter called the *induced width* of the graph, which depends on the clustering and the connectedness of the graph. It is important to notice that the width of the graph does not depend directly on the size of the graph, which means that certain types of problems can be easily solved although they are very large. Specifically, large but loose problems can be solved efficiently with DPOP. For more detail and formal proofs, please refer to Petcu and Faltings [15].

5.2 Experimental Evaluation

It is well known that the VCG mechanism requires optimal solutions in order to guarantee faithfulness. When applied to hard problems, this can render such schemes unfeasible, because finding the optimal solutions may be computationally impossible.

We present results from a simulation study to understand the scalability of our algorithm for solving operator placement problems in a multi-user, multi-server environment. We explore the relationship between the algorithm's computational and communication complexity and the number of agents and servers in the system. All results were generated using the FRODO multiagent simulation platform [13].

Users are divided into *similarity classes* so that users in different classes have (mainly) non-overlapping interest in different operators. This models a real overlay, where different classes of users issue non-overlapping queries.

ex post Nash: for example, truth-revelation is only a best-response if the other agents choose to implement the rules of the VCG mechanism correctly (which they will in equilibrium.)

Agents	Srvs	Vars	Constr	Msgs	MaxMsgSize	TotalMsgSize
20	10	32	42	31	96896	403399
40	19	60	78	59	117248	386314
60	29	92	117	91	303104	458533
80	40	128	171	127	1255424	2295570
100	49	156	210	155	1128675	2966304
120	59	188	255	187	20067123	33897613

Table 1: M-DPOP experiments on operator placement problems.

In these experiments, the classes are generated and then around 10 random users are added to a class. Classes are grouped in a hierarchy, according to similarity criteria at the class level. Users from neighboring (similar) classes can sometimes issue queries that cross class boundaries. Each user generates a query consisting of a random subset of operators from this user's similarity class (high probability) or another similar class (low probability). Each query consists of between one and five operators, which is consistent with previous work on stream queries [20]. Servers with random capabilities are then generated: each server is able to execute some random subset of the union of operators from a set of neighboring similarity classes. For each operator, the user generates a random *possibility set*, as described in Section 2. The possibility set for each operator is limited, since each server is allowed to run only a subset of the available operators. This models a real overlay where operators may require servers with special capabilities, memory, operating systems, etc. The user assigns random valuations for combinations of operator placements onto servers from each operators' possibility set. With all utility assignments made, the users then run M-DPOP.

Table 1 shows how our algorithm scales up with the size of the problems. The columns mean, in order: number of users in the network, number of servers, number of variables in the resulting problem, number of constraints in the resulting problem, number of messages required by DPOP to solve the problem, the size of the largest UTIL message in DPOP, and the total size of the UTIL messages generated in one solving process. The unit of the message size is valuations; in DPOP terms, it is the number of valuations of a UTIL message. The maximum message size is the size (in valuations) of the largest message, and the total is the summed size (in valuations) of all messages.

This table shows an explosion in message size as the problem gets large. This explosion is caused by two factors: the domain size of the variables, and the width of the DFS structure used by the M-DPOP algorithm. The width is adversely affected by the overlap of many queries on the same possible servers that could execute them. In terms of our operator placement problem, one can increase the scalability if one can make the similarity classes smaller and more disjoint and the possibility sets more disjoint. We plan to investigate how

these conditions can be achieved in practice in order to achieve good scalability, for instance by imposing constraints on the problem that will enforce sufficient structure.

6 Discussion

6.1 Dynamic allocation problems

The streaming application problem is really a dynamic problem. For dynamically evolving environments, Petcu and Faltings proposed in [17] a self-stabilizing version of DPOP that is guaranteed to continuously follow the evolution of a dynamic problem, always finding the optimal solution. An extended version of this technique ([16]) also ensures that when a new solution is derived upon a change, the cost of revising previously taken decisions is also taken into account.

One can leverage these techniques for the purpose of optimizing streaming applications as well. An important requirement is that the rate of change in the environment is small enough to allow the algorithm to stabilize and find the optimal solution. Provided this is the case, one can simply regard this evolution as a sequence of M-DPOP executions, where computation can be reused from $DCOP(\mathcal{A}, t_k)$ to $DCOP(\mathcal{A}, t_{k+1})$, and from $DCOP(-A_i, t_k)$ to $DCOP(-A_i, t_{k+1})$. Optimal solutions are computed, and taxes are levied once per time period. The ex post faithfulness properties are retained as long as user utilities can be decomposed in a linear fashion across time periods (e.g., when query streams are interruptible and utility accrues for each period of time a stream of a particular quality is received.) However, it will be important to understand whether new, undesirable equilibria are introduced in moving to the multi-period setting.

6.2 Approximations

M-DPOP is a complete algorithm (in the AI sense) and is guaranteed to terminate with the optimal solution. However, this guarantee comes at the cost of potentially large messages sizes. A practical systems solution must avoid such worst-case behavior. In earlier work, Petcu and Faltings [14] have shown that approximations of dramatically lower complexity still provide results that can be expected to be quite close to the optimum. The challenge in adopting approximate solutions within the framework of M-DPOP, and thus mechanism design, is that approximations can cause the *faithfulness* and incentives for truthfulness to unravel. The VCG payments continue to provide "self-correcting" incentives even with approximations (see Nisan and Ronen for example [12]), but progress in identifying useful equilibrium concepts in this context remains an important open problem in computational mechanism design. One way to retain faithfulness while introducing approximations is to impose constraints on the space of solutions that will be considered, and in a

way that is fixed and independent of agent messages. Progress in this direction remains future work.

7 Concluding Remarks

We have presented a distributed optimization approach to the optimal operator placement problem. We have introduced a DCOP model for this problem, and showed how one can apply M-DPOP to these problems. M-DPOP is a recently introduced optimization algorithm that makes faithful execution an *ex-post* Nash equilibrium. As M-DPOP is a derivative of DPOP, various techniques like self-stabilization for dynamic systems (see [17]) or linear size messages (see [18]) can be applied.

Preliminary results from simulation demonstrate that it will be of critical importance in large problem instance to identify, and then leverage, useful problem structure. We believe, for example, that one can take advantage of the special structure of the capacity constraints to develop more computationally efficient techniques, like Kumar et al. [9]; we will investigate these avenues in future work.

References

[1] D. Abadi, Y. Ahmad, H. Balakrishnan, et al. The Design of the Borealis Stream Processing Engine. Technical Report CS-04-08, Brown University, July 2004.

[2] Y. Ahmad and U. Çetintemel. Network-Aware Query Processing for Stream-based Applications. In *VLDB*, Aug. 2004.

[3] M. Balazinska, H. Balakrishnan, and M. Stonebraker. Contract-Based Load Management in Federated Distributed Systems. In *Proc. of NSDI'04*, San Francisco, CA, Mar. 2004.

[4] L. Chen, K. Reddy, and G. Agrawal. GATES: A Grid-Based Middleware for Processing Distributed Data Streams. In *Proc. of HPDC*, June 2004.

[5] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[6] E. Ephrati and J. Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proceedings of the National Conference on Artificial Intelligence, AAAI-91*, pages 173–178, Anaheim, CA, July 1991.

[7] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan. IrisNet: An Architecture for a World-Wide Sensor Web. *IEEE Pervasive Computing*, 2(4), Oct. 2003.

[8] R. Huebsch, J. M. Hellerstein, N. Lanham, et al. Querying the Internet with PIER. In *VLDB*, Sept. 2003.

[9] A. Kumar, A. Petcu, and B. Faltings. H-DPOP: Using hard constraints to prune the search space. In *IJCAI'07 - Distributed Constraint Reasoning workshop, DCR'07*, Hyderabad, India, Jan 2007.

[10] K. Lai and M. Baker. Measuring bandwidth. In *INFOCOM*, pages 235–245, 1999.

[11] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *AI Journal*, 161:149–180, 2005.

[12] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. In *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, pages 242–252, New York, NY, USA, 2000. ACM Press.

[13] A. Petcu. FRODO: A FRamework for Open/Distributed constraint Optimization. Technical Report No. 2006/001, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2006. http://liawww.epfl.ch/frodo/.

[14] A. Petcu and B. Faltings. Approximations in distributed optimization. In *Proceedings of the Eleventh International Conference on Principles and Practice of Constraint Programming (CP'05)*, Sitges, Spain, October 2005.

[15] A. Petcu and B. Faltings. DPOP: A scalable method for multiagent constraint optimization. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-05*, Edinburgh, Scotland, Aug 2005.

[16] A. Petcu and B. Faltings. Optimal solution stability in continuous time optimization. In *IJCAI05 - Distributed Constraint Reasoning workshop, DCR05*, Edinburgh, Scotland, August 2005.

[17] A. Petcu and B. Faltings. S-DPOP: Superstabilizing, fault-containing multiagent combinatorial optimization. In *Proceedings of the National Conference on Artificial Intelligence, AAAI-05*, Pittsburgh, USA, July 2005.

[18] A. Petcu and B. Faltings. O-DPOP: An algorithm for Open/Distributed Constraint Optimization. In *Proceedings of the National Conference on Artificial Intelligence, AAAI-06*, Boston, USA, July 2006.

[19] A. Petcu, B. Faltings, and D. Parkes. M-DPOP: Faithful Distributed Implementation of Efficient Social Choice Problems. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-06)*, Hakodate, Japan, May 2006.

[20] P. Pietzuch, J. Ledlie, J. Shneidman, M. Roussopoulos, M. Welsh, and M. Seltzer. Network-Aware Operator Placement for Stream-Processing Systems. In *ICDE*, April 2006.

[21] J. Shneidman and D. C. Parkes. Specification faithfulness in networks with rational nodes. In *Proc. of the 23rd ACM Symposium on Principles of Distributed Computing (PODC'04)*, St. John's, Canada, 2004.

[22] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: a lightweight network location service without virtual coordinates. In *SIGCOMM*, pages 85–96, New York, NY, USA, 2005. ACM Press.

[23] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem - formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):673–685, 1998.

QuickRank: A Recursive Ranking Algorithm

Amy Greenwald and John Wicks
Department of Computer Science
Brown University, Box 1910
Providence, RI 02912
{amy,jwicks}@cs.brown.edu

Abstract

This paper presents QuickRank, an efficient algorithm for ranking individuals in a society, given a network that encodes their relationships, assuming that network possesses an accompanying hierarchical structure: e.g., the Enron email database together with the corporation's organizational chart. The QuickRank design is founded on the "peer-review" principle, defined herein, and an hypothesis due to Bonacich. Together, these premises leads to a recursive ranking algorithm which is scalable, parallelizable, and easily updateable. Moreover, it is also potentially more resistant to link-spamming than other popular ranking algorithms.

1 Introduction

A fundamental problem in the field of social network analysis is to rank individuals in a society according to their implicit "importance" (e.g., power or influence), derived from a network's underlying topology. More precisely, given a social network, the goal is to produce a (cardinal) *ranking*, whereby each individual is assigned a nonnegative real value, from which an ordinal ranking (an ordering of the individuals) can be extracted if desired. In this paper, we propose a solution to this problem specifically geared toward social networks that possess an accompanying hierarchical structure.

A social network is typically encoded in a *link graph*, with individuals represented by vertices and relationships represented by directed edges, or "links," annotated with weights. Given a link graph, there are multiple ways to assign meaning to the weights. On one hand, one can view the weight on a link from i to j as expressing the distance from i to j—a quantity inversely related to j's importance. On the other hand, one can view each weight as the level of endorsement, or respect, i grants j—a quantity directly proportional to j's importance. We adopt this latter interpretation.

Under either interpretation (weights as distances or weights as endorsements), a social network can be seen as a collection of judgments, one made by each individual in the society. Correspondingly, we seek a means of aggregating individual judgments into a single collective ranking. In other words, we consider the aforementioned fundamental problem in social network analysis as akin to a key question in voting: how to aggregate the preferences of many individuals into a single collective persuasion that reflects the preferences of the population as a whole.

Given a link graph, perhaps the most basic ranking scheme is degree centrality, in which i's rank is a combined measure of its indegree, the strength of the endorsements i

receives, and outdegree, the strength of the endorsements i makes. It is straightforward to compute this metric. However, it could be argued that it is also sensible to take into account inferred endorsements: e.g., if i endorses j and j endorses k, then i endorses k in a sense. At the opposite end of the spectrum lie ranking schemes that incorporate all such inferred endorsements.

Central to these alternatives is a hypothesis due to Bonacich (1972): *an individual is deemed important if he is endorsed by other important individuals*. In other words, the strength of an endorsement should be construed relative to the rank of the individual making the endorsement. In terms of our voting analogy, Bonacich suggests relating the collective ranking to the sum of all individual judgments, each weighted by its respective rank as determined by the collective. The fixed point of this averaging process—the principal eigenvector of the link graph—defines Bonacich's metric, also known as eigenvector centrality. Although intuitively appealing, the computation of this fixed point can be prohibitive in large networks.

Recently, computer scientists have developed related schemes to rank web pages based on the Web's underlying topology. Viewed as a social network, web pages are individuals and hyperlinks are links. The most prominent approach to ranking web pages is the PageRank algorithm (Page and Brin, 1998; Page et al., 1998), upon which the Google search engine is built. PageRank aggregates the information contained in the Web's hyperlinks to generate a ranking using a process much like Bonacich's method for computing eigenvector centrality.

In this paper, we present QuickRank, an efficient algorithm for computing a ranking in an *hierarchical social network*. Many social networks are hierarchical. One apt example already mentioned is the Web, where the individuals are web pages, the network structure is provided by hyperlinks from one web page to another, and an explicit hierarchical structure is given by the Web's domains, subdomains, and so on. Another fitting example is the Enron email database, where individuals are employees, the network structure is given by emails from one employee to another, and an explicit hierarchical structure is given by the corporate hierarchy. Yet another compelling example is a citation index. In this case, the individuals are publications, the network structure is dictated by the references from one publication to another, and an explicit hierarchical structure is given by the categorization of publications by fields (e.g., computer science), subfields (e.g., AI, theory, and systems), and so on.

As we sketch the key ideas behind the QuickRank algorithm in this introductory section, we allude to the sample hierarchical social network shown in Figure 1, a network of web pages within a domain hierarchy. The web pages, indicated by gray rectangles, are the individuals in this society. Social relationships between these individuals (i.e., hyperlinks between web pages) are shown as dashed lines with arrows. The domain hierarchy is drawn using solid lines with domains and subdomains as interior nodes, indicated by solid black circles, and web pages as leaves (gray rectangles).

Up to normalization, a ranking is a probability distribution. Given any normalized ranking (i.e., probability distribution) of the individuals in an hierarchical social network, by conditioning that global distribution on a particular subcommunity (e.g., CS), we can derive a *conditional* ranking of only those individuals within that subcommunity (e.g., Pr[page 1 | CS], Pr[page 2 | CS], etc.). Likewise, from the respective

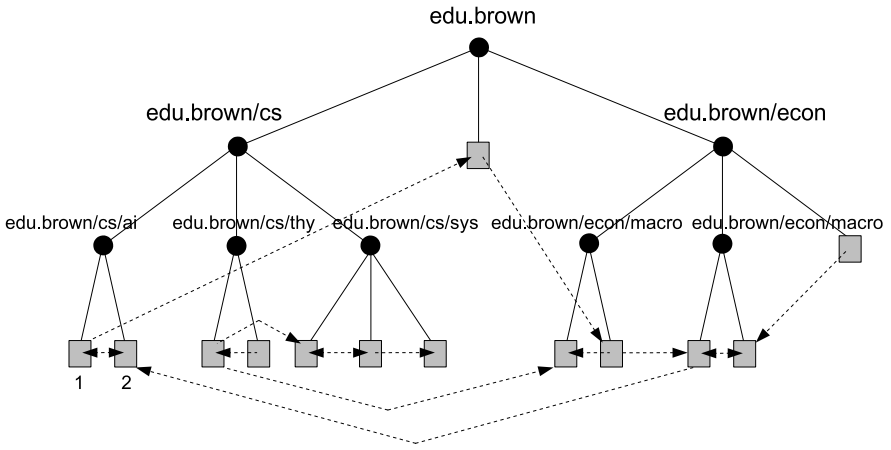


Figure 1: A sample hierarchical social network.

marginal probability of each subcommunity, we can infer what we call a *marginal* ranking[1] of subcommunities themselves (e.g., Pr[AI | CS], Pr[theory | CS], etc.). Conversely, it is straightforward to recover the global ranking by combining the conditional and marginal rankings using the chain rule. For example, Pr[page 1] = Pr[page 1 | AI] Pr[AI | CS] Pr[CS].

Hence, to compute a global ranking of the individuals in an hierarchical social network, it suffices to compute marginal rankings at all interior nodes (i.e., rank the children of all interior nodes), and combine those marginal rankings via the chain rule. To facilitate recursive implementation, QuickRank localizes the computation of each marginal ranking: any links to or from leaves outside the subtree at hand are ignored in such computations. Beyond this computational motivation, localizing marginal ranking computations can be motivated by the following "peer-review principle:" *endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate*.

Intuitively, it is plausible that ranking information among individuals in a tightly-knit community would be more reliable than ranking information among individuals who are only loosely connected. Recall the citation index, a natural example of an hierarchical social network. When a researcher cites a topic in his area of expertise, he is likely to select the most appropriate references. In contrast, if for some reason a researcher with expertise in one area (e.g., computer science) is citing a result in another (e.g., sociology), he may choose only somewhat relevant references. Hence, we contend that the peer-review principle, which justifies localized marginal ranking computations, befits at least some application areas.

[1] Viewing each interior node as the root of a subtree, we informally refer to the ranking of the children of an interior node as a marginal ranking, although such a ranking is technically a *conditional* marginal ranking, conditioned on the subcommunity defined by that subtree.

To fully implement the peer-review principle it is necessary to define some notion of approximate endorsements. To this end, we interpret an endorsement by an individual i in community A for another individual $j \neq i$ in another community $B \neq A$ as comprising part of an endorsement by A of B. More precisely, we aggregate endorsements by individuals in A for individuals in B into an endorsement by A of B by first scaling the endorsements from each i to each j by i's marginal rank, and then summing the resulting weighted endorsements. If we were to replace the target j of an endorsement by any other $j' \in B$, the resulting aggregate endorsement remains unchanged. In this sense, the original endorsement is viewed as "fuzzy" or "approximate." Moreover, by interpreting links originating at i as i's judgment, this aggregation process can be seen as an application of Bonacich's hypothesis (to obtain endorsements of each $j \in B$ by A) followed by a summation over all $j \in B$ (to obtain an endorsement of B).

Together, the principle of peer review and Bonacich's hypothesis lead to the QuickRank algorithm, which we illustrate on the example in Figure 1. We begin by restricting the link graph to, say, the AI subdomain, thereby constructing a local link subgraph. Next, we apply any "flat" ranking scheme (e.g., degree and eigenvector centrality and PageRank) to this link subgraph to produce a marginal ranking of the pages in the AI subdomain (i.e., a distribution over 1 and 2). Then, we scale the links from 1 to 4 and 2 to 3 by the marginal ranks of 1 and 2, respectively, to generate links from AI to 4 and 3. Finally, we sum these results to produce an aggregate link from AI to theory.

Repeating this procedure for the theory and systems subdomains, we "collapse" each of the CS subdomains into a leaf, and substitute these subdomains for their corresponding web pages in the link graph. We then proceed recursively, constructing a local link subgraph, and computing a marginal ranking of the CS subdomains. Combining this marginal ranking with the marginal rankings of the web pages in each CS subdomain yields a single marginal ranking of all the web pages in the CS domain. We repeat this process until the entire hierarchy has been collapsed into a single node, at which point we obtain a ranking of all pages in the `edu.brown` domain.

Overview This paper purports to contribute to the literature on social network analysis by introducing the QuickRank algorithm. As suggested by the previous example, QuickRank is parameterized by a "BaseRank" procedure (i.e., a flat ranking scheme, such as degree centrality) used to compute marginal rankings. We begin in the next section by precisely defining BaseRank procedures and identifying desirable properties of such procedures. In Section 3, we present pseudocode for the QuickRank algorithm. We also consider to what extent QuickRank preserves our previously identified desirable properties of BaseRank procedures. Then, in Section 4, we provide sample QuickRank calculations. Our first example illustrates the distinction between stand alone "BaseRanks" and "QuickRanks," the rankings output by these schemes. A further example shows how QuickRank is potentially more resistant to link-spamming than corresponding BaseRank procedures. We conclude in Section 5. A discussion of related work is deferred to the QuickRank technical report, currently in preparation.

2 A Unified View of Flat Ranking Algorithms

QuickRank is parametized by a flat (i.e., non-hierarchical) ranking algorithm, or a "BaseRank" procedure. In this section, we precisely define a BaseRank procedure, and we formulate the four flat ranking schemes mentioned in the introduction as such. We also present four desirable properties of BaseRank procedures, and discuss to what extent the four aforementioned ranking schemes satisfy these properties.

2.1 Preliminary Definitions

A social network encodes relationships among individuals in a society. Such a network can be represented by a *link graph*. Individuals $i, j \in \mathcal{I}$ are represented as *vertices*, and the fact that individual i relates to individual j is represented by a directed *link* from vertex i to vertex j, augmented by a nonnegative real-valued weight indicating the strength of i's relationship to j.

A *judgment* is a nonnegative, real-valued vector indexed on $\mathcal{I}$. We define an equivalence relation on judgments with r^1 and r^2 equivalent if $cr^1 = r^2$. For our purposes, a *ranking* is such an equivalence class $\langle r \rangle$ (although we often refer to a ranking by any representive of the class). A ranking has exactly one representative that is a probability distribution, which can be obtained by normalizing any other representative. Further, a ranking represents a consistent estimate of the relative merit of pairs of individuals: i.e., for all pairs of individuals i and j, the ranking of i relative to j, namely $\frac{r_i}{r_j} \in [0, \infty]$, is well-defined.

A *link graph* is a nonnegative, real-valued square matrix indexed on $\mathcal{I}$. We restrict attention to the case where the weights in the link graph may reasonably be interpreted as endorsements, rather than distances.[2] A *judgment* graph is a link graph further constrained to have *positive* diagonal entries. Each column in a judgment graph represents the judgment of one individual. The requirement that the diagonal be positive can be interpreted to mean that individuals are required to judge others relative to themselves. Whereas rankings are scale invariant, judgments are scale dependent.

In the introduction, we presented ranking schemes as operating on link graphs. That was a convenient oversimplification. More precisely, they map a judgment graph and a *prior* ranking to a *posterior* ranking. We view the inference of a judgment graph from a link graph as a preprocessing step. This step might consist of inserting self-loops: replacing zeros on the diagonal with ones. In the case of the Web or a citation database, for example, such self-loops would model each web page or publication as implicitly referring to (i.e., endorsing) itself.

Analogously, we define a *BaseRank* procedure as a higher-order function that takes a judgment graph to a mapping which infers a posterior ranking from a prior. When used within the QuickRank algorithm, we require that the posterior ranking output by the BaseRank procedure be normalized to a probability distribution. The prior ranking may be viewed as the persuasion of the "center" (i.e., the implementer of the ranking

[2] It seems conceivable that QuickRank can be suitably modified to handle the distance interpretation by redefining the peer-review notion of approximation as aggregating by taking a minimum instead of summing, but we have not yet explored any applications of this sort.

scheme). A BaseRank procedure then is a means of aggregating the judgments of the individuals in the society, and the center, into a single collective posterior ranking.

Given a judgment graph R and a prior ranking $\langle r \rangle$, Bonacich's hypothesis suggests that we may infer a collective judgment as $r' = Rr$. In this way, individual j's posterior position is the sum of each individual i's conception of j, weighted by the prior rank of i. By ignoring scale in r', we can infer the posterior ranking $\langle r' \rangle$. Note that the result of these two inference steps is well-defined, in that $\langle r' \rangle$ depends only on $\langle r \rangle$ and not on r itself. We use the term *linear* to describe a BaseRank procedure whose mapping from a prior ranking to a posterior abides by Bonacich's hypothesis.

2.2 Sample BaseRank Procedures

We now describe how the four ranking schemes mentioned in the introduction (i.e., indegree, outdegree, eigenvector centrality and PageRank) can be viewed BaseRank procedures. We assume that the link graph has been preprocessed, with self-loops inserted as necessary, to yield an "initial" judgment graph. Since the inference step is fixed, the key step in a linear BaseRank procedure is the way in which a "final" judgment graph is inferred from the initial judgment graph. The degree centrality metrics and PageRank are examples of linear BaseRank procedures, as is eigenvector centrality under certain assumptions (see Theorem 2.2).

The indegree and outdegree of individual i are defined respectively, as follows: given an initial judgment graph R,

$$\textsc{in}(i) = \sum_j R_{ij} \qquad \textsc{out}(i) = \sum_j R_{ji} \tag{1}$$

Both these centrality metrics can be understood as linear BaseRank procedures that infer a posterior ranking from a uniform prior. Indegree is simply the identity function: the initial and final judgment graphs are identical. Outdegree is the transpose operation: the initial and final judgment graphs are transposes of one another.

The PageRank algorithm is parameterized by a value $\epsilon \in (0, 1)$ and a distribution v, often referred to as a "personalization vector." In a preprocessing step, the columns of the judgment graph are normalized to yield a Markov matrix M. PageRank operates on the convex combination of M with the rank one Markov matrix vJ^t (where J ambiguously denotes any vector of all 1's), namely $M_\epsilon = (1-\epsilon)M + \epsilon v J^t$. This matrix is easily seen to be *regular* (i.e., possessing a single closed class, cf. Wicks and Greenwald (2005)), hence with a unique stable distribution v_∞. Moreover, Haveliwala and Kamvar (2003) have shown that M_ϵ has a second largest eigenvalue of $1-\epsilon$, so that $\lim_{k\to\infty} M_\epsilon^k v_0 = v_\infty$, for any initial distribution v_0, with convergence as $(1-\epsilon)^k$. This result follows alternatively by writing v_∞ as the limit of a geometric series:

Theorem 2.1 *If M is a Markov matrix and $M_\epsilon = (1-\epsilon)M + \epsilon v J^t$, then*

$$v_\infty = \lim_{k\to\infty} M_\epsilon^k v_0 = \epsilon \sum_{i=0}^{\infty} (1-\epsilon)^i M^i v \tag{2}$$

This theorem implies that PageRank is a linear BaseRank procedure, which takes an initial judgment graph M to a final judgment graph $\epsilon \sum_{i=0}^{\infty}(1-\epsilon)^i M^i$. The prior ranking corresponds to the personalization vector and the posterior ranking is a discounted sum of all the inferred rankings (including the prior).

Unlike degree centrality and PageRank, which we have shown are linear BaseRank procedures, eigenvector centrality is not. Given a judgment graph R and an prior ranking v_0, the algorithm infers a sequence of posterior rankings $v_{n+1} = \frac{Rv_n}{\|Rv_n\|_1}$. It can be shown that this sequence eventually converges to a fixed point v_∞, which can be interpreted as the collective ranking. Moreover, this iterative process can be expressed as a linear inference $v_\infty = \frac{R_\alpha v_0}{\|R_\alpha v_0\|_1}$, where α, and hence R_α, depend on the support of v_0. In particular, eigenvector centrality is a *piecewise*-linear BaseRank procedure. In the special case where the judgment graph is strongly-connected (i.e., R is irreducible), eigenvector centrality is linear, because R_α is constant (i.e., independent of α) and v_∞ is independent of v_0. Formally,

Theorem 2.2 *If a judgment graph $R \geq 0$ is irreducible with non-zero diagonal, there exists a unique ranking $v > 0$, such that $\|v\|_1 = 1$ and $Rv = \rho(R)v$, where $\rho(R)$ is the magnitude of the largest eigenvalue of R. Moreover, for any $v_0 \geq 0$, if $v_{n+1} = \frac{Rv_n}{\|Rv_n\|_1}$, $\lim_{n\to\infty} v_n = v$. That is, $v_\infty = v$ and for all α, $R_\alpha = vJ^t$.*

2.3 Generalized Proxy Voting

If we view each individual's rank as a collection of proxy (i.e., infinitely divisible and transferable) votes, then a judgment graph may be interpreted as a *proxy-vote specification* indicating how each individual is willing to assign his proxy votes to others. Given a prior ranking (i.e., an initial allocation of proxy votes), the posterior inferred by a linear BaseRank procedure is a reallocation based on the results of a single round of proxy voting. More generally, in *generalized proxy-voting* (GPV), individuals cast their votes repeatedly over time (i.e., each posterior serves as a prior in the next round), until ultimately, the sequence of posteriors is averaged into a final vote count: i.e., a final ranking.

While historically PageRank has been viewed in terms of a "random-surfer" model (cf. Page et al. (1998)), Theorem 2.1 suggests that it may be more aptly viewed as a GPV mechanism with a discount factor $\gamma \in [0, 1)$. In particular, for a given prior ranking v, the posterior computed by PageRank can be expressed as $(1-\gamma)^{-1}\sum_{i=0}^{\infty}\gamma^i M^i v$. Notice that this is just the average of the inferred rankings $M^i v$, where i is distributed geometrically with mean γ. It is natural to generalize to allow weighting by arbitrary distributions, $\sum_{i=0}^{\infty}\alpha_i M^i v$, or even as the limit of such, $\lim_{N\to\infty}\sum_{i=0}^{N}\alpha_{i,N}M^i v$. Formally, we define a generalized proxy-voting mechanism as a (linear) BaseRank procedure that takes an initial judgment graph M into a final judgment graph $\lim_{N\to\infty}\sum_{i=0}^{N}\alpha_{i,N}M^i$.

Observe that all the flat ranking schemes mentioned above, except outdegree, are not only linear BaseRank procedures, but can be seen as GPV mechanisms as well. Indegree is a trivial instance of GPV with $\alpha_{i,N} = \delta_{i,1}$. By Theorem 2.1, PageRank is a GPV mechanism with $\alpha_{i,N} = \epsilon(1-\epsilon)^i$. Finally, if we restrict atten-

tion to irreducible judgment graphs, eigenvector centrality is a GPV mechanism, with $\alpha_{i,N} = \begin{cases} \frac{1}{N+1} & \text{if } 0 \leq i \leq N \\ 0 & \text{otherwise} \end{cases}$. This final claim follows Theorem 2.2 and the well-known fact that $\lim_{i\to\infty} s_i = \lim_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k-1} s_i$. Although outdegree, which takes R to R^t is linear, it is not a GPV mechanism.

2.4 Axioms

Next, we identify two types of judgment graphs that have natural interpretations, and on which a particular behavior for a BaseRank procedure seems preferred. First, consider the identity matrix I as a judgment graph—the *identity* graph—in which each individual ranks himself infinitely superior to all others. Such a ranking graph provides no basis for modifying a prior ranking. Thus, on this input, it seems reasonable that a BaseRank procedure should act as the identity function (i.e., posterior = prior).

Second, consider the case of a *consensus* graph, that is, a judgment graph xy^t, where x is a distribution and y_i is individual i's arbitrary scaling factor. In other words, a consensus graph is a rank 1 matrix: everyone agrees on the ranking x, up to a multiple. Since there is consensus among the individuals in the society, we contend that any prior ranking should be ignored. A BaseRank procedure should simply return the consensus x. We restate these two properties succinctly, as follows:

Identity: $BaseRank(I) = \text{id}$

Consensus: $BaseRank(xy^t) = x$

Another important issue associated with ranking schemes is that of manipulation via "link spamming." The goal of link spamming is to game a ranking system by creating many false nodes, sometimes called sybils (Cheng and Friedman, 2006), that link to some node n, thereby attempting to influence the rank of node n. Web spamming is a particularly popular form of link spamming (Gyongyi and Garcia-Molina, 2004).

A judgment graph inhabited by sybils takes the following form: $M' = \begin{bmatrix} M & \overline{N} \\ 0 & \overline{M} \end{bmatrix}$, where M is the original judgment graph (i.e., without the sybils), $\overline{N}$ describes the links from the sybils to existing members of the society, and $\overline{M}$ describes the links among sybils. Since sybils are new to the community, and hence unknown its original members, we assume that there are no links from those members to sybils.

Observe that generalized proxy-voting mechanisms are spam-resistant in the following sense: Given a prior ranking which places no weight on sybils, the posterior ranking computed with respect to the modified judgment graph M' is, for all intents and purposes, equivalent to the posterior ranking computed with respect to the original judgment graph M. That is,

Property	Indegree	Outdegree	Eigenvector	PageRank
Linear	Yes	Yes	*No*	Yes
GPV	Yes	*No*	Yes	Yes
Identity	Yes	Yes	Yes	Yes
Consensus	Yes	Yes	Yes	*No*

Table 1: Some properties of ranking schemes.

Theorem 2.3 *If* $M' = \begin{bmatrix} M & \overline{N} \\ 0 & \overline{M} \end{bmatrix}$, $v' = \begin{bmatrix} v \\ 0 \end{bmatrix}$, *and* $BaseRank(\cdot) = \lim_{N\to\infty} \sum_{i=0}^{N} \alpha_{i,N} (\cdot)^i$, *then* $BaseRank(M')v' = \begin{bmatrix} BaseRank(M)v \\ 0 \end{bmatrix}$.

For example, since PageRank is a GPV mechanism, we apply Theorem 2.3 to show that the posterior ranking of non-sybils is unaffected by their presence, if we assign sybils a prior rank of 0. In other words, if sybils can be detected *a priori*, then PageRank may be rendered immune to such an attack. Although the corresponding Markov matrix need not be irreducible for such a "personalization" vector, we conclude from Theorem 2.1 that the Markov process converges for *all* prior rankings v_0. Note that this conclusion follows specifically from our interpretation of PageRank as a GPV mechanism, as opposed to the traditional "random surfer" model.

Table 1 summarizes how each of the four ranking schemes discussed in this section behave with respect to the four properties of BaseRank procedures discussed in this section. PageRank does *not* satisfy the consensus property because it is always biased to some degree by the prior ranking. However, using the notation introduced above, if we instead define $M_\epsilon = (1-\epsilon)M + \epsilon M v J^t$, the resulting algorithm satisfies all four properties. This modified PageRank corresponds to a linear BaseRank procedure with final judgment graph $\epsilon \sum_{i=0}^{\infty} (1-\epsilon)^i M^{i+1}$, that is, the posterior is a discounted sum of all inferred rankings *excluding* the prior.

Fundamentally, QuickRank's design is based on the two key ideas discussed in the introduction, namely the peer-review principle and Bonacich's hypothesis. However, as QuickRank is parameterized by a BaseRank procedure, it is also designed to preserve the Identity and Consensus properties. In the next section, we detail the algorithm and argue informally that it indeed preserves these two properties of BaseRank procedures, although it fails to preserve linearity. When we present sample calculations in Section 4, we note that QuickRank preserves the spam-resistance of its BaseRank procedure, and we illustrate its potential to resist spam even further.

3 QuickRank: The Algorithm

QuickRank operates on a hierarchical social network, that is a judgment[3] graph R whose vertices are simultaneously leaves of a tree T. At a high level, QuickRank first ranks the leaves using the link information contained in the local subgraphs; it then propagates those local[4] rankings up the tree, aggregating them at each level, until they have been aggregated into a single global ranking. Ultimately, *a node's QuickRank is the product of its own local rank and the local rank of each of its ancestors.* QuickRank is parameterized by a BaseRank procedure, which it uses to compute local rankings. It also takes as input a prior ranking of the leaves. It outputs a posterior distribution.

Although we present QuickRank pseudocode (see Algorithm 1) that is top-down and recursive, like many algorithms that operate on trees, the simplest way to visualize the QuickRank algorithm is bottom-up. From this point of view, QuickRank repeatedly identifies "collapsible" nodes in T, meaning the root nodes of subtrees of depth 1, and collapses them into leaf nodes (i.e., subtrees of depth 0) until there are no further opportunities for collapsing: i.e., until T itself is a leaf node. Collapsing node n entails: (i) computing a local ranking at n, that is a ranking of n's children, and (ii) based on this local ranking, aggregating the rankings and the judgments of n's children into a single ranking and a single judgment, both of which are associated with n.

Note that QuickRank is a well-defined algorithm: that is, the order in which local rankings are computed does not impact the global ranking. This property is immediate, since QuickRank propagates strictly local calculations up the tree in computing its global output. Moreover, the collapse operation replaces a subtree of depth 1 with a subtree of depth 0 so that QuickRank is guaranteed to terminate.

Data Structures Algorithm 1 takes as input T_n, subtree of T rooted at node n, and returns two data structures: (i) a ranking of all leaves (with support only on T_n) and (ii) a judgment, which is the average of all judgments of T_n's leaves, weighted by the ranking computed in (i). At leaf node n, the ranking is simply the probability distribution with all weight on n, denoted e_n, and the judgment is given by R_n.

Computing Local Rankings Recall that the main idea underlying QuickRank is to first compute local rankings, and to then aggregate those local rankings into a single global ranking. Given a collapsible node n, a local ranking is a ranking of n's children. To compute such a ranking, QuickRank relies on a BaseRank procedure.

There are two inputs to this BaseRank procedure. The first is n's local (i.e., marginal) prior ranking. The second is a local judgment graph M. For j and k both children of node n, the entry of M in the row corresponding to k and the column corresponding to j is the aggregation of all endorsements from leaves in T_j to leaves in T_k, equal to the sum of all entries in the jth judgment corresponding to leaves of T_k.

Aggregating Rankings and Links To aggregate the rankings of n's m children into a single ranking associated with n, QuickRank averages the rankings $r^1, \ldots, r^m$ ac-

[3] As above, we assume the link graph has been preprocessed to form a judgment graph.

[4] Whereas in the introduction, we used the term marginal, we now use the term local to refer to the ranking of a node's children. The salient point here is: this ranking is computed using strictly local information.

cording to the weights specified by the local ranking r. If we concatenate the m rankings into a matrix $Q = \left[\begin{array}{ccc} r^1 & \cdots & r^m \end{array} \right]$, then the aggregation of rankings can be expressed simply as Qr. Also associated with each child j of a collapsible node n is a judgment l^j. These judgments are aggregated in precisely the same way as rankings.

Algorithm 1 QuickRank(node n)

```
1:  if n.isLeaf() then
2:     return ⟨n.getJudgment(), e_n⟩
3:  else
4:     m = n.numChildren()
5:     for j = 1 to m do
6:        ⟨l^j, r^j⟩ ← QuickRank(n.getChild(j))
7:        for k = 1 to m do
8:           M_kj = Sum(l^j, n.getChild(k))
9:        end for
10:    end for
11:    P = [ l^1  ...  l^m ]
12:    Q = [ r^1  ...  r^m ]
13:    r = BaseRank(M, n.getLocalPriorRanking())
14:    return ⟨Pr, Qr⟩
15: end if
```

We now argue that if the BaseRank procedure satisfies the Identity and Consensus properties, then so, too, does QuickRank. First, notice that, when restricted to any subcommunity (i.e., square, diagonal block), an identity or consensus graph yields the same type of graph again. Moreover, aggregating links in such a community within the original graph (i.e., summing rows and averaging columns) also results in the same type of graph. Consequently, if QuickRank employs a BaseRank procedure with the Identity property, it will output the prior distribution on the identity graph, since the prior local rankings will remain unchanged at each level in the hierarchy.

Now consider a consensus graph with ranking x *s.t.* $\|x\|_1 = 1$. Restriction to a subcommunity gives a consensus graph on the corresponding conditional distribution of x. Likewise, aggregation produces a consensus graph on the corresponding marginal distribution of x. If QuickRank employs a BaseRank algorithm with the consensus property on a consensus graph, it will gradually replace the prior distribution at the leaves with the conditional distributions of x, until it finally outputs x itself.

We conclude this section by pointing out that, even if the BaseRank procedure is linear, QuickRank may not be expressible as a linear inference. Normalizing local rankings to form distributions can introduce non-linearities. In the next section, we provide sample QuickRank calculations.

4 Examples

We now present two examples that verify our intuition regarding QuickRank and illustrate some of its novel features. Recall that QuickRank, as it operates on an hierarchical social network (HSN), is parameterized by a prior ranking and a BaseRank procedure.

First, consider the HSN shown in Figure 2a. The hierarchy is drawn using solid lines. The link graph is indicated by dotted lines between the numbered leaves. All weights are assumed to be 1. Computing QuickRanks for this HSN, varying the BaseRank procedure among indegree, eigenvector centrality, and PageRank,[5] but always assuming a uniform prior ranking, leads to the rankings, cardinal and ordinal, shown in Table 2. The values in the posterior distributions have been rounded; hence, the ordinal rankings more precisely reflect the exact values in those distributions.

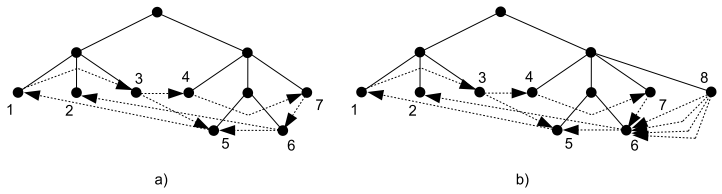


Figure 2: Two examples of hierarchical social networks.

Table 2: BaseRanks and QuickRanks from Figure 2a and uniform prior.

		Indegree	Eigenvector	PageRank
Flat	cardinal	{0.13, 0.13, 0.13, 0.13, 0.2, 0.13, 0.13}	{0.19, 0.08, 0.16, 0.14, 0.22, 0.10, 0.12}	{0.14, 0.32, 0.11, 0.09, 0.14, 0.09, 0.11}
	ordinal	$5 > 1 = 2 = 3 = 4 = 6 = 7$	$5 > 1 > 3 > 4 > 7 > 6 > 2$	$2 > 1 > 5 > 3 > 7 > 6 > 4$
QuickRank	cardinal	{0.10, 0.10, 0.19, 0.09, 0.23, 0.11, 0.18}	{0, 0, 0.41, 0, 0.59, 0, 0}	{0.04, 0.14, 0.25, 0.04, 0.41, 0.06, 0.06}
	ordinal	$5 > 3 > 7 > 6 > 1 = 2 > 4$	$5 > 3 > 1 = 2 = 4 = 6 = 7$	$5 > 3 > 2 > 7 > 6 > 1 > 4$

For each BaseRank procedure, we list two pairs of rankings: that which results from ignoring the hierarchy, and that which results from exploiting it using QuickRank. When we ignore the hierarchy, all three algorithms rank leaf 1 above (or equal to) 3. However, since 1 defers to 3 (i.e., 1 endorses 3, but not vice versa), based on our peer-review principle, 3 should be ranked higher than 1. This outcome indeed prevails in the QuickRanks, for all three BaseRank procedures.

As an added benefit, QuickRank can be more resistant to link spamming than BaseRank procedures that do not exploit hierarchies. To demonstrate this phenomenon, in Figure 2b, we introduce a sybil, leaf 8, into our original example to try and raise the rank of 6 by recommending it highly. Note the multiplicity of links from 8 to 6.

[5]The results of ranking with outdegree are not qualititatively different, but are omitted for lack of space.

Table 3: Figure 2b with Indegree as BaseRank.

		Uniform Prior	Weighted Prior
Flat	cardinal	{0.10, 0.10, 0.10, 0.10, 0.10, 0.35, 0.10, 0.05}	{0.13, 0.13, 0.13, 0.13, 0.13, 0.2, 0.13, 0.0}
	ordinal	$6 > 1 = 2 = 3 = 4 = 5 = 7 > 8$	$6 > 1 = 2 = 3 = 4 = 5 = 7 > 8$
QuickRank	cardinal	{0.09, 0.09, 0.18, 0.06, 0.28, 0.14,0.11, 0.06}	{0.10, 0.10, 0.19, 0.09, 0.23, 0.11, 0.19, 0.0}
	ordinal	$5 > 3 > 6 > 7 > 1 = 2 > 4 = 8$	$5 > 3 > 7 > 6 > 1 = 2 > 4 > 8$

Applying QuickRank with indegree as BaseRank to this example yields the rankings shown in Table 3. Using a uniform prior, the sybil is able to raise the rank of 6 over 7 and 6 over 4, whether we exploit the hierarchy (i.e., use QuickRank) or not (i.e., compute indegrees directly). QuickRank cannot prevent this outcome, since the sybil is an accepted member of 4's and 7's community. However, the influence of the sybil is somewhat mitigated under QuickRank. Since the resulting ranking must respect the hierarchy, the effect of the sybil is to raise the ranks of *both* 5 and 6 (i.e., both values in the posterior distribution). No amount of link spam from a sybil outside their local community can increase the rank of 6 relative to 5.

Moreover, if one is able to identify sybils *a priori*, by setting the prior ranks of sybils to zero, one can reduce their influence even further. If we use a prior ranking which is weighted against the sybil, say uniform over 1-7 and zero on 8, Table 3 shows that indegree produces the same rankings as in Table 2, that is, *without* the sybil, whether we exploit the hierarchy or not. In general, Theorem 2.3 states that any BaseRank procedure which is a GPV mechanism will necessarily exhibit this same behavior. QuickRank is not a GPV scheme (recall that QuickRank is nonlinear but that GPV schemes are linear). Still, QuickRank preserves the spam-resistance property characteristic of GPV mechanisms.

5 Conclusion

Social network, or link, analysis is regularly applied to information networks to compute rankings (Garfield, 1972; Kleinberg, 1998; Page and Brin, 1998; Page et al., 1998) and to social networks (Bonacich, 1972; Hubbell, 1965; Katz, 1953; Wasserman and Faust, 1994) to determine standing. We discuss two examples of information networks with inherent hierarchical structure: the Web and citation indices. Social networks, like the Enron email database, also exhibit hierarchical structure. Simon (1962) suggests that such hierarchies are ubiquitous:

> Almost all societies have elementary units called families, which may be grouped into villages or tribes, and these into larger groupings, and so on. If we make a chart of social interactions, of who talks to whom, the clusters of dense interaction in the chart will identify a rather well-defined hierarchic[6] structure.

[6]Simon's use of the terminology "hierarchic" is slightly broader than our use of "hierarchical structure,"

Still, to our knowledge, link analysis procedures largely ignore any hierarchical structure accompanying an information or social network. In this paper, we introduced QuickRank, a link analysis technique for ranking individuals that exploits hierarchical structure. The foundational basis for QuickRank is the peer-review principle, which implies that the relative ranking between two individuals be determined by their local ranks in the smallest community to which they both belong. This principle, together with an hypothesis due to Bonacich, leads to a recursive algorithm which is scalable, parallelizable, and easily updateable.

For a large-scale network such as the Web, we anticipate that QuickRank will yield substantial computational gains over standard ranking methods (e.g., calculating PageRanks via the power method). Moreover, it appears more resistant to link-spamming than other popular ranking algorithms on contrived examples, although it remains to verify this claim empirically.

Acknowledgments

This research was supported by NSF Career Grant #0133689 and NSF Grant #0534586.

References

Phillip Bonacich. Factoring and weighting approaches to status scores and clique detection. *Journal of Mathematical Sociology*, pages 113–120, 1972.

Alice Cheng and Eric Friedman. Manipulability of pagerank under sybil strategies. In *First Workshop on the Economics of Networked Systems (NetEcon06)*, 2006. URL `http://www.cs.duke.edu/nicl/netecon06/papers/ne06-sybil.pdf`.

Eugene Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178:471–479, 1972.

Zoltan Gyongyi and Hector Garcia-Molina. Web spam taxonomy. Technical report, Stanford University Technical Report, 2004.

T. Haveliwala and S. Kamvar. The second eigenvalue of the google matrix. Technical report, Stanford University Technical Report, 2003.

Charles H. Hubbell. An input-output approach to clique identification. *Sociometry*, 28:377–399, 1965.

L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 668–677, 1998.

Lawrence Page and Sergey Brin. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference (WWW7)*, 1998.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

Herbert A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962.

Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.

John R. Wicks and Amy Greenwald. An algorithm for computing stochastically stable distributions with applications to multiagent learning in repeated games. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 623–632, 2005.

by which we mean tree structure. Still, the point remains: hierarchies (or approximations thereof) arise naturally in societies.

Hybrid Elections Broaden Complexity-Theoretic Resistance to Control[1]

Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe

Abstract

Electoral control refers to attempts by an election's organizer ("the chair") to influence the outcome by adding/deleting/partitioning voters or candidates. The groundbreaking work of Bartholdi, Tovey, and Trick [BTT92] on (constructive) control proposes computational complexity as a means of resisting control attempts: Look for election systems where the chair's task in seeking control is itself computationally infeasible.

We introduce and study a method of combining two or more candidate-anonymous election schemes in such a way that the combined scheme possesses all the resistances to control (i.e., all the NP-hardnesses of control) possessed by *any* of its constituents: It combines their strengths. From this and new resistance constructions, we prove for the first time that there exists an election scheme that is resistant to all twenty standard types of electoral control.

Key words: multiagent systems, computational social choice, preference aggregation, computational complexity, electoral control.

1 Introduction

Elections are a way of, from a collection of voters' (or agents') individual preferences over candidates (or alternatives), selecting a winner (or outcome). The importance of and study of elections is obviously central in political science, but also spans such fields as economics, mathematics, operations research, and computer science. Within computer science, the applications of elections are most prominent in distributed AI, most particularly in the study of multiagent systems. For example, voting has been concretely proposed as a computational mechanism for planning [ER91,ER93] and has also been suggested as an approach to collaborative filtering [PHG00]. However, voting also has received attention within the study of systems. After all, many distributed algorithms must start by selecting a leader, and election techniques have also been proposed to attack the web page rank aggregation problem and the related issue of lessening the spam level of results from web searches [DKNS01,FKS03]. Indeed, in these days of a massive internet with many pages, many surfers, and many robots, of intracorporate decision-making potentially involving electronic input

[1]Conference version to appear in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*; a full version is available as [HHR06]. Supported in part by DFG grants RO 1202/9-1 and RO 1202/9-3, NSF grants CCR-0311021 and CCF-0426761, a Friedrich Wilhelm Bessel Research Award, and the Alexander von Humboldt Foundation's TransCoop program.

from many units/individuals/warehouses/trucks/sources, and more generally of massive computational settings including many actors, it is easy to note any number of situations in which elections are natural and in which the number of candidates and/or voters might be massive. For example, suppose amazon.com were to select a "page of the week" via an election where the candidates were all its web pages and the voters were all visiting surfers (with preferences inferred from their page-viewing times or patterns); such an election would have an enormous number of candidates and voters. All these applications are exciting, but immediately bring to a theoretician's mind the worry of whether the complexity of implementing election systems is satisfyingly low and whether the complexity of distorting (controlling or manipulating) election systems is reassuringly high.

Since the complexity of elections is a topic whose importance has made itself clear, it is natural to ask whether the standard tools and techniques of complexity-theoretic study exist in the context of elections. One important technique in complexity is the combination of problems. For example, for sets in complexity theory, a standard approach to combination is the join (also known as the disjoint union and as the marked union): $A \oplus B = \{0x \mid x \in A\} \cup \{1y \mid y \in B\}$.

In some sense, our work in this paper can be thought of as simply providing, for elections, an analog of the join. That is, we will propose a method of combining two (or more) elections in a way that will maintain desirable simplicity properties (e.g., if all of the constituent elections have polynomial-time winner algorithms then so will our combined election) while also inheriting quite aggressively desirable hardness properties (we will show that any resistance-to-control—in the sense that is standard [BTT92] and that we will provide a definition of later—possessed by even one of the constituent elections will be possessed by the combined election). One cannot directly use a join to achieve this, because the join of two sets modeling elections is not itself an election. Rather, we must find a way of embedding into election specifications—lists of voter preferences over candidates—triggers that both allow us to embed and switch between all the underlying election systems and to not have such switching go uncontrollably haywire when faced with electoral distortions such as adding/deleting/partitioning voters/candidates, since we wish hardness with respect to control by such mechanisms to be preserved.

We above have phrased this paper's theme as the development of a way of combining multiple election systems—and in doing so, have desirable types of simplicity/complexity inheritance. However, this paper also has in mind a very specific application—both for its own interest and as a sounding board against which our election hybridization scheme can be tested. This application is the control of election systems.

In election control, we ask whether an election's organizer (the chair) can by some specific type of manipulation of the election's structure (adding/deleting/partitioning voters/candidates) cause a specified candidate to be the (unique) winner. As mentioned earlier, the complexity-theoretic study

of control was proposed by Bartholdi, Tovey, and Trick in 1992 [BTT92]. We will closely follow their model. In this model, the chair is assumed to have knowledge of the vote that will be cast by each voter, and there are ten different types of control (candidate addition, candidate deletion, voter addition, voter deletion, partition of candidates, run-off partition of candidates, and partition of voters [BTT92]—and for each of the three partition cases one can have subelection ties promote or can have subelection ties eliminate, see [HHR05a]).

Of course, the dream case would be to find an election system that has the desirable property of having a polynomial-time algorithm for evaluating who won, but that also has the property that for every single one of the ten standard types of control it is computationally infeasible (NP-hard) to assert such control. Unfortunately, no system yet has been proven resistant to all ten types of control. In fact, given that broad "impossibility" results exist for niceness of preference aggregation systems (e.g., Arrow's Theorem [Arr63]) and for nonmanipulability of election systems (e.g., the Gibbard–Satterthwaite and Duggan–Schwartz Theorems ([Gib73,Sat75,DS00], see also [Tay05])), one might even momentarily wonder whether the "dream case" mentioned above can be proven impossible via proving a theorem of the following form: "For no election system whose winner complexity is in P are all ten types of control NP-hard." However, such a claim is proven impossible by our work: Our hybrid system in fact will allow us to combine *all* the resistance types of the underlying elections. And while doing so, it will preserve the winner-evaluation simplicity of the underlying elections. Thus, in particular, we conclude that the "dream case" holds: There is an election system—namely, our hybridization of plurality and Condorcet elections—that is resistant to all ten types of constructive control. We also show—by building some artificial election systems achieving resistance to destructive control types for which no system has been previously proven resistant and then invoking our hybridization machinery—that there is an election system that is resistant to all ten types of destructive control (in which the chair's goal is to preclude a given candidate from being the (unique) winner) as well as to all ten types of constructive control (Theorem 3.8).

Our hybridization system takes multiple elections and maintains their simplicity while inheriting each resistance-to-control possessed by any one of its constituents. Thus, it in effect unions together all their resistances—thus the "broaden" of our title. We mention in passing that in the quite different setting of election manipulation (which regards not actions by the chair but rather which regards voters altering their preferences in an attempt to influence who becomes the winner) [BTT89a], there has been some work by Conitzer and Sandholm [CS03] regarding making manipulation hard, even for systems where it is not hard, by changing the system by going to a two-stage election in which a single elimination preround is added, and Elkind and Lipmaa [EL05] have generalized this to a sequence of elimination rounds conducted under some system(s) followed by an election under some other system. Though the latter paper like this paper uses the term "hybrid," the domains differ sharply and the methods of election combination are nearly opposite: Our approach (in or-

der to broaden resistance to control) embeds the election systems in parallel and theirs (in order to fight manipulation) strings them out in sequence. Of the two approaches, ours far more strongly has the flavor of our simple motivating example, the join.

The previous work most closely related to that of this paper is the constructive control work of Bartholdi, Tovey, and Trick [BTT92] and the destructive control work of Hemaspaandra, Hemaspaandra, and Rothe [HHR05a]. Work on bribery is somewhat related to this paper, in the sense that bribery can be viewed as sharing aspects of both manipulation and control [FHH06]. Of course, all the classical [BTT89b,BTT89a,BO91] and recent papers (of which we particularly point out, for its broad framework and generality, the work of Spakowski and Vogel [SV00]) on the complexity of election problems share this paper's goal of better understanding the relationship between complexity and elections.

We here omit proofs due to lack of space, but detailed proofs are available in the full version of this paper [HHR06].

2 Definitions and Discussion

2.1 Elections

An election system (or election rule or election scheme or voting system) $\mathcal{E}$ is simply a mapping from (finite though arbitrary-sized) sets (actually, mathematically, they are multisets) V of votes (each a preference order—strict, transitive, and complete—over a finite candidate set) to (possibly empty, possibly nonstrict) subsets of the candidates. All votes in a given V are over the same candidate set, but different V's of course can be over different (finite) candidate sets. Each candidate that for a given set of votes is in $\mathcal{E}$'s output is said to be a *winner*. If for a given input $\mathcal{E}$ outputs a set of cardinality one, that candidate is said to be the *unique winner*. Election control focuses on making candidates be unique winners and on precluding them from being unique winners.

Throughout this paper, a voter's preference order will be exactly that: a tie-free linear order over the candidates. And we will discuss and hybridize only election systems based on preference orders.

We now define two common election systems, plurality voting and Condorcet voting. In *plurality voting*, the winners are the candidates who are ranked first the most. In *Condorcet voting*, the winners are all candidates (note: there can be at most one and there might be zero) who strictly beat each other candidate in head-on-head majority-rule elections (i.e., get *strictly* more than half the votes in each such election). For widely used systems such as plurality voting, we will write plurality rather than $\mathcal{E}_{\text{plurality}}$.

We say that an election system $\mathcal{E}$ is *candidate-anonymous* if for every pair of sets of votes V and V', $\|V\| = \|V'\|$, such that V' can be created from V by applying some one-to-one mapping h from the candidate names in V onto new candidate names in V' (e.g., each instance of "George" in V is mapped by h

to "John" in V' and each instance of "John" in V is mapped by h to "Hillary" in V' and each instance of "Ralph" in V is mapped by h to "Ralph" in V') it holds that $\mathcal{E}(V') = \{c' \mid (\exists c \in \mathcal{E}(V))\,[h(c) = c']\}$. Informally put, candidate-anonymity says that the strings we may use to name the candidates are all created equal. Note that most natural systems are candidate-anonymous. For example, both the election systems mentioned immediately above—plurality-rule elections and the election system of Condorcet—are candidate-anonymous.

2.2 Our Hybridization Scheme

We now define our basic hybridization scheme, *hybrid*.

Definition 2.1 *Let $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ be election rules that take as input voters' preference orders. Define hybrid$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ to be the election rule that does the following: If there is at least one candidate and all candidate names (viewed as natural numbers via the standard bijection between Σ^* and $\mathbb{N}$) are congruent, modulo k, to i (for some i, $0 \leq i \leq k-1$) then use election rule $\mathcal{E}_i$. Otherwise use, by convention, $\mathcal{E}_{k-1}$ as the default election rule.*

Having defined our system there is much to discuss. Why did we choose this system? What are its properties? What other approaches did we choose not to use, and why? What aspects of the input is our method for switching between election systems using, and what aspects is it choosing not to exploit, and what are the costs associated with our choices?

As to the properties of this system, Section 3 is devoted to that, but most crucially we will see that this system possesses every resistance-to-control property possessed by even one of its constituents. And this will hold essentially due to the fact that *hybrid* is a close analog of the effect of a join: It splices the constituents together in such a way that key questions about the constituent systems can easily be many-one polynomial-time reduced ($\leq_m^{\mathrm{p}}$-reduced or reduced, for short) to questions about their hybrid.

As to why we chose this particular system, note that *hybrid* "switches" between constituent systems via wildly redundant information. This will let us keep deletions/partitions of voters/candidates from causing a switch between the underlying systems (if the starting state routed us to a nondefault case). Note that some other approaches that one might take are more sensitive to deletions. For example, suppose we wanted to hybridize just two election systems and decided to do so by using the first election system exactly if the first voter's most disliked candidate's name is lexicographically less than the first voter's second-most-disliked candidate's name. Note that if, as part of our control problem, that voter is deleted, that might suddenly change the system to which the problem is routed. Or, as another example, if we use the modulo k value of the name of the lexicographically smallest candidate to control switching between the k election systems, then that hybridization approach would be very sensitive to jumping between systems if, as part of our control problem, that candidate is deleted. These examples give some idea of why we

chose the approach we did, though admittedly even it can in some cases be nudged into jumping between systems—but at least this happens in very limited, very crisply delineated cases and in ways that we will generally be able to appropriately handle.

Finally, we come to what we allow ourselves to use to control the switching, what we choose not to use, and what price we pay for our choices. What we use (as is allowed in the [BTT92] model) are the candidates' names and only the candidates' names. We use absolutely nothing else to control switching between elections. We do not use voters' names. Indeed, in the [BTT92] model that we follow, voters (unlike candidates) do not even have names. But since the votes are input as a list, their ordering itself could be used to pass bits of information—e.g., we could look at whether the first vote in the list viewed as a string is lexicographically less than the last vote in the list viewed as a string. We in no way "cheat" by exploiting such input-order information, either for the votes or for the list of candidates (as per [BTT92], formally the candidate set is passed in separately to cover a certain boundary case). Our "switch" is based purely on candidates' names and just candidates' names. This also points to the price we pay for this choice: Even when all its constituent elections are candidate-anonymous, *hybrid* may not possess candidate-anonymity.

2.3 Types of Constructive and Destructive Control

Constructive control problems ask whether a certain class of actions by the election's chair can make a specified candidate the election's unique winner. Constructive control was first defined and studied by Bartholdi, Tovey, and Trick [BTT92]. Destructive control problems ask whether a certain class of actions by the election's chair can make a specified candidate fail to be a unique winner of the election. Destructive control was defined and studied by Hemaspaandra, Hemaspaandra, and Rothe [HHR05a], and in the different context of electoral manipulation destruction was introduced even earlier by Conitzer, Lang, and Sandholm [CS02,CLS03].

Bartholdi, Tovey, and Trick's [BTT92] groundbreaking paper defined seven types of electoral control. Among those seven, three are partition problems for which there are two different natural approaches to handling ties in subelections (see [HHR05a] which introduced these tie-handling models for this context): eliminating tied subelection winners (the "TE" model) or promoting tied subelection winners (the "TP" model). Thus, there are $(7-3)+2\cdot 3 = 10$ different standard types of constructive control, and there are essentially the same ten types of destructive control.

Since it is exceedingly important to not use a slightly different problem statement than earlier work whose results we will be drawing on, we will state the seven standard constructive control types (which become ten with the three partition control types each having both "TE" and "TP" versions) and their destructive analogs using word-for-word definitions from [HHR05a,HHR05b], which themselves are based closely and often identically on [BTT92] (see the

discussion in [HHR05a,HHR05b]).

Though V, the set of votes, is conceptually a multiset as in the previous related work, we take the view that the votes are input as a list ("the ballots"), and in particular are not directly input as a multiset in which cardinalities are input in binary (though we will mention later that our main result about *hybrid* holds also in that quite different model).

Constructive (Destructive) Control by Adding Candidates: Given a set C of qualified candidates and a distinguished candidate $c \in C$, a set D of possible spoiler candidates, and a set V of voters with preferences over $C \cup D$, is there a choice of candidates from D whose entry into the election would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Deleting Candidates: Given a set C of candidates, a distinguished candidate $c \in C$, a set V of voters, and a positive integer $k < \|C\|$, is there a set of k or fewer candidates in C whose disqualification would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Partition of Candidates: Given a set C of candidates, a distinguished candidate $c \in C$, and a set V of voters, is there a partition of C into C_1 and C_2 such that c is (not) the unique winner in the sequential two-stage election in which the winners in the subelection (C_1, V) who survive the tie-handling rule move forward to face the candidates in C_2 (with voter set V)?

Constructive (Destructive) Control by Run-Off Partition of Candidates: Given a set C of candidates, a distinguished candidate $c \in C$, and a set V of voters, is there a partition of C into C_1 and C_2 such that c is (not) the unique winner of the election in which those candidates surviving (with respect to the tie-handling rule) subelections (C_1, V) and (C_2, V) have a run-off with voter set V?

Constructive (Destructive) Control by Adding Voters: Given a set of candidates C and a distinguished candidate $c \in C$, a set V of registered voters, an additional set W of yet unregistered voters (both V and W have preferences over C), and a positive integer $k \leq \|W\|$, is there a set of k or fewer voters from W whose registration would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Deleting Voters: Given a set of candidates C, a distinguished candidate $c \in C$, a set V of voters, and a positive integer $k \leq \|V\|$, is there a set of k or fewer voters in V whose disenfranchisement would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Partition of Voters: Given a set of candidates C, a distinguished candidate $c \in C$, and a set V of voters, is there a partition of V into V_1 and V_2 such that c is (not) the unique winner in the hierarchical two-stage election in which the survivors of (C, V_1) and (C, V_2) run against each other with voter set V?

2.4 Immunity, Susceptibility, Vulnerability, Resistance

Again, to allow consistency with earlier papers and their results, we take this definition from [HHR05a,HHR05b], with the important exception regarding resistance discussed below Definition 2.2. It is worth noting that immunity and susceptibility both are "directional" (can we change *this*?) but that vulnerability and resistance are, in contrast, outcome-oriented (can we end up with *this* happening?) and complexity-focused.

Definition 2.2 *We say that a voting system is* immune to control *in a given model of control (e.g., "destructive control via adding candidates") if the model regards constructive control and it is never possible for the chair to by using his/her allowed model of control change a given candidate from being not a unique winner to being the unique winner, or the model regards destructive control and it is never possible for the chair to by using his/her allowed model of control change a given candidate from being the unique winner to not being a unique winner. If a system is not immune to a type of control, it is said to be* susceptible *to that type of control.*

A voting system is said to be (computationally) vulnerable to control *if it is susceptible to control and the corresponding language problem is computationally easy (i.e., solvable in polynomial time).*

A voting system is said to be resistant to control *if it is susceptible to control but the corresponding language problem is computationally hard (i.e.,* NP*-hard).*

We have diverged from all previous papers by defining resistance as meaning NP-hardness (i.e., NP-$\leq^{\mathrm{p}}_m$-hardness) rather than NP-completeness (i.e., NP-$\leq^{\mathrm{p}}_m$-completeness). In [BTT92], where the notion was defined, all problems were trivially in NP. But control problems might in difficulty exceed NP-completeness, and so the notion of resistance is better captured by NP-hardness.

An anonymous IJCAI referee commented that even polynomial-time algorithms can be expensive to run on sufficiently large inputs. We mention that though the comment is correct, almost any would-be controller would probably much prefer that challenge, solving a P problem on large inputs, to the challenge our results give him/her, namely, solving an NP-complete problem on large inputs. We also mention that since the hybrid scheme is designed so as to inherit resistances from the underlying schemes, if a hybrid requires extreme ratios between the number of candidates and the number of voters to display asymptotic hardness, that is purely due to inheriting that from the underlying systems. Indeed, if anything the hybrid is less likely to show that behavior since, informally put, if even one of the underlying systems achieves asymptotic hardness even away from extreme ratios between the number of candidates and the number of voters, then their hybrid will also.

2.5 Inheritance

We will be centrally concerned with the extent to which $hybrid(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ inherits the properties of its constituents. To do so, we formally define our

notions of inheritance (if all the constituents have a property then so does their hybrid) and of strong inheritance (if even one of the constituents has a property then so does the hybrid).

Definition 2.3 *We say that a property* Γ *is* strongly inherited *(respectively,* inherited*) by hybrid if the following holds: Let* $k \in \mathbb{N}^+$*. Let* $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ *be candidate-anonymous election systems (each taking as input* (C, V)*, with* V *a list of preference orders). It holds that hybrid*$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ *has property* Γ *if at least one of its constituents has (respectively, all its constituents have) property* Γ*.*

Definition 2.3 builds in the assumption that all constituents are candidate-anonymous. This assumption isn't overly onerous since as mentioned earlier candidate-anonymity is very common—but will be used in many of our proofs.

Though we will build candidate-anonymity into the assumptions underlying inheritance, we will often try to let interested readers know when that assumption is not needed. In particular, when we say "inherited (and flexibly so)" or "strongly inherited (and flexibly so)," the "(and flexibly so)" indicates that the claim holds even if in Definition 2.3 the words "candidate-anonymous" are deleted. For example, the following easy but quite important claim follows easily from the definition of *hybrid*.

Proposition 2.4 *"Winner problem membership in* P*," "unique winner problem membership in* P*," "winner problem membership in* NP*," and "unique winner problem membership in* NP *" are inherited (and flexibly so) by hybrid.*

3 Inheritance and Hybrid Elections: Results

In this section we will discuss the inheritance properties of *hybrid* with respect to susceptibility, resistance, immunity, and vulnerability. Table 1 summarizes our results for the cases of constructive control and destructive control. (This table does not discuss/include the issue of when "(and flexibly so)" holds, i.e., when the candidate-anonymity assumption is not needed, but rather focuses on our basic inheritance definition.)

3.1 Susceptibility

We first note that susceptibility strongly inherits. We remind the reader that throughout this paper, when we speak of an election system, we always implicitly mean an election system that takes as input (C, V) with V a list of preference orders over C.

Theorem 3.1 *Let* $k \in \mathbb{N}^+$ *and let* $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ *be election systems. Let* Φ *be one of the standard twenty types of (constructive and destructive) control. If for at least one* i*,* $0 \leq i \leq k-1$*,* $\mathcal{E}_i$ *is candidate-anonymous and susceptible to* Φ*, then hybrid*$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ *is susceptible to* Φ*.*

Control by	Susceptibility	Resistance	Immunity	Vulnerability
Adding Candidates	SI	SI	Not I / I*	I
Deleting Candidates	SI	SI	I / Not I*	I iff P = NP
Partition of Candidates (TE)	SI	SI	Not I	On ($\star\star$) systems: I iff SI iff P = NP
Partition of Candidates (TP)	SI	SI	Not I	On ($\star$) systems: I iff SI iff P = NP
Run-off Partition of Candidates (TE)	SI	SI	Not I	On ($\star\star$) systems: I iff SI iff P = NP
Run-off Partition of Candidates (TP)	SI	SI	Not I	On ($\star$) systems: I iff SI iff P = NP
Adding Voters	SI	SI	I	I
Deleting Voters	SI	SI	I	I
Partition of Voters (TE)	SI	SI	I	I
Partition of Voters (TP)	SI	SI	I	I

Table 1: Inheritance results that hold or provably fail for *hybrid*. Key: I = Inherits. SI = Strongly Inherits. Boxes without a * state results for both constructive and destructive control. In boxes with a *, the * refers to the destructive control case. "On ($\star$) systems" is a shorthand for "On election systems having winner problems in the polynomial hierarchy." "On ($\star\star$) systems" is a shorthand for "On election systems having unique winner problems in the polynomial hierarchy."

Corollary 3.2 *hybrid strongly inherits susceptibility to each of the standard twenty types of control.*

3.2 Resistance

We now come to the most important inheritance case, namely, that of resistance. Since our hope is that hybrid elections will broaden resistance, the ideal case would be to show that resistance is strongly inherited. And we will indeed show that, and from it will conclude that there exist election systems that are resistant to all twenty standard types of control.

We first state the key result, which uses the fact that *hybrid* can embed its constituents to allow us to $\leq_m^{\mathrm{p}}$-reduce from control problems about its constituents to control problems about *hybrid*.

Theorem 3.3 *Let $k \in \mathbb{N}^+$ and let $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ be election systems. Let Φ be one of the standard twenty types of (constructive and destructive) control. If for at least one i, $0 \leq i \leq k-1$, $\mathcal{E}_i$ is candidate-anonymous and resistant to Φ, then hybrid$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ is resistant to Φ.*

Corollary 3.4 *hybrid strongly inherits resistance to each of the standard twenty types of control.*

Before we turn to applying this corollary, let us note that Theorem 3.3 and Corollary 3.4 are both, as is this entire paper, within the most natural, most typical model: Votes are input as a list ("nonsuccinct" input) and each vote counts equally ("unweighted" votes). We mention that for each of the other three cases—"succinct, weighted," "succinct, unweighted," and "nonsuccinct, weighted"—Theorem 3.3 and Corollary 3.4 both still hold.

Let us apply Corollary 3.4 to obtain election systems that are broadly resistant to control.

Corollary 3.5 *There exist election systems—for example, hybrid*(plurality, Condorcet)*—that are resistant to all the standard ten types of constructive control.*

To make the same claim for destructive control, a bit more work is needed, since for three of the standard ten types of destructive control no system has been, as far as we know, proven to be resistant. So we first construct an artificial system, $\mathcal{E}_{\text{not-all-one}}$ (defined in the full version), having the missing three resistance properties.

Lemma 3.6 *There exists a candidate-anonymous election system, $\mathcal{E}_{\text{not-all-one}}$, that is resistant to (a) destructive control by deleting voters, (b) destructive control by adding voters, and (c) destructive control by partition of voters in the TE model.*

Corollary 3.7 *There exist election systems that are resistant to all ten standard types of destructive control.*

We cannot apply Theorem 3.3 directly to rehybridize the systems of Corollaries 3.5 and 3.7, because *hybrid* itself is not in general candidate-anonymous. However, we can get the same conclusion by directly hybridizing all the constituents underlying Corollaries 3.5 and 3.7.

Theorem 3.8 *There exist election systems that are resistant to all twenty standard types of control.*

The proof simply is to consider $\mathit{hybrid}(\text{plurality}, \text{Condorcet}, \mathcal{E}_{\text{not-all-one}})$.

3.3 Immunity

We now turn to inheritance of immunity. Here, for each of constructive and destructive control, five cases inherit and five cases provably fail to inherit.

Theorem 3.9 *Any candidate-anonymous election system that is immune to constructive control by deleting candidates can never have a unique winner.*

Since "never having a unique winner" is inherited by *hybrid*, Theorem 3.9 implies:

Theorem 3.10 *Immunity to constructive control by deleting candidates is inherited by hybrid.*

By applying a duality result of Hemaspaandra, Hemaspaandra, and Rothe multiple times, we can retarget this to a type of destructive control.

Proposition 3.11 ([HHR05b]) *A voting system is susceptible to constructive control by deleting candidates if and only if it is susceptible to destructive control by adding candidates.*

Corollary 3.12 *Immunity to destructive control by adding candidates is inherited by hybrid.*

hybrid's immunity to all voter-related types of control is immediate.

Theorem 3.13 *Immunity to constructive and destructive control under each of (a) adding voters, (b) deleting voters, (c) partition of voters in model TE, and (d) partition of voters in model TP is inherited (and flexibly so) by hybrid.*

For the ten remaining cases, inheritance does not hold.

3.4 Vulnerability

hybrid strongly inherited resistance, which is precisely what one wants, since that is both the aesthetically pleasing case and broadens resistance to control. However, for vulnerability it is less clear what outcome to root for. Inheritance would be the mathematically more beautiful outcome. But on the other hand, what inheritance would inherit is vulnerability, and vulnerability to control is in general a bad thing—so maybe one should hope for "Not I(nherits)" entries for our table in this column. In fact, our results here are mixed. In particular, we for ten cases prove that inheritance holds unconditionally and for ten cases prove that inheritance holds (though in some cases we have to limit ourselves to election systems with winner/unique winner problems that fall into the polynomial hierarchy) if and only if P = NP.

4 Conclusions

Table 1 summarizes our inheritance results. The main contribution of this paper is the *hybrid* system, the fact that *hybrid* strongly inherits resistance, and the consequence that there is an election system that resists all twenty standard types of electoral control. The authors jointly with P. Faliszewski are currently working to show that some natural election systems may exhibit broad resistance to control.

Acknowledgments: We thank Holger Spakowski, COMSOC '06 referees, and IJCAI '07 referees for helpful comments.

References

[Arr63] K. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 1951 (revised edition 1963).

[BO91] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[BTT89a] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT89b] 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(2):157–165, 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.

[CLS03] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 201–214. ACM Press, 2003.

[CS02] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 314–319. AAAI Press, 2002.

[CS03] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 781–788. Morgan Kaufmann, August 2003.

[DKNS01] 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.

[DS00] J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

[EL05] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Proceedings of the 16th International Symposium on Algorithms and Computation*, pages 206–215. Springer-Verlag *Lecture Notes in Computer Science #3827*, December 2005.

[ER91] E. Ephrati and J. Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proceedings of the 9th National Conference on Artificial Intelligence*, pages 173–178. AAAI Press, 1991.

[ER93] E. Ephrati and J. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 423–429, 1993.

[FHH06] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 641–646. AAAI Press, July 2006.

[FKS03] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 301–312. ACM Press, 2003.

[Gib73] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[HHR05a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 95–101. AAAI Press, 2005.

[HHR05b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. Technical Report cs.GT/0507027, Computing Research Repository, http://www.acm.org/repository/, July 2005. Revised, May 2006.

[HHR06] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report TR-900, Department of Computer Science, University of Rochester, Rochester, NY, June 2006. Revised, August 2006. Conference version to appear in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*.

[PHG00] 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.

[Sat75] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[SV00] H. Spakowski and J. Vogel. Θ_2^p-completeness: A classical approach for new results. In *Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 348–360. Springer-Verlag *Lecture Notes in Computer Science #1974*, December 2000.

[Tay05] A. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.

Edith Hemaspaandra
Department of Computer Science, Rochester Institute of Technology
Rochester, NY 14623, USA, `eh` "at" `cs.rit.edu`.

Lane A. Hemaspaandra
Department of Computer Science, University of Rochester
Rochester, NY 14627, USA, `lane` "at" `cs.rochester.edu`.

Jörg Rothe
Institut für Informatik, Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany, `rothe` "at" `cs.uni-duesseldorf.de`.

Decentralization and Mechanism Design for Online Machine Scheduling[1]

Birgit Heydenreich[2] and Rudolf Müller and Marc Uetz

Abstract

We study the online version of the classical parallel machine scheduling problem to minimize the total weighted completion time from a new perspective: We assume that the data of each job, namely its release date r_j, its processing time p_j and its weight w_j is only known to the job itself, but not to the system. Furthermore, we assume a decentralized setting where jobs choose the machine on which they want to be processed themselves. We study this problem from the perspective of algorithmic mechanism design. We introduce the concept of a myopic best response equilibrium, a concept weaker than the dominant strategy equilibrium, but appropriate for online problems. We present a polynomial time, online scheduling mechanism that, assuming rational behavior of jobs, results in an equilibrium schedule that is 3.281-competitive. The mechanism deploys an online payment scheme that induces rational jobs to truthfully report their private data. We also show that the underlying local scheduling policy cannot be extended to a mechanism where truthful reports constitute a dominant strategy equilibrium.

1 Introduction

We study the online version of the classical parallel machine scheduling problem to minimize the total weighted completion time[3] from a new perspective: We assume a strategic setting, where the data of each job, namely its release date r_j, its processing time p_j and its weight w_j is only known to the job itself, but not to the system. Any job j is interested in being finished as early as possible, and the weight w_j represents its indifference cost for spending one additional unit of time waiting. The time when job j is finished is called its completion time C_j. While jobs may strategically report false values $(\tilde{r}_j, \tilde{p}_j, \tilde{w}_j)$ in order to be scheduled earlier, the total social welfare is maximized whenever the weighted sum of completion times $\sum w_j C_j$ is minimized. Furthermore, we assume a restricted communication paradigm, referred to as *decentralization*: Jobs may communicate with machines, but neither do jobs communicate with each other, nor do machines communicate with each other. In particular, there is no central coordination authority hosting all the data of the problem. This

[1]A version of this paper has been published in the Proceedings of the Scandinavian Workshop on Algorithm Theory (SWAT) 2006, LNCS 4059, pp. 136-147, Springer

[2]Supported by NWO grant 2004/03545/MaGW 'Local Decisions in Decentralised Planning Environments'.

[3]The problem is $\mathrm{P} \,|\, r_j \,|\, \sum w_j C_j$ in the notation of Graham et al. [1].

leads to a setting where the jobs themselves must select the machine to be processed on, and any machine sequences the jobs according to a (known) local sequencing policy.

The problem $P \mid r_j \mid \sum w_j C_j$ is well-understood in the non-strategic setting with centralized coordination. First, scheduling to minimize the weighted sum of completion times with release dates is NP-hard, even in the off-line case [2]. Second, no online algorithm for the single machine problem can be better than 2-competitive [3] regardless of the question whether or not P=NP, and lower bounds exist for parallel machines, too [4]. The best possible algorithm for the single machine case is 2-competitive [5]. For the parallel machine setting, the currently best known online algorithm is 2.61-competitive [6].

In the strategic setting, selfish agents trying to maximize their own benefit can do so by reporting strategically about their private information, thus manipulating the resulting schedule. In the model we propose, a job can report an arbitrary weight, an elongated processing time (e.g. by adding unnecessary work), and it can artificially delay its true release date r_j. We do not allow a job to report a processing time shorter than p_j, as this can easily be discovered and punished by the system, e.g. by preempting the job after the declared processing time $\tilde{p}_j$ before it is actually finished. Furthermore, as we assume that any job j comes into existence only at its release date r_j, it obviously does not make sense that a job reports a release date smaller than the true value r_j.

Our goal is to set up a mechanism that yields a reasonable overall performance with respect to the objective function $\sum w_j C_j$. To that end, the mechanism needs to motivate the jobs to reveal their private information truthfully. In addition, as we require decentralization, each machine must be equipped with a local sequencing policy that is publicly known, and jobs must be induced to select the machines in such a way that $\sum w_j C_j$ is not too large. Known algorithms with the best performance ratio, e.g. [6, 7], crucially require central coordination to distribute jobs over machines. An approach by Megow et al. [8], developed for an online setting with release dates and stochastic job durations, however, turns out to be appropriate for being adopted to the decentralized, strategic setting.

Related Work and Contribution. Mechanism design in combination with the design of approximation algorithms for scheduling problems has been studied, e.g., by Nisan and Ronen [10], Archer and Tardos [11], and Kovacs [12]. In those papers, not the jobs but the machines are the selfishly behaving parts of the system, and their private information is the time they need to process the jobs. A scheduling model where the jobs are the selfish agents of the system has been studied by Porter [13]. He addresses a single machine scheduling problem, where the private data of each job consists of a release date, its processing time, its weight, and a deadline. In all mentioned papers, the only action of an agent (machine or job, respectively) is to reveal its private data; the resulting mechanisms are also called direct revelation mechanisms. The mechanism suggested in this paper is not a direct revelation mechanism, since in addition to the revelation of private data, jobs must select the machine

to be processed on.

In the algorithm of Megow et al. [8], jobs are locally sequenced according to an online variant of the well known WSPT rule [9], and arriving jobs are assigned to machines in order to minimize an expression that approximates the (expected) increase of the objective value. This algorithm achieves a performance ratio of 3.281. The mechanism we propose develops their idea further. We present a polynomial time, decentralized online mechanism, called Decentralized LocalGreedy Mechanism. Thereby we provide also a new algorithm for the non-strategic, centralized setting, inspired by the MinIncrease Algorithm of [8], but improving upon the latter in terms of simplicity. We show that the Decentralized LocalGreedy Mechanism is 3.281-competitive as well. The currently best known bound for the non-strategic setting is 2.61 [6].

As usual in mechanism design, the Decentralized LocalGreedy Mechanism defines *payments* that have to be made by the jobs for being processed. Naturally, we require from an *online* mechanism that also the payments are computed online. Hence they can be completely settled by the time at which a job leaves the system. We also show that the payments result in a balanced budget. The payments induce the jobs to select 'the right' machines. Intuitively, the mechanism uses the payments to mimic a corresponding LocalGreedy online algorithm in the classical (non-strategic, centralized) parallel machine setting $\mathrm{P} \mid r_j \mid \sum w_j C_j$. Moreover, the payments induce rational jobs to truthfully report about their private data. With respect to release dates and processing times, we can show that truthfulness is a dominant strategy equilibrium. With respect to the weights, however, we can only show that truthful reports are myopic best responses (in a sense to be made precise later). In addition, we show that there does not exist a payment scheme extending the allocation rule of the Decentralized LocalGreedy Mechanism to a mechanism where truthful reporting of all private information is a dominant strategy equilibrium.

This extended abstract is organized as follows. We formalize the model and introduce the required notation in Section 2. In Section 3 the LocalGreedy algorithm is defined. In Section 4, this algorithm is adapted to the strategic setting and extended by a payment scheme, yielding the Decentralized LocalGreedy Mechanism. Moreover, our main results are presented in that section. We analyze the performance of the mechanism in Section 5, mention a negative result in Section 6, and conclude with a short discussion in Section 7.

2 Model and Notation

The considered problem is online parallel machine scheduling with non-trivial release dates, with the objective to minimize the weighted sum of completion times. We are given a set of jobs $J = \{1, \ldots, n\}$, where each job needs to be processed on any of the parallel, identical machines from the set $M = \{1, \ldots, m\}$. The processing of each job must not be preempted, and each machine can pro-

cess at most one job at a time. Each job j is viewed as a selfish agent and has the following private information: a release date $r_j \geq 0$, a processing time $p_j > 0$, and an indifference cost, or weight, denoted by $w_j \geq 0$. The release date denotes the time when the job comes into existence, whereas the weight represents the cost to a job for one additional unit of time spent waiting. Without loss of generality, we assume that the jobs are numbered in order of their release dates, i.e., $j < k \Rightarrow r_j \leq r_k$. The triple (r_j, p_j, w_j) is also denoted as the *type* of a job, and we use the shortcut notation $t_j = (r_j, p_j, w_j)$. By $T = \mathbb{R}_0^+ \times \mathbb{R}^+ \times \mathbb{R}_0^+$ we denote the space of possible types of each job.

Definition 1. *A* decentralized online scheduling mechanism *is a procedure that works as follows.*

1. *Each job j has a release date r_j, but may pretend to come into existence at any time $\tilde{r}_j \geq r_j$. At that chosen release date, the job communicates to every machine reports $\tilde{w}_j$ and $\tilde{p}_j$ (which may differ from the true w_j and p_j)*[4].
2. *Machines communicate on the basis of that information a (tentative) completion time $\hat{C}_j$ and a (tentative) payment $\hat{\pi}_j$ to the job. This information is tentative due to the online situation. The values $\hat{C}_j$ and $\hat{\pi}_j$ can only change if later another job chooses the same machine.*
3. *Based on this response, the job chooses a machine. This choice is binding. The entire communication takes place at one point in time, namely $\tilde{r}_j$.*
4. *There is no communication between machines or between jobs.*
5. *Depending on later arrivals of jobs, machines may revise $\hat{C}_j$ and $\hat{\pi}_j$. Eventually, the mechanism leads to an (ex-post) completion time C_j and an (ex-post) payment π_j of each job.*

Hereby, we assume that jobs with equal reported release date arrive in some given order and communicate to machines in that order. Next, we define an online property of the payment scheme and the performance ratio of an online mechanism.

Definition 2. *If in a decentralized online scheduling mechanism for every job j payments to and from j are only made between time $\tilde{r}_j$ and time C_j, then we call the payment scheme of the mechanism an* online payment scheme.

Definition 3. *Let A be an online mechanism that seeks to minimize a certain objective function. Let $V_A(I)$ be the objective value computed by A for an instance I and let $V_{OPT}(I)$ be the offline optimal objective value for I. Then A is called ϱ-*competitive *if for all instances I*

$$V_A(I) \leq \varrho \cdot V_{OPT}(I).$$

[4] A job could even report different values to different machines. However, we prove existence of equilibria where the jobs do not make use of that option.

The factor ϱ is also called performance ratio *of the mechanism.*

We assume that each job j prefers a lower completion time to a higher one and model this by the valuation $v_j(C_j \mid t_j) = -w_j\, C_j$. We assume *quasi-linear utilities*, that is, the utility of job j equals $u_j(C_j, \pi_j \mid t_j) = v_j(C_j \mid t_j) - \pi_j$, which is equal to $-w_j\, C_j - \pi_j$. In this model, the utility u_j is always negative. Therefore, we assume that a job has a constant and sufficiently large utility for 'being processed at all'. Note that the total social welfare is maximized whenever the weighted sum of completion times $\sum_{j \in J} w_j\, C_j$ is minimum, which is independent of whether we do or do not carry these constants with us.

The communication with machines, and the decision for a particular machine are called *actions* of the jobs; they constitute the strategic actions jobs can take in the non-cooperative game induced by the mechanism. A *strategy* s_j of a job j maps a type t_j to an action for every possible state of the system in which the job is required to take some action. A strategy profile is a vector $(s_1, \ldots, s_n)$ of strategies, one for each job. Given a mechanism, a strategy profile, and a realization of types t, we denote by $u_j(s, t)$ the utility that agent j receives.

Definition 4. *A strategy profile $s = (s_1, \ldots, s_n)$ is called a* dominant strategy equilibrium *if for all jobs $j \in J$, all types t of the jobs, all strategies $\tilde{s}_{-j}$ of the other jobs, and all strategies $\tilde{s}_j$ that j could play instead of s_j,*

$$u_j((s_j, \tilde{s}_{-j}), t) \geq u_j((\tilde{s}_j, \tilde{s}_{-j}), t)\,.$$

We could simplify notation if we restricted ourselves to *direct revelation mechanisms*, that is mechanisms in which the only action of a job is to report its type. However, a decentralized online scheduling mechanism requires that jobs decide themselves on which machine they are scheduled. Since these decisions are likely to influence the utility of the jobs, they have to be modelled as actions in the game. Therefore, it is not sufficient to restrict oneself to direct revelation mechanisms.

We will see that the mechanism proposed in this paper does not have a dominant strategy equilibrium, whatever modification we might apply to the payment scheme. However, a weaker equilibrium concept applies, which we define next. That definition uses the concept of the tentative utility, i.e., the utility a job would have if it was the last to be accepted on its machine.

Definition 5. *Given a decentralized, online scheduling mechanism as in Definition 1, a strategy profile s, and type profile t. Let $\hat{C}_j$ and $\hat{\pi}_j$ denote the tentative completion time and the tentative payment of job j at time $\tilde{r}_j$. Then $\hat{u}_j(s, t) := \hat{C} w_j - \hat{\pi}_j$ denotes j's tentative utility at time $\tilde{r}_j$.*

If s and t are clear from the context, we will use $\hat{u}_j$ as short notation.

Definition 6. *A strategy profile $(s_1, \ldots, s_n)$ is called a* myopic best response equilibrium*, if for all jobs $j \in J$, all types t of the jobs, all strategies $\tilde{s}_{-j}$ of the other jobs and all strategies $\tilde{s}_j$ that j could play instead of s_j,*

$$\hat{u}_j((s_j, \tilde{s}_{-j}), t) \geq \hat{u}_j((\tilde{s}_j, \tilde{s}_{-j}), t).$$

2.1 Critical jobs

For convenience of presentation, we make the following assumption for the main part of the paper. Fix some constant $0 < \alpha \leq 1$ (α will be discussed later). Let us call job j *critical* if $r_j < \alpha p_j$. Intuitively, a job is critical if it is long and appears comparably early in the system. The assumption we make is that such critical jobs do not exist, that is

$$r_j \geq \alpha\, p_j \quad \text{for all jobs } j \in J\,.$$

This assumption is a tribute to the desired performance guarantee, and in fact, it is well known that critical jobs must not be scheduled early to achieve constant performance ratios [5, 7]. However, the assumption is only made due to cosmetic reasons. In the following we first define an algorithm and a mechanism on the refined type space, where all jobs are non-critical. In Section 5.1, we extend the type space and slightly adapt the mechanism such that also critical jobs can be dealt with. This slight adaption leads to a constant performance bound while preserving all desired properties concerning the strategic behaviour of the jobs.

3 The LocalGreedy Algorithm

We next formulate an online scheduling algorithm that is inspired by the Min-Increase Algorithm from Megow et al. [8]. For the time being, we assume that the job characteristics, namely release date r_j, processing time p_j and indifference cost w_j, are given. In the next section, we discuss how to turn this algorithm into a mechanism for the strategic, decentralized setting that we aim at.

The idea of the algorithm is that each machine uses (an online version of) the well known WSPT rule [9] locally. More precisely, each machine implements a priority queue containing the not yet scheduled jobs that have been assigned to the machine. The queue is organized according to WSPT, that is, jobs with higher ratio w_j/p_j have higher priority. In case of ties, jobs with lower index have higher priority. As soon as the machine falls idle, the currently first job from this priority queue is scheduled (if any). Given this local scheduling policy on each of the machines, any arriving job is assigned to that machine where the increase in the objective $\sum w_j\, C_j$ is minimal.

In the formulation of the algorithm, we utilize some shortcut notation. We let $j \to i$ denote the fact that job j is assigned to machine i. Let S_j be the time when job j eventually starts being processed. For any job j, $H(j)$ denotes the set of jobs that have higher priority than j, $H(j) = \{k \in J \mid w_k p_j > w_j p_k\} \cup \{k \leq j \mid w_k p_j = w_j p_k\}$. Note that $H(j)$ includes j, too. Similarly, $L(j) = J \setminus H(j)$ denotes the set of jobs with lower priority. At a given point t in time, machine i might be busy processing a job. We let $b_i(t)$ denote the remaining processing time of that job at time t, i.e., at time t machine i will be blocked during $b_i(t)$ units of time for new jobs. If machine i is idle at time t, we let $b_i(t) = 0$.

Algorithm 1: LocalGreedy algorithm

Local Sequencing Policy:
Whenever a machine becomes idle, it starts processing the job with highest (WSPT) priority among all jobs assigned to it.
Assignment:
(1) At time r_j job j arrives; the immediate increase of the objective $\sum w_j C_j$, given that j is assigned to machine i, is

$$z(j,i) := w_j \Big[r_j + b_i(r_j) + \sum_{\substack{k \in H(j) \\ k \to i \\ k<j \\ S_k \geq r_j}} p_k + p_j \Big] + p_j \sum_{\substack{k \in L(j) \\ k \to i \\ k<j \\ S_k > r_j}} w_k.$$

(2) Job j is assigned to machine $i_j \in \operatorname{argmin}_{i \in M} z(j,i)$ with minimum index.

Clearly, the LocalGreedy algorithm still makes use of central coordination in Step (2). In the sequel we will introduce payments that allow to transform the algorithm into a decentralized online scheduling mechanism.

4 Payments for Myopic Rational Jobs

The payments we introduce can be motivated as follows: A job j pays at the moment of its placement on one of the machines an amount that compensates the decrease in utility of the other jobs. The final payment of each job j resulting from this mechanism will then consist of the immediate payment j has to make when selecting a machine and of the payments j receives when being displaced by other jobs. We will prove that utility maximizing jobs have an incentive to report truthfully and to choose the machine that the LocalGreedy Algorithm would have selected, too. Furthermore, the WSPT rule can be run locally on every machine and does not require communication between the machines. We will see in the next section that this yields a constant-factor approximation of the off-line optimum, given that the jobs behave rationally. The algorithm including the payments is displayed below as the Decentralized LocalGreedy Mechanism. Let the indices of the jobs be defined according to the reported release dates, i.e. $j < k \Rightarrow \tilde{r}_j \leq \tilde{r}_k$. Let $\tilde{H}(j)$ and $\tilde{L}(j)$ be defined analogously to $H(j)$ and $L(j)$ on the basis of the reported weights.

Algorithm 2: DecentralizedLocalGreedy Mechanism

Local Sequencing Policy:
Whenever a machine becomes idle, it starts processing the job with highest (WSPT) priority among all available jobs queuing at this machine.
Assignment:
(1) At time $\tilde{r}_j$ job j arrives and reports a weight $\tilde{w}_j$ and a processing time $\tilde{p}_j$ to all machines.

(2) Every machine i computes

$$\hat{C}_j(i) = \tilde{r}_j + b_i(\tilde{r}_j) + \sum_{\substack{k \in \tilde{H}(j) \\ k \to i \\ k<j \\ S_k \geq \tilde{r}_j}} \tilde{p}_k + \tilde{p}_j \quad \text{and} \quad \hat{\pi}_j(i) = \tilde{p}_j \sum_{\substack{k \in \tilde{L}(j) \\ k \to i \\ k<j \\ S_k > \tilde{r}_j}} \tilde{w}_k.$$

and informs j about both $\hat{C}_j(i)$ and $\hat{\pi}_j(i)$.
(3) Job j chooses a machine $i_j \in M$. Its tentative utility for being queued at machine i is $\hat{u}_j(i) := -w_j \hat{C}_j(i) - \hat{\pi}_j(i)$.
(4) The job is queued at i_j according to WSPT among all currently available jobs on i_j whose processing has not started yet. The payment $\hat{\pi}_j(i_j)$ has to be paid by j.
(5) The (tentative) completion time for every job k with $k \in \tilde{L}(j)$, $k \to i_j$, $k < j$, $S_k > \tilde{r}_j$ increases by $\tilde{p}_j$ due to j's presence. As compensation, k receives a payment of $\tilde{w}_k \tilde{p}_j$.

The DecentralizedLocalGreedy Mechanism together with the stated payments results in a balanced budget for the scheduler. That is, the payments paid and received by the jobs sum up to zero, since every arriving job immediately makes its payment to the jobs that are displaced by it. Notice that the payments are made online in the sense of Definition 2.

Theorem 7. *Regard any type vector t, any strategy profile s and any job j such that j reports $(\tilde{r}_j, \tilde{p}_j, \tilde{w}_j)$ and chooses machine $\tilde{m} \in M$. Then changing the report to $(\tilde{r}_j, \tilde{p}_j, w_j)$ and choosing a machine that maximizes its tentative utility at time $\tilde{r}_j$ does not decrease j's tentative utility under the* Decentralized LocalGreedy *Mechanism.*

Proof. We only give the idea here. For the single machine case, an arriving job j gains tentative utility $\tilde{p}_k w_j - \tilde{p}_j \tilde{w}_k$ from displacing an already present job k. WSPT assigns j in front of k if and only if $\tilde{p}_k \tilde{w}_j - \tilde{p}_j \tilde{w}_k > 0$. Thus, $\tilde{w}_j = w_j$ maximizes j's tentative utility. For $m > 1$, the theorem follows from the fact that j can select a machine itself. □

Lemma 8. *Consider any job $j \in J$. Then, under the* Decentralized LocalGreedy *Mechanism, for all reports of all other agents as well as all choices of machines of the other agents, the following is true:*
(a) If j reports $\tilde{w}_j = w_j$, then the tentative utility when queued at any of the machines will be preserved over time, i.e. it equals j's ex-post utility.
(b) If j reports $\tilde{w}_j = w_j$, then selecting the machine that the LocalGreedy *Algorithm would have selected maximizes j's ex-post utility.*

Proof. See full version of the paper. □

Theorem 9. *Consider the restricted strategy space where all $j \in J$ report $\tilde{w}_j = w_j$. Then the strategy profile where all jobs j truthfully report $\tilde{r}_j = r_j$, $\tilde{p}_j = p_j$ and choose a machine that maximizes $\hat{u}_j$ is a dominant strategy equilibrium under the* Decentralized LocalGreedy *Mechanism.*

Proof. Let us start with $m = 1$. Suppose $\tilde{w}_j = w_j$, fix any pretended release date $\tilde{r}_j$ and regard any $\tilde{p}_j > p_j$. Let u_j denote j's (ex-post) utility when reporting p_j truthfully and let $\tilde{u}_j$ be its (ex-post) utility for reporting $\tilde{p}_j$. As $\tilde{w}_j = w_j$, the ex-post utility equals in both cases the tentative utility at decision point $\tilde{r}_j$ according to Lemma 8(a). Let us therefore regard the latter utilities. Clearly, according to the WSPT-priorities, j's position in the queue at the machine for report p_j will not be behind its position for report $\tilde{p}_j$. Let us divide the jobs already queuing at the machine upon j's arrival into three sets: Let $J_1 = \{k \in J \,|\, k < j,\, S_k > \tilde{r}_j,\, \tilde{w}_k/\tilde{p}_k \geq w_j/p_j\}$, $J_2 = \{k \in J \,|\, k < j,\, S_k > \tilde{r}_j,\, w_j/p_j > \tilde{w}_k/\tilde{p}_k \geq w_j/\tilde{p}_j\}$ and $J_3 = \{k \in J \,|\, k < j,\, S_k > \tilde{r}_j,\, w_j/\tilde{p}_j > \tilde{w}_k/\tilde{p}_k\}$. That is, J_1 comprises the jobs that are in front of j in the queue for both reports, J_2 consists of the jobs that are only in front of j when reporting $\tilde{p}_j$ and J_3 includes only jobs that queue behind j for both reports. Therefore, $\tilde{u}_j - u_j$ equals

$$\begin{aligned} & - \sum_{k \in J_1 \cup J_2} w_j \tilde{p}_k - \sum_{k \in J_3} \tilde{p}_j \tilde{w}_k - w_j \tilde{p}_j - \left(- \sum_{k \in J_1} w_j \tilde{p}_k - \sum_{k \in J_2 \cup J_3} p_j \tilde{w}_k - w_j p_j \right) \\ = & \sum_{k \in J_2} (p_j \tilde{w}_k - w_j \tilde{p}_k) - \sum_{k \in J_3} (\tilde{p}_j - p_j) \tilde{w}_k - w_j (\tilde{p}_j - p_j). \end{aligned}$$

According to the definition of J_2, the first term is smaller than or equal to zero. As $\tilde{p}_j > p_j$, the whole right hand side becomes non-positive. Therefore $\tilde{u}_j \leq u_j$, i.e. truthfully reporting p_j maximizes j's ex-post utility on a single machine.

Let us now fix $\tilde{w}_j = w_j$ and any $\tilde{p}_j \geq p_j$ and regard any false release date $\tilde{r}_j > r_j$. There are two effects that can occur when arriving later than r_j. Firstly, jobs queued at the machine already at time r_j may have been processed or may have started receiving service by time $\tilde{r}_j$. But either j would have had to wait for those jobs anyway or it would have increased its immediate utility at decision point r_j by displacing a job and paying the compensation. So, j cannot gain from this effect by lying. The second effect is that new jobs have arrived at the machine between r_j and $\tilde{r}_j$. Those jobs either delay j's completion time and j looses the payment it could have received from those jobs by arriving earlier. Or the jobs do not delay j's completion time, but j has to pay the jobs for displacing them when arriving at $\tilde{r}_j$. If j arrived at time r_j, it would not have to pay for displacing such a job. Hence, j cannot gain from this effect either. Thus the immediate utility at time r_j will be at least as large as its immediate utility at time $\tilde{r}_j$. Therefore, j maximizes its immediate utility at time $\tilde{r}_j$ by choosing $\tilde{r}_j = r_j$. As $\tilde{w}_j = w_j$, it follows from Lemma 8(a) that choosing $\tilde{r}_j = r_j$ also maximizes the job's ex-post utility on a single machine.

For $m > 1$, note that on every machine, the immediate utility of job j at decision point $\tilde{r}_j$ is equal to its ex-post utility and that j can select a machine itself that maximizes its immediate utility and therefore its ex-post utility. Therefore, given that $\tilde{w}_j = w_j$, a job's ex-post utility is maximized by choosing $\tilde{r}_j = r_j$, $\tilde{p}_j = p_j$ and, according to Lemma 8(b), by choosing a machine that minimizes the immediate increase in the objective function. □

Theorem 10. *Given the types of all jobs, the strategy profile where each job j reports $(\tilde{r}_j, \tilde{p}_j, \tilde{w}_j) = (r_j, p_j, w_j)$ and chooses a machine maximizing its tentative utility $\hat{u}_j$ is a myopic best response equilibrium under the* DECENTRALIZED LOCALGREEDY *Mechanism.*

Proof. Regard job j. According to the proof of Theorem 7, $\hat{u}_j$ on any machine is maximized by reporting $\tilde{w}_j = w_j$ for any $\tilde{r}_j$ and $\tilde{p}_j$. According to Theorem 9 and Lemma 8(b), $\tilde{p}_j = p_j$, $\tilde{r}_j = r_j$ and choosing a machine that maximizes j's tentative utility at time $\tilde{r}_j$ maximize j's ex-post utility if j truthfully reports $\tilde{w}_j = w_j$. According to Lemma 8(a) this ex-post utility is equal to $\hat{u}_j$ if j reports $\tilde{w}_j = w_j$. Therefore, any job j maximizes $\hat{u}_j$ by truthful reports and choosing the machine as claimed. □

Given the restricted communication paradigm, jobs do not know at their arrival which jobs are already queuing at the machines and what reports the already present jobs have made. Therefore it is easy to see that for any non-truthful report of an arriving job about its weight, instances can be constructed in which this report yields a strictly lower utility for the job than a truthful report would have given. With arguments similar to those in the proof of Theorem 9, the same holds for false reports about the processing time and the release date.

Note that in order to obtain the myopic best response equilibrium (Theorem 10), payments paid by an arriving job j need not necessarily be given to the jobs delayed by j. But by doing so, the resulting ex-post payments result in a balanced budget and the tentative utility at arrival is preserved and equals the ex-post utility of every job (Lemma 7). Furthermore, paying jobs for their delay results in a dominant strategy equilibrium in a restricted type space (Theorem 9).

5 Performance of the Mechanism

As shown in Section 4, jobs have a motivation to report truthfully about their data: According to Theorem 7, it is a myopic best response for a job j to report the true weight w_j, no matter what the other jobs do and no matter which $\tilde{p}_j$ and $\tilde{r}_j$ are reported by j itself. Given a true report of w_j, it was proven in Theorem 9 that reporting the true processing time and release date as well as choosing a machine maximizing the tentative utility at arrival maximizes the job's ex-post utility. Therefore we will call a job *rational* if it truthfully reports w_j, p_j and r_j and chooses a machine maximizing its tentative utility $\hat{u}_j$. In this section, we will show that if all jobs are rational, then the DECENTRALIZED LOCALGREEDY Mechanism is 3.281-competitive.

5.1 Handling Critical Jobs

Recall that from Section 2.1 on, we assumed that no critical jobs exist, i.e. we defined the DECENTRALIZED LOCALGREEDY Mechanism only for jobs j with

$r_j \geq \alpha\, p_j$. We will now relax this assumption and allow jobs to have types from the more general type space $\{(r_j, p_j, w_j) | r_j \geq 0,\, p_j \geq 0,\, w_j \in \mathbb{R}\}$. Without the assumption, the DecentralizedLocalGreedy Mechanism as stated above does not yet yield a constant performance ratio; simple examples can be constructed in the same flavor as in [7]. In fact, it is well known that early arriving jobs with large processing times have to be delayed [5, 7, 8]. In order to achieve a constant performance ratio, we also adopt this idea and use modified release dates as [7, 8]. To this end, we define the modified release date of every job $j \in J$ as $r'_j = \max\{r_j, \alpha p_j\}$, where $\alpha \in (0, 1]$ will later be chosen appropriately. For our decentralized setting, this means that a machine will not admit any job j to its priority queue before time $\max\{\tilde{r}_j, \alpha\tilde{p}_j\}$ if j arrives at time $\tilde{r}_j$ and reports processing time $\tilde{p}_j$. Moreover, machines refuse to provide information about the tentative completion time and payment to a job before its modified release date (with respect to the job's reported data). Note that this modification is part of the local scheduling policy of every machine and therefore does not restrict the required decentralization. Note further that any myopic rational job j still reports $\tilde{w}_j = w_j$ according to Theorem 7 and that a rational job reports $\tilde{p}_j = p_j$ as well as communicates to machines at the earliest opportunity, i.e. at time $\max\{r_j, \alpha p_j\}$, according to the arguments in the proof of Theorem 9. Moreover, the aforementioned properties concerning the balanced budget, the conservation of utility in the case of a truthfully reported weight, and the online property of the payments still apply to the algorithm with modified release dates.

5.2 Proof of the Performance Ratio

It is not a goal in itself to have a truthful mechanism, but to use the truthfulness in order to achieve a reasonable overall performance in terms of the social welfare $\sum w_j\, C_j$. We derive a constant performance ratio for the Decentralized LocalGreedy Mechanism by the following theorem:

Theorem 11. *Suppose every job is rational in the sense that it reports r_j, p_j, w_j and selects a machine that maximizes its tentative utility at arrival. Then the* Decentralized LocalGreedy *Mechanism is ϱ-competitive, with $\varrho = 3.281$.*

The proof of the theorem partly follows the lines of the corresponding proof of Megow et al. [8]. But the distribution of jobs over machines in their algorithm differs from the decentralized distribution in the Decentralized LocalGreedy Mechanism when rational jobs are assumed. Therefore, our result is not implied by the result of Megow et al. [8] and it is necessary to give a proof here.

Proof. A rational job communicates to the machines at time $r'_j = \max\{r_j, \alpha p_j\}$ and chooses a machine i_j that maximizes its utility upon arrival $\hat{u}_j(i_j)$. That

is, it selects a machine i that minimizes

$$-\hat{u}_j(i) = w_j \hat{C}_j(i) + \hat{\pi}_j(i) = w_j \big[r'_j + b_i(r'_j) + \sum_{\substack{k \in H(j) \\ k \to i \\ k < j \\ S_k \geq r'_j}} p_k + p_j \big] + p_j \sum_{\substack{k \in L(j) \\ k \to i \\ k < j \\ S_k > r'_j}} w_k .$$

This, however, exactly equals the immediate increase of the objective value $\sum w_j C_j$ that is due to the addition of job j to the schedule. We now claim that we can express the objective value Z of the resulting schedule as $Z = \sum_{j \in J} -\hat{u}_j(i_j)$, where i_j is the machine selected by job j. Here, it is important to note that $-\hat{u}_j(i_j)$ does not express the total (ex-post) contribution of job j to $\sum w_j C_j$, but only the increase *upon arrival* of j on machine i_j. However, further contributions of job j to $\sum w_j C_j$ only appear when job j is displaced by some later arriving job with higher priority, say k. This contribution by job j to $\sum w_j C_j$, however, will be accounted for when adding $-\hat{u}_k(i_k)$.

Next, since we assume that any job maximizes its utility upon arrival, or equivalently minimizes $-\hat{u}_j(i)$ when selecting a machine i, we can apply an averaging argument over the number of machines, like in [8], to obtain:

$$Z \leq \sum_{i \in J} \frac{1}{m} \sum_{i=1}^{m} -\hat{u}_j(i) \, .$$

The remainder of the proof utilizes the definitions of $\hat{u}_j(i)$ and particulary the fact that, upon arrival of job j on any of the machines i (at time r'_j), machine i is blocked for time $b_i(r'_j)$, which is upper bounded by r'_j/α. This upper bound is machine-independent, and follows from the definition of r'_j, since any job k in process at time r'_j fulfills $\alpha p_k \leq r'_k \leq r'_j$. Furthermore, the proof utilizes a lower bound on any (off-line) optimum schedule from Eastman et al. [14, Thm. 1]. For details, we refer to the full version of the paper. The resulting performance bound 3.281 is identical to the one of [8] (for deterministic processing times), when α is $(\sqrt{17m^2 - 2m + 1} - m + 1)/(4m)$. □

6 Negative Result

Theorem 12. *There does not exist a payment scheme that extends the* LocalGreedy *algorithm to a truthful mechanism. Therefore, it is not possible to turn the* Decentralized LocalGreedy *Mechanism into a mechanism with a dominant strategy equilibrium in which all jobs report truthfully by only modifying the payment scheme.*

Proof. If the Decentralized LocalGreedy Mechanism can be turned into a truthful mechanism by only modifying the payment scheme, then the LocalGreedy algorithm can be completed by a payment scheme to a truthful mechanism. Furthermore, we can show that a necessary condition for truthfulness, called weak monotonicity, is not satisfied by the LocalGreedy algorithm. Weak monotonicity has been introduced in [15]. □

7 Discussion

It would be interesting to find a constant competitive decentralized online scheduling mechanism such that there is a *dominant strategy equilibrium* in which the jobs report all data truthfully. As we have seen in Section 6, the LocalGreedy Algorithm cannot be extended by a payment scheme such that the resulting mechanism has the described properties. Furthermore, recall that the currently best known performance bound for the non-strategic, centralized setting is 2.61 [6]. This algorithm crucially requires a centralized distribution of jobs over machines, and therefore does not seem to be suited for decentralization. Nevertheless, it remains an interesting question to identify general rules for the transformation of centralized algorithms to decentralized mechanisms.

References

[1] Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: A survey. Ann. Discr. Math. **5** (1979), 287–326

[2] Lenstra, J.K., Rinnoy Kan, A.H.G., Brucker, P.: Complexity of machine scheduling problems. Ann. of Discr. Math. **1** (1977), 343–362

[3] Hoogeveen, J.A., Vestjens, A.P.A.: Optimal on-line algorithms for single machine scheduling. In: Cunningham, W.H., McCormick, S.T., Queyranne, M., eds.: IPCO 1996, LNCS 1084 (1996), 404-414

[4] Vestjens, A.P.A.: On-line Machine Scheduling. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (1997)

[5] Anderson, E.J., Potts, C.N.: Online scheduling of a single machine to minimize total weighted completion time. Math. Oper. Res. **29** (2004), 686-697

[6] Correa, J.R., Wagner, M.R.: LP-based online scheduling: from single to parallel machines. In: Jünger, M., Kaibel, V., eds.: IPCO 2005. LNCS 3509 (2005), 196-209

[7] Megow, N., Schulz, A.S.: On-line scheduling to minimize average completion time revisited. Oper. Res. Letters **32** (2004), 485-490

[8] Megow, N., Uetz, M., Vredeveld, T.: Models and algorithms for stochastic online scheduling. Math. Oper. Res., to appear.

[9] Smith, W.: Various optimizers for single stage production. Nav. Res. Log. Quarterly **3** (1956), 59-66

[10] Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior **35** (2001), 166-196

[11] Archer, A., Tardos, E.: Truthful mechanisms for one-parameter agents. In: Proc. 42nd FOCS. IEEE Computer Society (2001), 482-491

[12] Kovacs, A.: Fast monotone 3-approximation algorithm for scheduling related machines. In: Brodal, G.S., Leonardi, S., eds.: ESA 2005. LNCS 3669 (2005), 616-627

[13] Porter, R.: Mechanism design for online real-time scheduling. Proc. 5th ACM Conf. Electronic Commerce, ACM Press (2004), 61-70

[14] Eastman, W.L., Even, S., Isaacs, I.M.: Bounds for the optimal scheduling of n jobs on m processors. Management Science **11** (1964), 268–279

[15] S. Bikhchandani, S. Chatterjee, R. Lavi, A. Mu'alem, N. Nisan, and A. Sen. Weak monotonicity characterizes deterministic dominant strategy implementation,. *Econometrica*, 74(4):1109–1132, 2006.

Birgit Heydenreich and Rudolf Müller and Marc Uetz
Maastricht University,
Quantitative Economics,
P.O.Box 616,
6200 MD Maastricht,
The Netherlands.
Email: {b.heydenreich,r.muller,m.uetz}@ke.unimaas.nl

Guarantees for the Success Frequency of an Algorithm for Finding Dodgson-Election Winners[1]

Christopher M. Homan and Lane A. Hemaspaandra

Abstract

Dodgson's election system elegantly satisfies the Condorcet criterion. However, determining the winner of a Dodgson election is known to be Θ_2^p-complete ([HHR97], see also [BTT89]), which implies that unless P = NP no polynomial-time solution to this problem exists, and unless the polynomial hierarchy collapses to NP the problem is not even in NP. Nonetheless, we prove that when the number of voters is much greater than the number of candidates (although the number of voters may still be polynomial in the number of candidates), a simple greedy algorithm very frequently finds the Dodgson winners in such a way that it "knows" that it has found them, and furthermore the algorithm never incorrectly declares a nonwinner to be a winner.

1 Introduction

The *Condorcet paradox* [Con85], otherwise known as *the paradox of voting* or *the Condorcet effect*, says that rational (i.e., well-ordered) individual preferences can lead to irrational (i.e., cyclical) majority preferences.[2] It is a well-known and widely studied problem in the field of social choice theory [MU95]. A voting system is said to obey the *Condorcet criterion* [Con85] if whenever there is a Condorcet winner—a candidate who in each pairwise subcontest gets a strict majority of the votes—that candidate is selected by the voting system as the overall winner.

The mathematician Charles Dodgson (who wrote fiction under the now more famous name of Lewis Carroll) devised a voting system [Dod76] that has many lovely properties and meets the Condorcet criterion. In Dodgson's system, each voter strictly ranks (i.e., no ties allowed) all candidates in the election. If a Condorcet winner exists, he or she wins the Dodgson election. If no Condorcet winner exists, Dodgson's approach is to take as winners all candidates that are "closest" to being Condorcet winners, with closest being in terms of the fewest

[1]This paper appeared in technical report form as [HH06a] and a shorter version appeared as [HH06b]. Supported in part by NSF grant CCF-0426761, a Friedrich Wilhelm Bessel Research Award, the Alexander von Humboldt Foundation's TransCoop program, and an RIT Faculty Evaluation and Development grant.

[2]For instance, given a choice between a, b, and c, one-third of a group might rank (in order of strictly increasing preference) the candidates (a, b, c), another third might rank them (b, c, a), and the remaining third might rank them (c, a, b). Thus, each voter would have a cycle-free set of preferences, yet 2/3 of the voters would prefer b to a, another 2/3 would prefer a to c, and still another 2/3 would prefer c to b.

changes to the votes needed to make the candidate a Condorcet winner. We will in Section 2 describe what exactly Dodgson means by "fewest changes," but intuitively speaking, it is the smallest number of sequential switches between adjacent entries in the rankings the voters provide. It can thus be seen as a sort of "edit distance" [SK83].

Dodgson wrote about his voting system only in an unpublished pamphlet on the conduct of elections [Dod76] and may never have intended for it to be published. It was eventually discovered and disseminated by Black [Bla58] and is now regarded as a classic of social choice theory [MU95]. Dodgson's system was one of the first to satisfy the Condorcet criterion.[3]

Although Dodgson's system has many nice properties, it also poses a serious computational worry: The problem of checking whether a certain number of changes suffices to make a given candidate the Condorcet winner is NP-complete [BTT89], and the problem of computing an overall winner, as well as the related problem of checking whether a given candidate is at least as close as another given candidate to being a Dodgson winner, is complete for Θ_2^p [HHR97], the class of problems solvable with polynomial-time parallel access to an NP oracle [PZ83]. (More recent work has shown that some other important election systems are complete for Θ_2^p: Hemaspaandra, Spakowski, and Vogel [HSV05] have shown Θ_2^p-completeness for the winner problem in Kemeny elections, and Rothe, Spakowski, and Vogel [RSV03] have shown Θ_2^p-completeness for the winner problem in Young elections.) The above complexity-theoretic results about Dodgson elections show, quite dramatically, that unless the polynomial hierarchy collapses there is no efficient (i.e., polynomial-time) algorithm that is guaranteed to always determine the winners of a Dodgson election. Does this then mean that Dodgson's widely studied and highly regarded voting system is all but unusable?

It turns out that if a small degree of uncertainty is tolerated, then there is a simple, polynomial-time algorithm, `GreedyWinner` (the name's appropriateness will later become clear), that takes as input a Dodgson election and a candidate from the election and outputs an element in {"yes", "no"} × {"definitely", "maybe"}. The first component of the output is the algorithm's guess as to whether the input candidate was a winner of the input election. The second output component indicates the algorithm's confidence in its guess. Regarding the accuracy of `GreedyWinner` we have the following results.

Theorem 1.1. *1. For each (election, candidate) pair it holds that if* `GreedyWinner` *outputs "definitely" as its second output component, then its first output component correctly answers the question, "Is the input candidate a Dodgson winner of the input election?"*

[3]The Condorcet criterion may at first glance seem easy to satisfy, but Nanson showed [Nan82] that many well-known voting systems—such as the rank-order system [Bor84] widely attributed to Borda (which Condorcet himself studied [Con85] in the same paper in which he introduced the Condorcet criterion), in which voters assign values to each candidate and the one receiving the largest (or smallest) aggregate value wins—fail to satisfy the Condorcet criterion.

2. *For each $m, n \in \mathbb{N}^+$, the probability that a Dodgson election E selected uniformly at random from all Dodgson elections having m candidates and n votes (i.e., all $(m!)^n$ Dodgson elections having m candidates and n votes have the same likelihood of being selected*[4]*) has the property that there exists at least one candidate c such that* `GreedyWinner` *on input (E, c) outputs "maybe" as its second output component is less than $2(m^2 - m)e^{\frac{-n}{8m^2}}$.*

Thus, for elections where the number of voters greatly exceeds the number of candidates (though the former could still be within a (superquadratic) polynomial of the latter, consistently with the success probability for a family of election draws thus-related in voter-candidate cardinality going asymptotically to 1), if one randomly chooses an election $E = (C, V)$, then with high likelihood it will hold that for each $c \in C$ the efficient algorithm `GreedyWinner` when run on input (C, V, c) correctly determines whether c is a Dodgson winner of E, and moreover will "know" that it got those answers right. We call `GreedyWinner` a *frequently self-knowingly correct*[5] heuristic. (Though the `GreedyWinner` algorithm on its surface is about *recognizing* Dodgson winners, as discussed in Section 3 our algorithm can be easily modified into one that is about, given an $E = (C, V)$, *finding* the complete set of Dodgson Winners and that does so in a way that is, in essentially the same high frequency as for `GreedyWinner`, self-knowingly correct.) Later in this paper, we will introduce another frequently self-knowingly correct heuristic, called `GreedyScore`, for calculating the Dodgson score of a given candidate.

2 Dodgson Elections

As mentioned in the introduction, in Dodgson's voting system each voter strictly ranks the candidates in order of preference. Formally speaking, for $m, n \in \mathbb{N}^+$ (throughout this paper we by definition do not admit as valid elections with zero candidates or zero voters), a *Dodgson election* is an ordered pair (C, V) where C is a set $\{c_1, \ldots, c_m\}$ of candidates (as noted earlier, we without loss of generality view them as being named by 1, 2, $\ldots$, m) and V is a tuple $(v_1, \ldots, v_n)$ of *votes* and a *Dodgson triple*, denoted (C, V, c), is a Dodgson election (C, V) together with a candidate $c \in C$. Each vote is one of the $m!$ total orderings over the candidates, i.e., it is a complete, transitive, and antireflexive relation over the set of candidates. We will sometimes denote a vote by listing the candidates in

[4]Since Dodgson voting is not sensitive to the *names* of candidates, we will throughout this paper always tacitly assume that all m-candidate elections have the fixed candidate set $1, 2, \ldots, m$ (though in some examples we for clarity will use other names, such as a, b, c, and d). So, though we to be consistent with earlier papers on Dodgson elections allow the candidate set "C" to be part of the input, in fact we view this as being instantly coerced into the candidate set $1, 2, \ldots, m$. And we similarly view voter *names* as uninteresting.

[5]The full version of this paper [HH06a] contains a long discussion of how self-knowing correctness differs from other sorts of algorithmic analysis such as smoothed analysis and average-case complexity, but for space reasons we cannot include that here.

increasing order, e.g., (x, y, z) is a vote over the candidate set $\{x, y, z\}$ in which y is preferred to x and z is preferred to (x and) y. (Note: A candidate is never preferred to him- or herself.) For vote v and candidates $c, d \in C$, "$c <_v d$" means "in vote v, d is preferred to c" and "$c \prec_v d$" means "$c <_v d$ and there is no e such that $c <_v e <_v d$." Each Dodgson election gives rise to $\binom{m}{2}$ *pairwise races*, each of which is created by choosing two distinct candidates $c, d \in C$ and restricting each vote v to the two chosen candidates, that is, to either (c, d) or (d, c). The winner of the pairwise race is the one that a strict majority of voters prefer. Due to ties, a winner may not always exist in pairwise races.

A *Condorcet winner* is any candidate c that, against each remaining candidate, is preferred by a strict majority of voters. For a given election (i.e., for a given sequence of votes), it is possible that no Condorcet winner exists. However, when one does exist, it is unique.

For any vote v and any $c, d \in C$, if $c \prec_v d$, let $Swap_{c,d}(v)$ denote the vote v', where v' is the same total ordering of C as v except that $d <_{v'} c$ (note that this implies $d \prec_{v'} c$). If $c \not\prec_v d$ then $Swap_{c,d}(v)$ is undefined. In effect, a swap causes c and d to "switch places," but only if c and d are adjacent. The *Dodgson score* of a Dodgson triple (C, V, c) is the minimum number of swaps that, applied sequentially to the votes in V, make V a sequence of votes in which c is the Condorcet winner. A *Dodgson winner* is a candidate that has the smallest Dodgson score. This is the election system developed in the year 1876 by Dodgson (Lewis Carroll) [Dod76], and as noted earlier it gives victory to the candidate(s) who are "closest" to being Condorcet winners. Note that if no candidate is a Condorcet winner, then two or more candidates may tie, in which case all tying candidates are Dodgson winners.

Decision Problem: `DodgsonScore`
Instance: A Dodgson triple (C, V, c); a positive integer k.
Question Is $Score(C, V, c)$, the Dodgson score of candidate c in the election specified by (C, V), less than or equal to k?

Decision Problem: `DodgsonWinner`
Input: A Dodgson triple (C, V, c).
Question: Is c a winner of the election? That is, does c tie-or-defeat all other candidates in the election?

Bartholdi, Tovey, and Trick show that the problem of checking whether a certain number of changes suffices to make a given candidate the Condorcet winner is NP-complete and that the problem of determining whether a given candidate is a Dodgson winner is NP-hard [BTT89]. Hemaspaandra, Hemaspaandra, and Rothe show [HHR97] that this latter problem, as well as the related problem of checking whether a given candidate is at least as close as another given candidate to being a Dodgson winner, is complete for Θ_2^p. Hemaspaandra, Hemaspaandra, and Rothe's results show that determining a Dodgson winner is not even in NP unless the polynomial hierarchy collapses. This line of work has significance because the hundred-year-old problem of deciding if a given can-

didate is a Dodgson winner was more naturally conceived than the problems that were previously known to be complete for Θ_2^p (see [Wag87]).

3 The `GreedyScore` and `GreedyWinner` Algorithms

In this section, we study the greedy algorithms `GreedyScore` and `GreedyWinner`, stated as, respectively, Algorithm 1 (page 6) and Algorithm 2 (page 7), and we note that their running time is polynomial. We show that both algorithms are self-knowingly correct in the sense of the following definition.

Definition 3.1. *For sets S and T and function $f : S \to T$, an algorithm $\mathcal{A} : S \to T \times \{$"definitely", "maybe"$\}$ is* self-knowingly correct for f *if, for all $s \in S$ and $t \in T$, whenever $\mathcal{A}$ on input s outputs $(t,$ "definitely"$)$ it holds that $f(s) = t$.*

The reader may wonder whether "self-knowing correctness" is so easily added to heuristic schemes as to be uninteresting to study. After all, if one has a heuristic for finding certificates for an NP problem with respect to some fixed certificate scheme (in the standard sense of NP certificate schemes)—e.g., for trying to find a satisfying assignment to an input (unquantified) propositional boolean formula—then one can use the P-time checker associated with the problem to "filter" the answers one finds, and can put the label "definitely" on only those outputs that are indeed certificates. However, the problem studied in this paper does not seem amenable to such after-the-fact addition of self-knowingness, as in this paper we are dealing with heuristics that are seeking objects that are computationally much more complex than mere certificates related to NP problems. In particular, a polynomial-time function-computing machine seeking to compute an input's Dodgson score seems to require about logarithmically many adaptive calls to SAT.[6]

We call `GreedyScore` "greedy" because, as it sweeps through the votes, each swap it (virtually) does immediately improves the standing of the input candidate against some adversary that the input candidate is at that point losing to. The algorithm nonetheless is very simple. It limits itself to at most one swap per vote. Yet, its simplicity notwithstanding, we will eventually prove that this (self-knowingly correct) algorithm is very frequently correct.

[6] We say "seems to," but we note that one can make a more rigorous claim here. As mentioned in Section 2, among the problems that Hemaspaandra, Hemaspaandra, and Rothe [HHR97] prove complete for the language class Θ_2^p is `DodgsonWinner`. If one could, for example, compute Dodgson scores via a polynomial-time function-computing machine that made a (globally) constant-bounded number of queries to SAT, then this would prove that `DodgsonWinner` is in the boolean hierarchy [CGH$^+$88], and thus that Θ_2^p equals the boolean hierarchy, which in turn would imply the collapse of the polynomial hierarchy [Kad88]. That is, this function problem is so closely connected to a Θ_2^p-complete language problem that if one can save queries in the former, then one immediately has consequences for the complexity of the latter.

Algorithm 1: `GreedyScore`(C, V, c) [n = number of voters; m = number of candidates]

```
1: for all d ∈ C−{c} do
2:    Deficit[d] ← 1 − ⌈n/2⌉
3:    Swaps[d] ← 0
4: end for
5: for all votes v[] in V do
6:    state ← "nocount"
7:    for all i ∈ (1, ..., m) do
8:       if (state = "incrdef") ∨ (state = "swap") then
9:          Deficit[v[i]] ← Deficit[v[i]] + 1
10:         if state = "swap" then
11:            Swaps[v[i]] ← Swaps[v[i]] + 1
12:            state ← "incrdef"
13:         end if
14:      else if c = v[i] then
15:         state ← "swap"
16:      end if
17:   end for
18: end for
19: confidence ← "definitely"
20: score ← 0
21: for all d ∈ C−{c} do
22:    if Deficit[d] > 0 then
23:       score ← score + Deficit[d]
24:       if Deficit[d] > Swaps[d] then
25:          confidence ← "maybe"
26:          score ← score + 1
27:       end if
28:    end if
29: end for
30: return (score, confidence)
```

We now state the main result for this section, and a bit later we will briefly describe the algorithms in English.

Theorem 3.2. *1. `GreedyScore` (Algorithm 1) is self-knowingly correct for Score (recall that Score is defined in Section 2 in the statement of the `DodgsonScore` problem).*

2. `GreedyWinner` (Algorithm 2) is self-knowingly correct for `DodgsonWinner`*.*

3. `GreedyScore` and `GreedyWinner` both run in polynomial time.[7]

Note that Theorem 1.1.1 follows directly from Theorem 3.2.2. We will prove Theorem 1.1.2 in Section 4.

Theorem 3.2, since it just states polynomial time, is not heavily dependent on the encoding scheme used. However, we will for specificity give a specific scheme that can be used. Note that the scheme we use will encode the inputs as binary strings by a scheme that is easy to compute and invert and encodes each vote as an $\mathcal{O}(\|C\| \log \|C\|)$-bit substring and each Dodgson triple as an $\mathcal{O}(\|V\| \cdot \|C\| \cdot \log \|C\|)$-bit string, where (C, V, c) is the input to the encoding scheme. For a Dodgson triple (C, V, c), our encoding scheme is as follows.

[7]The number of times lines of Algorithm 1 (respectively, Algorithm 2) are executed is clearly $\mathcal{O}(\|V\| \cdot \|C\|)$ (respectively, $\mathcal{O}(\|V\| \cdot \|C\|^2)$), and so these are indeed polynomial-time algorithms.

For completeness, we mention that when one takes into account the size of the objects being manipulated (in particular, under the assumption—which in light of the encoding scheme we will use below is not unreasonable—that it takes $\mathcal{O}(\log \|C\|)$ time to look up a key in either *Deficit* or *Votes* and $\mathcal{O}(\log \|V\|)$ time to update the associated value, and each *Swap* operation takes $\mathcal{O}(\log \|C\|)$ time) the running time of the algorithm might be more fairly viewed as $\mathcal{O}(\|V\| \cdot \|C\| \cdot (\log \|C\| + \log \|V\|))$ (respectively, $\mathcal{O}(\|V\| \cdot \|C\|^2 \cdot (\log \|C\| + \log \|V\|))$), though in any case it certainly is a polynomial-time algorithm.

- First comes $\|C\|$, encoded as a binary string of length $\lceil \log(\|C\|+1) \rceil$,[8] preceded by the substring $1^{\lceil \log(\|C\|+1) \rceil}0$.

- Next comes the chosen candidate c, encoded as a binary string of length $\lceil \log(\|C\|+1) \rceil$.

- Finally each vote is encoded as a binary substring of length $\|C\| \cdot \lceil \log(\|C\|+1) \rceil$.

Regarding the notation used in Algorithm 1: A vote is represented as an array $v[]$ of length m, where $m = \|C\|$. For each vote $v[]$, $v[1]$ is the least preferred candidate, $v[2]$ is the second least preferred candidate, and so on, and $v[m]$ is the most preferred candidate. $Swap_i(v)$ means that the ith and $(i+1)$st values in $v[]$ are swapped.

Algorithm 2: `GreedyWinner`(C, V, c)

```
 1: (cscore, confidence) = GreedyScore(C, V, c)
 2: winner ← "yes"
 3: for all candidates d ∈ C−{c} do
 4:    (dscore, dcon) ← GreedyScore(C, V, d)
 5:    if dscore < cscore then
 6:       winner ← "no"
 7:    end if
 8:    if dcon = "maybe" then
 9:       confidence ← "maybe"
10:    end if
11: end for
12: return (winner, confidence)
```

We now describe in English what our algorithms actually do (however, all references above and below to specific variables such as $v[]$, $Swap[]$, and $Deficit[]$, refer to their included pseudocode versions). Briefly put, `GreedyScore`, for each candidate d, $c \neq d \in C$, computes (in $Deficit[d]$) the number of votes (if any) that c needs to gain in order to have strictly more votes than d (in a pairwise contest between them), and computes (in $Swaps[d]$) the number of votes v in which d is immediately adjacent to and preferred to c ($c \prec_v d$). If the former number is strictly greater than zero and the latter number is at least as large as the former number, then it is the case that by adjacent swaps in exactly the former number of votes—when done in that number of votes chosen from among those votes v satisfying $c \prec_v d$—c can be with perfect efficiency (every swap pays off by reducing a positive shortfall) be changed to beating d. If the number values just stated are not the case, the `GreedyScore` algorithm declares that it is stumped by the current input. If it is stumped for no candidate d, $c \neq d \in C$, then it simply adds up the costs of defeating each other candidate, and is secure in the knowledge that this is optimal (see also the proof below).

Turning to the `GreedyWinner` algorithm, it does the above for all candidates, and if while doing so `GreedyScore` is never stumped, then `GreedyWinner` uses in

[8] All logarithms in this paper are base 2. We use $\lceil \log(\|C\|+1) \rceil$-bit strings rather than $\lceil \log(\|C\|) \rceil$-bit strings as we wish to have the size of the coding scale at least linearly with the number of voters even in the pathological $\|C\| = 1$ case (in which each vote carries no information other than about the number of voters).

the obvious way the information it has obtained, and (correctly) states whether c is a Dodgson winner of the input election.

Proof of Theorem 3.2. For item 1, suppose that `GreedyScore`, on input (C, V, c), returns "definitely" as the second component of its output. Then, at the point in time when the algorithm completes, it must hold that, for each $d \in C - \{c\}$, $\mathit{Swaps}[d] \geq \mathit{Deficit}[d]$. Note that for each $d \in C - \{c\}$, $\mathit{Deficit}[d]$ is initially set to $1 - \lceil \|V\|/2 \rceil$ and then is incremented once for every vote v in which d is preferred to c. As noted above, it follows that $\mathit{Deficit}[d]$, if it after that process is nonnegative,

will be set to the minimum number of votes v where the relationship $c <_v d$ needs to be reversed in order for c to beat d. Also as noted above, $\mathit{Swap}[d]$ will by the time all votes are visited be set to the number of votes v such that $c \prec_v d$. Thus, since $\mathit{Swaps}[d] \geq \mathit{Deficit}[d]$ it is possible (i.e., by swapping c and d in $\mathit{Deficit}[d]$ of the votes that $\mathit{Swaps}[d]$ counts) to turn (C, V) into an election in which c beats d by performing only $\mathit{Deficit}[d]$ swaps involving d (which clearly is the fewest swaps that can result in c beating d) when $\mathit{Deficit}[d] > 0$, and by performing zero swaps involving d when $\mathit{Deficit}[d] \leq 0$. From this, and because for each $d, e \in C - \{c\}$ such that $d \neq e \wedge \mathit{Swaps}[d] \geq \mathit{Deficit}[d] > 0 \wedge \mathit{Swaps}[e] \geq \mathit{Deficit}[e] > 0$ it holds that $\{v \mid v \text{ is vote in } C \text{ and } c \prec_v e\} \cap \{v \mid v \text{ is vote in } C \text{ and } c \prec_v d\} = \emptyset$, one can by making $\mathit{Deficit}[d] + \mathit{Deficit}[e]$ swaps turn (C, V) into an election in which c beats both d and e. Similarly, one can by making $\Sigma_{d \in C - \{c\} : \mathit{Deficit}[d] > 0} \mathit{Deficit}[d]$ swaps turn (C, V) into an election in which c beats every $d \in C - \{c\}$. Because one swap reverses the preference relationship between exactly one pair of candidates in exactly one vote, $\Sigma_{d \in C - \{c\} : \mathit{Deficit}[d] > 0} \mathit{Deficit}[d]$ is the Dodgson score of c, which is the first component of the output of `GreedyScore` whenever the second component is "definitely."

For item 2, clearly `GreedyWinner` correctly checks whether c is a Dodgson winner if every call it makes to `GreedyScore` correctly calculates the Dodgson score. `GreedyWinner` then returns "definitely" exactly if each call it makes to `GreedyScore` returns "definitely." But, by item 1, `GreedyScore` is self-knowingly correct.

Item 3 follows from a straightforward analysis of the algorithm (see also footnote 7). □

Note that `GreedyWinner` could easily be modified into a new polynomial-time algorithm that, rather than checking whether a given candidate is the winner of the given Dodgson election, finds all Dodgson winners by taking as input a Dodgson election alone (rather than a Dodgson triple) and outputting a list of *all* the Dodgson winners in the election. This modified algorithm on any Dodgson election (C, V) would make exactly the same calls to `GreedyScore` that the current `GreedyWinner` (on input (C, V, c), where $c \in C$) algorithm makes, and the new algorithm would be accurate whenever every call it makes to `GreedyScore` returns "definitely" as its second argument. Thus, whenever the current `GreedyWinner` would return a "definitely" answer so would the new

Dodgson-winner-finding algorithm (when their inputs are related in the same manner as described above). These comments explain why in the title (and abstract), we were correct in speaking of "*finding* Dodgson-Election Winners" (rather than merely recognizing them).

4 Analysis of the Correctness Frequency of the Two Heuristic Algorithms

In this section, we prove that, as long as the number of votes is much greater than the number of candidates, `GreedyWinner` is a frequently self-knowingly correct algorithm.

Theorem 4.1. *For each $m, n \in \mathbb{N}^+$, the following hold. Let $C = \{1, \ldots, m\}$.*

1. *Let V satisfy $\|V\| = n$. For each $c \in C$, if for all $d \in C - \{c\}$ it holds that $\|\{i \in \{1, \ldots, n\} \mid c <_{v_i} d\}\| \leq \frac{2mn+n}{4m}$ and $\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| \geq \frac{3n}{4m}$ then $\texttt{GreedyScore}(C, V, c) = (Score(C, V, c)$, "definitely").*

2. *For each $c, d \in C$ such that $c \neq d$, $\Pr((\|\{i \in \{1, \ldots, n\} \mid c <_{v_i} d\}\| > \frac{2mn+n}{4m}) \vee (\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| < \frac{3n}{4m})) < 2e^{\frac{-n}{8m^2}}$, where the probability is taken over drawing uniformly at random an m-candidate, n-voter Dodgson election $V = (v_1, \ldots, v_n)$ (i.e., all $(m!)^n$ Dodgson elections having m candidates and n voters have the same likelihood of being chosen).*

3. *For each $c \in C$, $\Pr(\texttt{GreedyScore}(C, V, c) \neq (Score(C, V, c)$, "definitely")$) < 2(m-1)e^{\frac{-n}{8m^2}}$, where the probability is taken over drawing uniformly at random an m-candidate, n-voter Dodgson election $V = (v_1, \ldots, v_n)$.*

4. *$\Pr((\exists c \in C)[\texttt{GreedyWinner}(C, V, c) \neq (\texttt{DodgsonWinner}(C, V, c)$, "definitely")$])$ $< 2(m^2 - m)e^{\frac{-n}{8m^2}}$, where the probability is taken over drawing uniformly at random an m-candidate, n-voter Dodgson election $V = (v_1, \ldots, v_n)$.*

Note that Theorem 1.1.2 follows from Theorem 4.1.4.

The main intuition behind Theorem 4.1 is that, in any election having m candidates and n voters, and for any two candidates c and d, it holds that, in exactly half of the ways v a voter can vote, $c <_v d$, but for exactly $1/m$ of the ways, $c \prec_v d$. Thus, assuming that the votes are chosen independently of each other, when the number of voters is large compared to the number of candidates, with high likelihood the number of votes v for which $c <_v d$ will hover around $n/2$ and the number of votes for which $c \prec_v d$ will hover around n/m. This means that there will (most likely) be enough votes available for our greedy algorithms to succeed.

Throughout this section, regard $V = (v_1, \ldots, v_n)$ as a sequence of n independent observations of a random variable γ whose distribution is uniform over

the set of all votes over a set $C = \{1, 2, \ldots, m\}$ of m candidates, where γ can take, with equal likelihood, any of the $m!$ distinct total orderings over C. (This distribution should be contrasted with such work as that of, e.g., [RM05], which in a quite different context creates dependencies between voters' preferences.)

Proof of Theorem 4.1. For item 1, $\frac{2mn+n}{4m} = \frac{n}{2} + \frac{n}{4m}$, so, if $\|\{i \in \{1, \ldots, n\} \mid c <_{v_i} d\}\| \leq \frac{2mn+n}{4m}$ then either c already beats d or if not then the defection of more than $\frac{n}{4m}$ votes from preferring-d-to-c to preferring-c-to-d would (if such votes exist) ensure that c beats d. If $\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| \geq \frac{3n}{4m}$ then (keeping in mind that we have globally excluded as invalid all cases where at least one of n or m equals zero) $\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| > \frac{n}{4m}$, and so `GreedyScore` will be able to make enough swaps (in fact, and this is critically important in light of Algorithm 1, there is a sequence of swaps such that any vote has at most one swap operation performed on it) so that c beats d. Item 2 follows from applying the union bound (which of course does not require independence) to Lemma 4.3, which is stated and proven below. Item 3 follows from item 1 and from applying item 2 and the union bound to $\Pr(\bigvee_{d \in C - \{c\}}((\|\{i \in \{1, \ldots, n\} \mid c <_{v_i} d\}\| > \frac{2mn+n}{4m}) \vee (\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| < \frac{3n}{4m})))$. Item 4 follows from item 1 and from applying item 2 and the union bound to $\Pr(\bigvee_{c,d \in C \wedge c \neq d}((\|\{i \in \{1, \ldots, n\} \mid c <_{v_i} d\}\| > \frac{2mn+n}{4m}) \vee (\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| < \frac{3n}{4m})))$ (note that $\|\{(c, d) \mid c \in C \wedge d \in C \wedge c \neq d\}\| = m^2 - m$). □

We now turn to stating and proving Lemma 4.3, which is needed to support the proof of Theorem 4.1. Lemma 4.3 follows from the following variant of Chernoff's Theorem [Che52].

Theorem 4.2 ([AS00]). *Let $X_1, \ldots, X_n$ be a sequence of mutually independent random variables. If there exists a $p \in [0, 1] \subseteq \mathbb{R}$ such that, for each $i \in \{1, \ldots, n\}$, ($\Pr(X_i = 1 - p) = p$ and $\Pr(X_i = -p) = 1 - p$), then for all $a \in \mathbb{R}$ where $a > 0$ it holds that $\Pr(\Sigma_{i=1}^{n} X_i > a) < e^{-2a^2/n}$.*

Lemma 4.3. *1.* $\Pr(\|\{i \in \{1, \ldots, n\} \mid c <_{v_i} d\}\| > \frac{2mn+n}{4m}) < e^{\frac{-n}{8m^2}}$.

2. $\Pr(\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| < \frac{3n}{4m}) < e^{\frac{-n}{8m^2}}$.

Proof. 1. For each $i \in \{1, \ldots, n\}$, define X_i as $X_i = \begin{cases} 1/2 & \text{if } c <_{v_i} d, \\ -1/2 & \text{otherwise.} \end{cases}$ Then $\|\{i \in \{1, \ldots, n\} \mid c <_{v_i} d\}\| > \frac{2mn+n}{4m}$ exactly if $\sum_{i=1}^{n} X_i > \frac{1}{2}\left(\frac{2mn+n}{4m}\right) - \frac{1}{2}\left(n - \frac{2mn+n}{4m}\right)$. Since $\frac{1}{2}\left(\frac{2mn+n}{4m}\right) - \frac{1}{2}\left(n - \frac{2mn+n}{4m}\right) = \frac{n}{4m}$, setting $a = \frac{n}{4m}$ and $p = \frac{1}{2}$ in Theorem 4.2 yields the desired result.

2. For each $i \in \{1, \ldots, n\}$, define X_i as $X_i = \begin{cases} 1/m & \text{if } c \not\prec_{v_i} d, \\ 1/m - 1 & \text{otherwise.} \end{cases}$ Then $\|\{i \in \{1, \ldots, n\} \mid c \prec_{v_i} d\}\| < \frac{3n}{4m}$ if and only if $\|\{i \in \{1, \ldots, n\} \mid c \not\prec_{v_i} d\}\| > n - \frac{3n}{4m}$ if and only if $\sum_{i=1}^{n} X_i > \frac{1}{m}\left(n - \frac{3n}{4m}\right) + \left(\frac{1}{m} - 1\right)\frac{3n}{4m}$. Since $\frac{1}{m}\left(n - \frac{3n}{4m}\right) + \left(\frac{1}{m} - 1\right)\frac{3n}{4m} = \frac{n}{4m}$, setting $a = \frac{n}{4m}$ and $p = 1 - \frac{1}{m}$ in Theorem 4.2 yields the desired result. □

We now have proven Theorem 1.1.

Proof of Theorem 1.1. As mentioned in Section 3, Theorem 1.1.1 follows from Theorem 3.2.2. Theorem 1.1.2 follows from Theorem 4.1.4. □

5 Conclusion and Open Directions

The Dodgson voting system elegantly satisfies the Condorcet criterion. Although it is NP-hard (and so if P $\neq$ NP is computationally infeasible) to determine the winner of a Dodgson election or compute scores for Dodgson elections, we provided heuristics, `GreedyWinner` and `GreedyScore`, for computing winners and scores for Dodgson elections. We showed that these heuristics are computationally simple, and we showed that, over all elections of a given size where the number of voters is much greater than the number of candidates (although the number of voters may still be polynomial in the number of candidates) in a randomly chosen election, these algorithms, with likelihood approaching one, get the right answer and know that they are correct.

We consider the fact that one can prove this even for such simple greedy algorithms to be an *advantage*—it is good that one does not have to resort to involved algorithms to guarantee extremely frequent success. Nonetheless, it is also natural to wonder to what degree these heuristics can be improved. What would be the effect of adding, for instance, limited backtracking or random nongreedy swaps to our heuristics? Regarding our analysis, in the distributions we consider, each vote is cast independently of every other. What about distributions in which there are dependencies between voters?

It is also natural to wonder whether one can state a general, abstract model of what it means to be frequently self-knowingly correct. That would be a large project (that we heartily commend as an open direction), and here we merely make a brief definitional suggestion for a very abstract case—in some sense simpler to formalize than Dodgson elections, as Dodgson elections have both a voter-set size and a candidate-set size as parameters, and have a domain that is not Σ^* but rather is the space of valid Dodgson triples—namely the case of function problems where the function is total and the simple parameter of input-length is considered the natural way to view and slice the problem regarding its asymptotics. Such a model is often appropriate in computer science (e.g., a trivial such problem—leaving tacit the issues of encoding integers as bit-strings—is $f(n) = 2n$, and harder such problems are $f(n)$ equals the number of primes less than or equal to n and $f(0^i) = \|\mathrm{SAT} \cap \Sigma^i\|$).

Definition 5.1. *Let A be a self-knowingly correct algorithm for $g : \Sigma^* \to T$.*

1. *We say that A is* frequently self-knowingly correct *for g if* $\lim_{n\to\infty} \frac{\|\{x \in \Sigma^n \mid A(x) \in T \times \{\text{“}maybe\text{”}\}\}\|}{\|\Sigma^n\|} = 0$.

2. *Let h be some polynomial-time computable mapping from $\mathbb{N}$ to the rationals. We say that A is* h-frequently self-knowingly correct for g *if* $\frac{\|\{x \in \Sigma^n \mid A(x) \in T \times \{\text{“}maybe\text{”}\}\}\|}{\|\Sigma^n\|} = O(h(n))$.

Since the probabilities that the above definition is tracking may be quite encoding dependent, the second part of the above definition allows us to set more severe demands regarding how often the heuristic (which, being self-knowingly correct, always has the right output when its second component is “definitely”) is allowed to remain uncommitted.

Acknowledgments In an undergraduate project in one of our courses, G. Goldstein, D. Berlin, K. Osipov, and N. Rutar proposed a (far more complex) greedy algorithm for Dodgson elections and experimentally observed that it was often successful. The present paper was motivated by their exciting experimental insight, and seeks to prove rigorously that a greedy approach can be frequently successful on Dodgson elections. We thank the anonymous MFCS06 and COMSOC06 referees for helpful comments.

References

[AS00] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley–Interscience, second edition, 2000.

[Bla58] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[Bor84] J. C. de Borda. Mémoire sur les élections au scrutin. *Histoire de L'Académie Royale des Sciences Année 1781*, 1784.

[BTT89] 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.

[CGH+88] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, 1988.

[Che52] H. Chernoff. A measure of the asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[Con85] M. 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.

[Dod76] C. Dodgson. A method of taking votes on more than two issues. Clarendon Press, Oxford, pamphlet, 1876.

[HH06a] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. Technical Report TR-2005-881, University of Rochester Department of Computer Science, June 2006.

[HH06b] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. In *Proceedings of the 31st International Conference on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science #4162, pages 528–539. Springer-Verlag, August/September 2006.

[HHR97] 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.

[HSV05] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, 1988. Erratum appears in the same journal, 20(2):404.

[MU95] I. McLean and A. Urken. *Classics of Social Choice*. University of Michigan Press, Ann Arbor, Michigan, 1995.

[Nan82] E. Nanson. Methods of election. *Transactions and Proceedings of the Royal Society of Victoria*, 19:197–240, 1882.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings of the 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.

[RM05] G. Raffaelli and M. Marsili. Statistical mechanics model for the emergence of consensus. *Physical Review E*, 72(1):016114, 2005.

[RSV03] 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.

[SK83] D. Sankoff and J. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Computation*. Addison-Wesley, 1983.

[Wag87] K. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1–2):53–80, 1987.

Christopher M. Homan
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623 USA
URL: `http://www.cs.rit.edu/~cmh`

Lane A. Hemaspaandra
Department of Computer Science
University of Rochester
Rochester, NY 14627 USA
URL: `http://www.cs.rochester.edu/u/lane`

Approval Voting: Local Search Heuristics and Approximation Algorithms for the Minimax Solution

Rob LeGrand Evangelos Markakis Aranyak Mehta

Abstract

Voting has been the most general scheme for preference aggregation in multi-agent settings involving agents of diverse preferences. Here, we study a specific type of voting protocols for multi-winner elections, namely *approval voting*, and we investigate the complexity of computing or approximating the *minimax solution* in approval voting, concentrating on elections for committees of fixed size. Given an approval voting election, where voters can vote for as many candidates as they like, a minimax outcome is a committee that minimizes the maximum Hamming distance to the voters' ballots. We first show that the problem is **NP**-hard and give a simple 3-approximation algorithm. We then introduce and evaluate various heuristics based on local search. Our heuristics have low running times (and provably polynomial) and our experimental results show that they perform very well on average, computing solutions that are very close to the optimal minimax solutions. Finally, we address the issue of manipulating minimax outcomes. We show that even though exact algorithms for the minimax solution are manipulable, we can have approximation algorithms that are non-manipulable.

1 Introduction

Voting has been a very popular method for preference aggregation in multi-agent environments. It is often the case that a set of agents with different preferences need to make a choice among a set of alternatives, where the alternatives could be various entities such as potential committee members, or joint plans of action. A standard methodology for this scenario is to have each agent express his preferences and then select an alternative (or more than one alternative in multi-winner elections) according to some voting protocol. Several decision making applications in AI have followed this approach including problems in collaborative filtering [19] and planning [9, 10].

In this work we focus on solution concepts for approval voting, which is a voting scheme for committee elections (multi-winner elections). 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, including the American Mathematical Society (AMS), the Institute of Electrical and Electronics Engineers (IEEE), the Game Theory Society (GTS) and the European Association for Logic, Language and Information.

A ballot in an approval voting protocol can be seen as a binary vector that indicates the candidates approved of by the voter. Given the ballots, the obvious question is: what should the outcome of the election be? The solution concept that has been used in almost all such elections 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. If there is no restriction on the size of the elected committee this is equivalent to a majority vote on each candidate. If there is a restriction, *e.g.*, if the elected committee should be of size exactly k, then the minisum solution consists of the k candidates with the highest number of approvals [4].

Recently, a new solution concept, the minimax solution, was proposed by Brams, Kilgour and Sanver [3]. The minimax solution chooses a committee which, when seen as a 0/1-vector, minimizes the *maximum* Hamming distance to all ballots. When there is a restriction that the size of the committee should be exactly k, then the minimax solution picks, among all committees of size k, the one that minimizes the maximum Hamming distance to the ballots.

The main motivation behind the minimax solution is to address the issues of fairness and compromise. Since minimax minimizes the disagreement with the least satisfied voter, it tends to result in outcomes that are more widely acceptable than the minisum solution. Also, majority tyranny is avoided: a majority of voters cannot guarantee a specific outcome, unlike under minisum. On the other hand, advantages of the minisum approach include simplicity, ease of computation and nonmanipulability. A further discussion on the properties and the pros and cons of the minisum and the minimax solutions can be found in [3, 4].

In this work we address computational aspects of the minimax solution, with a focus on elections for committees of fixed size. In contrast to the minisum solution, which is easy to compute in polynomial time, we show that finding a minimax solution is **NP**-hard. We therefore resort to polynomial-time heuristics and approximation algorithms.

We first exhibit a simple algorithm that achieves an approximation factor of 3. We then propose a variety of local search heuristics, some of which use the solution of our approximation algorithm as an initial point. All our heuristics run relatively fast and we evaluated the quality of their output both on randomly generated data as well as on the 2003 Game Theory Society election. Our simulations show that the heuristics perform very well, finding a solution very close to optimal on average. In fact for some heuristics the average error in the approximation can be as low as 0.05%.

Finally, in Section 5, we focus on the question of manipulating the minimax solution. We show that any algorithm that computes an optimal minimax solution is manipulable. However, the same may not be true for approximation algorithms. As an example, we show that our 3-approximation algorithm is nonmanipulable.

1.1 Related Work

The minimax solution concept that we study here was introduced by Brams, Kilgour and Sanver [3]. In subsequent work by the same authors [4, 14], a weighted version of the minimax solution is studied, which takes into account the number of voters who voted for each distinct ballot and the proximity of each ballot to the other voters' ballots. The algorithms that are proposed in [3, 4, 14] are all exponential, and this is not surprising since the problem is **NP**-hard, as we exhibit in Section 3. Approximation algorithms have previously been established only for the version in which there is no restriction on the size of the committee (which includes as a possibility that no candidate is elected). This variant is referred to as the endogenous minimax solution and it also arises in coding theory under the name of the Minimum Radius Problem or the Hamming Center Problem and in computational biology, where it is known as the Closest String Problem. In the context of computational biology, it was shown by Li, Ma and Wang [17] that the endogenous version admits a Polynomial Time Approximation Scheme (PTAS), *i.e.*, a $(1+\epsilon)$-approximation for any constant ϵ (with the running time depending exponentially in $1/\epsilon$). Other constant-factor approximations for the endogenous version had been obtained before [12, 15]. We are not aware of any polynomial-time approximation algorithms or any heuristic approaches for the non-endogenous versions, *i.e.*, in the presence of any upper or lower bounds on the size of the committee. Complexity considerations for winner determination in multi-winner elections have also been addressed recently [21] but not for the minimax solution.

2 Definitions and Notation

We now formally define our problem. We have an election with m ballots and n candidates. Each ballot is a binary vector $v \in \{0,1\}^n$, with the meaning that the ith coordinate of v is 1 if the voter approves of candidate i. For two binary vectors v_i, v_j of the same length, let $H(v_i, v_j)$ denote their Hamming distance, which is the number of coordinates in which they differ. For a vector $v \in \{0,1\}^n$, we will denote by $\mathrm{wt}(v)$ the number of coordinates that are set to 1 in v. The maxscore of a binary vector is defined as the Hamming distance between it and the ballot farthest from it: $\mathrm{maxscore}(v) \equiv \max_i H(v, v_i)$ where v_i is the ith ballot. We first define the problem in its generality.

> **Problem [Bounded-size Minimax (BSM(k_1, k_2))]** Given m ballots, $v_1, \ldots, v_m \in \{0,1\}^n$, and 2 integers k_1, k_2, with $0 \leq k_1, k_2 \leq n$, find a vector v^* such that $k_1 \leq \mathrm{wt}(v^*) \leq k_2$ so as to minimize $\mathrm{maxscore}(v^*)$.

Clearly BSM includes as a special case the endogenous version, which is BSM$(0, n)$, *i.e.*, no restrictions on the size of the committee. Also, since in some committee elections, the size of the committee to be elected is fixed (*e.g.*,

the Game Theory Society elections), we are interested in the following variant of BSM with $k_1 = k_2 = k$:

> **Problem [Fixed-size Minimax (FSM(k))]** Given m ballots, $v_1, \ldots, v_m \in \{0,1\}^n$, and an integer k with $1 \leq k \leq n$, find a vector v^* of weight k so as to minimize maxscore(v^*).

In this preliminary version, we focus on elections with committees of fixed size and report our findings for FSM. We briefly mention in the relevant sections throughout the paper as well as in Section 6 which of our results extend to the general BSM problem.

As we show in the next section, BSM and FSM are **NP**-hard. Therefore, a natural approach is to focus on polynomial-time approximation algorithms. We use the standard notion of approximation algorithms, defined below:

Definition 1. *An algorithm for a minimization problem achieves an approximation ratio (or factor) of α ($\alpha \geq 1$), if for every instance of the problem the algorithm outputs a solution with cost at most α times the cost of an optimal solution.*

3 NP-hardness and Approximation Algorithms

We first show that it is unlikely to have a polynomial-time algorithm for the minimax solution. In fact for the endogenous version of BSM, BSM$(0, n)$, **NP**-hardness has already been established by Frances and Litman in [11], where the problem is stated in the context of coding theory. It follows that BSM in general is **NP**-hard. We next show that FSM is also **NP**-hard.

Theorem 1. *FSM is* **NP***-hard.*

Proof. Suppose we had a polynomial-time algorithm for FSM. Then we could run such an algorithm first with $k = 0$, then with $k = 1$ and so on up to $k = n$ and output the best solution. That would give an optimal solution for BSM$(0, n)$. Hence FSM is also **NP**-hard. An alternative proof for the **NP**-hardness of FSM (and consequently of BSM as well) via a reduction from Vertex Cover was also obtained by LeGrand [16]. □

FSM(k) can be solved in polynomial time if k is an absolute constant, since then we can just go through all the $\binom{n}{k}$ different committees and output the best one. Also, if m is an absolute constant then we can express the problem as an integer program with a constant number of constraints, which by a result of Papadimitriou [18] can be solved in polynomial time.

The standard approach in dealing with **NP**-hard problems is to search for approximation algorithms. We will now show that a very simple and fast algorithm achieves an approximation ratio of 3 for FSM(k), for every k. In fact, we will see that the algorithm has a factor of 3 for approval voting problems with much more general constraints.

Before stating the algorithm we need to introduce some more notation. Given a vector v, we will say that u is a *k-completion* of v, if $\mathrm{wt}(u) = k$, and $H(u, v)$ is the minimum possible Hamming distance between v and any vector of weight k. It is very easy to obtain a k-completion for any vector v: if $\mathrm{wt}(v) < k$, then pick any $k - \mathrm{wt}(v)$ coordinates in v that are 0 and set them to 1; if $\mathrm{wt}(v) > k$ then pick any $\mathrm{wt}(v) - k$ coordinates that are set to 1 and set them to 0.

The algorithm is now very simple to state: Pick arbitrarily one of the m ballots, say v_j. Output a k-completion of v_j, say u.

Obviously the algorithm runs in time $O(n)$, independent of the number of voters.

Theorem 2. *The above algorithm achieves an approximation ratio of* 3.

Proof. Let v^* be an optimal solution ($\mathrm{wt}(v^*) = k$) and let $\mathrm{OPT} = \mathrm{maxscore}(v^*) = \max_i H(v^*, v_i)$ be the maximum distance of a ballot from the optimal solution. Let v_j be the ballot picked by the algorithm and let u be the k-completion of v_j that is output by the algorithm. We need to show that for every i, $H(u, v_i) \leq 3\,\mathrm{OPT}$. By the triangle inequality, we know that for every $1 \leq i \leq m$, $H(u, v_i) \leq H(u, v_j) + H(v_j, v_i)$. By applying the triangle inequality again we have:

$$H(u, v_i) \leq H(u, v_j) + H(v_j, v^*) + H(v^*, v_i)$$

Since v^* is an optimal solution, we have that $H(v^*, v_i) \leq \mathrm{OPT}$ and $H(v^*, v_j) \leq \mathrm{OPT}$. Also since u is a k-completion of v_j, by definition $H(u, v_j) \leq H(v^*, v_j) \leq \mathrm{OPT}$. Hence in total we obtain that $H(u, v_i) \leq 3\,\mathrm{OPT}$, as desired. □

Remark 1. *Note that if we know that there is at least one voter of weight k, say $\mathrm{wt}(v_j) = k$, then we can prove that the algorithm achieves a ratio of* 2*, since then $u = v_j$ and we need to apply triangle inequality only once.*

Remark 2. *The algorithm can be easily adapted to give a ratio of* 3 *for the BSM version too. We only need to modify the notion of a k-completion accordingly. In fact, for BSM$(0, n)$, we can show that the ratio will be* 2*.*

Note also that the analysis shows that there can be many different solutions that constitute a 3-approximation, since every ballot can potentially have many different k-completions.

Remark 3. Generalized Constraints: *One may define an approval voting problem with constraints that are more general than simply those on the size of the committee (as in BSM). For example, one may have constraints on the number of members elected from a particular subgroup of candidates (quotas), or constraints which require exactly one out of two particular candidates to be in the committee (XOR constraints). Suppose, for any vote vector v, we can compute in polynomial time a* feasible-completion *of v, which is a committee*

that satisfies the constraints, and is closest to v in Hamming distance. Then, we can extend our algorithm to this setting in a natural manner, and prove that it provides a factor 3 approximation.

We are not aware of any better approximation algorithm for FSM. The endogenous version BSM$(0, n)$, admits a Polynomial Time Approximation Scheme (PTAS), *i.e.*, for every constant ϵ, there exists a $(1+\epsilon)$-approximation, which is polynomial in n and m and exponential in $1/\epsilon$. The PTAS was obtained in [17], in the context of computational biology. Before that, constant-factor approximations for BSM$(0, n)$ had been obtained in [12] and [15]. We believe that algorithms with such better factors may also be obtainable for FSM(k).

4 Local Search Heuristics for Fixed-size Minimax

Even though the algorithm of Section 3 gives us a theoretical worst-case guarantee (in fact, we may even have a better performance in practice for some instances), a factor 3-approximation may still be far away from acceptably good outcomes. In this section we focus on polynomial-time heuristics, which turn out to perform very well in practice, if not optimally, even though we cannot obtain an improved theoretical worst-case guarantee. The heuristics that we will investigate are based on local search; some of them use the 3-approximation as a starting point and retain its ratio guarantee.

4.1 A Framework for FSM Heuristics

Our overall heuristic approach is as follows. We start from a binary vector (picked according to some rule) and then we investigate if neighboring solutions to the current one improve the current maxscore. The local moves that we allow are removing some candidates from the current committee and adding the same number of candidates in, from the set of candidates who do not belong to the current committee:

1. Start with some $c \in \{0,1\}^n$.
2. Repeat until maxscore(c) does not change for n loop iterations:
   (a) Let A be the set of all binary vectors reachable from c by flipping up to p number of 0-bits of c to 1 and p 1-bits to 0, where p is an integer constant. (Note that c will necessarily be a member of A.)
   (b) Let $A^\star$ be the set that includes all members of A with smallest maxscore.
   (c) Choose at random one member of $A^\star$ and make it the new c.
3. Take c as the solution.

It is obviously important that the heuristic find a solution in time polynomial in the size of the input. In the worst case, the loop in the heuristic could run for n iterations for each step down in maxscore, so even if the maxscore of the initial c is the largest possible, n, no more than $O(n^2)$ iterations of the loop will be made. Each loop iteration runs in $O(mn^{2p+1})$ time, since the number of swaps to be considered is $O(n^{2p})$ and calculating the maxscore of each takes $O(mn)$ time, so the worst-case running time for the heuristic is $O(mn^{2p+3})$, which is of course polynomial in m and n as long as p is constant.

This heuristic framework has two parameters: the starting point for the binary vector c and the constant p. While many combinations are possible, we will investigate using four different approaches to determining the c starting point and two values of p—1 and 2—resulting in eight specific heuristics. The c starting points are

1. A fixed-size-minisum solution: the set of the k candidates most approved on the ballots.
2. The FSM 3-approximation presented above: a k-completion of a ballot.
3. A random set of k candidates.
4. A k-completion of a ballot with highest maxscore.

For approach 2, the ballot and k-completion are not chosen randomly: Of the ballots with Hamming weight nearest to k, the v^* minimizing $\text{sumscore}(v) \equiv \sum_i H(v^*, v_i)$ is chosen, and bits flipped are chosen to minimize resulting sumscore. The endogenous minimax equivalent of each of these approaches was investigated by LeGrand [16].

We will use the notation $h_{i,j}$ to refer to the heuristic with starting point i and $p = j$. For example, $h_{3,1}$ is the heuristic that starts with a random set of k candidates and swaps at most one 0-bit with one 1-bit at a time.

4.2 Evaluating the Heuristics

We show that the heuristics find good, if not optimal, winner sets on average. The approach is as follows. Given n, m and k, some large number of simulated elections are run. For each election, m ballots of n candidates are generated according to some distribution. The maxscores of the optimal minimax set and the winner sets found using each of the heuristics are then calculated.

We used two ballot-generating distributions: "unbiased" and "biased". The unbiased distribution simply sets each bit on each ballot to 0 or 1 with equal probability, like flipping an unbiased coin. The biased distribution generates for each candidate two approval probabilities, π_1 and π_2, between 0 and 1 with uniform randomness. The ballots are then divided into three groups. 40% of the ballots are generated according to the π_1 values; that is, each ballot approves each candidate with probability equal to its π_1 value. Another 40%

of the ballots are generated according to the π_2 values, and the remaining 20% are generated as in the unbiased distribution.

We ran 5000 simulated elections in each of seven different configurations, varying n, m, k and the ballot-generating distribution. In the tables of results below, the last column gives the results of running the heuristics 5000 times each on the ballots from the 2003 Game Theory Society council election.

Table 1 gives the highest realized approximation ratio (maxscore found divided by optimal maxscore) found over all 5000 elections for each heuristic, our 3-approximation (with ballot and flipped bits chosen at random), the minisum set (for comparison), and a maximax set. A maximax set is a set of size k that has the highest possible maxscore; it can be found by choosing a ballot with Hamming weight nearest to $n - k$ and performing a $(n - k)$-completion on it.

It can be seen that our 3-approximation in practice performs appreciably better than its guarantee—its ratio was less than 2 for every simulated election. (We were able to find instances of ratio-3 performance for smaller values of n, *e.g.*, 6.) As Table 1 shows, the heuristics reliably find solutions with ratios well below 2, but the average ratios found, given in Table 2, show that the average performance of the heuristics is more impressive still.

Finally, we compared the maxscores found by the heuristics with the worst possible maxscore of a winner set, and scaled them so that the maxscore of the exact minimax set becomes 100% and that of a maximax set becomes 0%, giving a more intuitive performance metric for heuristics. For example, if the minimax set has a maxscore of 12, a maximax set has a maxscore of 20 and a heuristic finds a solution with maxscore 13, the heuristic's scaled performance for that election will be $(20 - 13)/(20 - 12) = 87.5\%$. The averages of these scaled performances can be found in Table 3.

We draw the following conclusions from our experiments.

- The heuristics perform well. Given the ballot distributions we used, very rarely would a heuristic find a solution that is unacceptably poorer than the optimal minimax solution. In particular, $h_{2,1}$ and $h_{2,2}$ vastly outperform the plain 3-approximation (while retaining its ratio-3 guarantee) with only a modest increase in running time.

- The heuristics perform significantly better on average when $p = 2$ than when $p = 1$. Increasing p further can be expected to improve performance further, at the expense of increased running time.

- Comparing the performance of the heuristics with equal p, all four perform similarly overall, but the best c-starting-point approach on average seems to be the first (a fixed-size-minisum solution); it significantly outperforms the other three sometimes (*e.g.*, when $p = 1$ in the unbiased-coin cases with 50 ballots) and is never outperformed by them with any statistical significance.

Table 1: Largest approximation ratios found for local search heuristics

n	20	20	24	20	20	24
k	10	10	12	10	10	12
m	50	200	50	50	200	161
ballots	unbiased	unbiased	unbiased	biased	biased	GTS 2003
minimax	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$h_{1,1}$	1.1818	1.0769	1.1538	1.2000	1.0909	1.0714
$h_{2,1}$	1.1818	1.0769	1.1538	1.2000	1.1818	1.0714
$h_{3,1}$	1.1818	1.0769	1.1538	1.2000	1.1818	1.0714
$h_{4,1}$	1.1818	1.0769	1.1538	1.2000	1.1818	1.0714
$h_{1,2}$	1.0909	1.0769	1.0769	1.1000	1.0833	1.0000
$h_{2,2}$	1.0909	1.0769	1.0769	1.1000	1.0833	1.0000
$h_{3,2}$	1.0909	1.0769	1.0769	1.1000	1.0833	1.0000
$h_{4,2}$	1.0909	1.0769	1.0769	1.1000	1.0833	1.0000
3-approx.	1.6667	1.4615	1.6154	1.8182	1.5833	1.3571
minisum	1.5455	1.4615	1.6923	1.6364	1.5833	1.2143
maximax	1.8182	1.5385	1.8462	2.2222	1.8182	1.7143

Table 2: Average approximation ratios found for local search heuristics

n	20	20	24	20	20	24
k	10	10	12	10	10	12
m	50	200	50	50	200	161
ballots	unbiased	unbiased	unbiased	biased	biased	GTS 2003
minimax	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$h_{1,1}$	1.0058	1.0320	1.0093	1.0083	1.0210	1.0012
$h_{2,1}$	1.0118	1.0365	1.0147	1.0112	1.0251	1.0017
$h_{3,1}$	1.0122	1.0370	1.0151	1.0122	1.0262	1.0057
$h_{4,1}$	1.0117	1.0364	1.0149	1.0116	1.0262	1.0059
$h_{1,2}$	1.0004	1.0129	1.0011	1.0004	1.0025	1.0000
$h_{2,2}$	1.0004	1.0164	1.0014	1.0005	1.0029	1.0000
$h_{3,2}$	1.0004	1.0164	1.0018	1.0005	1.0031	1.0000
$h_{4,2}$	1.0003	1.0167	1.0014	1.0006	1.0029	1.0000
3-approx.	1.2477	1.1871	1.2567	1.3121	1.2424	1.3571
minisum	1.1650	1.1521	1.1665	1.2119	1.1932	1.2143
maximax	1.6746	1.4895	1.7320	1.8509	1.6302	1.7143

Table 3: Average scaled performance of local search heuristics

n	20	20	24	20	20	24
k	10	10	12	10	10	12
m	50	200	50	50	200	161
ballots	unbiased	unbiased	unbiased	biased	biased	GTS '03
$h_{1,1}$	99.18%	94.05%	98.83%	99.07%	96.86%	99.82%
$h_{2,1}$	98.33%	93.23%	98.11%	98.74%	96.24%	99.77%
$h_{3,1}$	98.27%	93.13%	98.06%	98.62%	96.06%	99.20%
$h_{4,1}$	98.33%	93.24%	98.08%	98.68%	96.08%	99.18%
$h_{1,2}$	99.95%	97.60%	99.87%	99.95%	99.63%	100.00%
$h_{2,2}$	99.95%	96.96%	99.83%	99.94%	99.57%	100.00%
$h_{3,2}$	99.95%	96.95%	99.79%	99.94%	99.54%	100.00%
$h_{4,2}$	99.96%	96.89%	99.83%	99.94%	99.57%	100.00%
3-approx.	63.36%	62.31%	65.04%	63.36%	61.73%	50.00%
minisum	75.57%	69.40%	77.29%	75.04%	69.49%	70.00%

5 Manipulation

Gibbard [13] and Satterthwaite [22] proved independently that any election system that chooses exactly one winner from a slate of more than two candidates and satisfies a few obviously desirable assumptions (such as an absence of bias for some candidates over others) is sometimes manipulable. In other words, there exist situations under any reasonable single-winner system in which some voters can gain better outcomes for themselves by voting insincerely.

Happily, the Gibbard–Satterthwaite theorem does not apply to the minimax and minisum solutions since they are free to choose winner sets of any size. In fact, the minisum procedure is completely nonmanipulable when any set of winners is allowed, as shown by Brams *et al.* [4]. This is true because a minisum election with n candidates is exactly equivalent to n elections of two "candidates" each: approve or disapprove that candidate. Since a voter's decision to approve or disapprove one candidate has absolutely no effect on whether other candidates are chosen as winners, there is no more effective strategy than voting sincerely. Consequently, it is reasonable to expect a set of minisum ballots to have been sincerely voted.

Unfortunately, in addition to being possibly hard to compute exactly, the minimax solution is easily shown to be manipulable for the FSM version.

Definition 2. *Fix an approval voting algorithm A and a set of ballots $\mathbf{v} = (v_1, v_2, ..., v_m)$. Fix a voter i, and let $\mathbf{v^{-i}}$ denote the ballots of the rest of the voters. The loss $L^i_A(\mathbf{v})$ of voter i is defined as $H(v_i, A(\mathbf{v}))$. Algorithm A is said to be manipulable if there exist ballots $\mathbf{v}$, a voter i, and a ballot $v' \neq v_i$, s.t. $L^i_A(v_i, \mathbf{v^{-i}}) > L^i_A(v', \mathbf{v^{-i}})$.*

Theorem 3. *Any algorithm that computes an optimal solution for FSM is manipulable.*

Proof. Consider the following set of *sincere* ballots:

00110, **00011**, **00111**, **00001**, **10111**, **01111**

The minimax winner sets of size 2 are **00011** and **00101** with a maxscore of 2. The first voter, however, could manipulate the result by voting the insincere ballot **11110**. In that case, it can be checked that the optimal solution of size 2 is **00110**, which is exactly the most preferred outcome of the first voter. □

An analogous example for the endogenous version was provided by LeGrand [16]. These examples illustrate a general guideline to manipulating a minimax election: If there are candidates of which the majority disapproves, a voter may be able to vote safely in favor of those candidates to force more agreement with his relatively controversial choices. Put another way, if the minimax set can be seen as a kind of average of all ballots, a voter can move his ballot farther away from the current consensus to drag it closer to his ideal outcome. The minimax solution is extremely sensitive to "outliers" compared to the minisum solution, in much the same way that the average of a sample of data is more sensitive to outliers than the median.

Although algorithms that always compute an optimal minimax solution are manipulable, the same may not be true if we allow approximation algorithms. The following theorem shows that we can have nonmanipulable algorithms if we are willing to settle for approximate solutions.

Theorem 4. *The voting procedure that results from using the* 3*-approximation algorithm described in Section 3 is nonmanipulable.*

Proof. The algorithm picks a ballot v_j at random and outputs a k-completion of v_j. For a voter i, if the algorithm did not pick v_i, then the voter cannot change the output of the algorithm by lying. Furthermore, if the algorithm did pick v_i, then the best outcomes of size k for v_i are precisely all the k-completions of v_i. Therefore, by lying, the voter cannot possibly alter the outcome to his benefit. □

We conjecture that the heuristics of Section 4 are also hard to manipulate. Although we do not have a proof for this, our intuition is the following. The heuristics use a lot of randomization—in all of them, either the starting point or the local move is based on a random choice. It therefore seems unlikely for a voter to be able to change his vote in such a way that the random choices of the algorithms will (even in expectation) work towards his benefit.

The above theorems give rise to the following question:

Question 1. *What is the smallest value of* α *for which there exists a nonmanipulable polynomial-time approximation algorithm with ratio* α*?*

Another interesting question is whether there exist algorithms (either exact or approximate) which are **NP**-hard to manipulate (i.e., although they are manipulable, the voter would have to solve an **NP**-hard problem in order to cheat). See Bartholdi *et al.* [1, 2] as well as more recent work [5, 6, 7, 8] along this line of research. In another recent work [20], average-case complexity is introduced as a complexity measure for manipulation instead of worst-case complexity (**NP**-hardness).

6 Discussion and Future Work

We have initiated a study of the computational issues involved in committee elections. Our results, along with the analysis of the endogenous version by LeGrand [16], show that local search heuristics perform very well in approximating the minimax solution in polynomial time.

There are still many interesting directions for future research. In terms of heuristic approaches, we are planning to adjust our heuristics for the weighted version of the minimax solution, as introduced by Brams *et al.* [4]. This version takes into account both the number of voters that vote each distinct ballot and the proximity of each ballot to the other voters' ballots. We are also investigating variations of local search that may improve even more the performance, *e.g.*, can there be a better starting point in our heuristics, or can we enrich the set of local moves without increasing too much the running time? Another interesting topic would be to compare local search with other heuristic approaches that could be adapted for our problem, like simulated annealing or genetic algorithms.

In terms of theoretical results, the most compelling question is to determine the best approximation ratio that can be achieved in polynomial time for the minimax solution. The questions stated in Section 5 regarding manipulation would also be interesting to pursue.

7 Acknowledgements

We would like to than Eric van Damme, secretary-treasurer of the Game Theory Society, for letting us use the ballot data of the 2003 Game Theory Society council election in our experiments. We would also like to thank Steven Brams for introducing us to the problem and for his valuable comments and pointers to the literature.

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] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. How hard is it to control an election? *Mathematical Computational Modeling*, 16(8/9):27–40, 1992.

[3] S. J. Brams, D. M. Kilgour, and M. R. Sanver. A minimax procedure for negotiating multilateral treaties. In M. Wiberg, editor, *Reasoned Choices: Essays in Honor of Hannu Nurmi*. Finnish Political Science Association, 2004.

[4] S. J. Brams, D. M. Kilgour, and M. R. Sanver. A minimax procedure for electing committees. manuscript, 2006.

[5] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-03)*, pages 201–214, Bloomington, Indiana, 2003.

[6] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 314–319, Edmonton, Canada, 2002.

[7] 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, Acapulco, Mexico, 2003.

[8] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *The 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, pages 206–215, Sanya, Hainan, China, Dec. 2005.

[9] E. Ephrati and J. Rosenschein. The clarke tax as a consensus mechanism among automated agents. In *AAAI*, pages 173–178, 1991.

[10] E. Ephrati and J. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *IJCAI*, pages 423–429, 1993.

[11] M. Frances and A. Litman. On covering problems of codes. *Theory of Computing Systems*, 30:113–119, Mar. 1997.

[12] L. Gasieniec, J. Jansson, and A. Lingas. Efficient approximation algorithms for the Hamming center problem. In *SODA*, 1999.

[13] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(3):587–601, May 1973.

[14] D. M. Kilgour, S. J. Brams, and M. R. Sanver. How to elect a representative committee using approval balloting. In B. Simeone and F. Pukelsheim, editors, *Mathematics and Democracy: Recent Advances in Voting Systems and Collective Choice*. Springer, forthcoming, 2007.

[15] J. Lanctot, M. Li, B. Ma, S. Wang, and L. Zhang. Distinguishing string selection problems. *Information and Computation*, 185:41–55, 2003.

[16] R. LeGrand. Analysis of the minimax procedure. Technical Report WUCSE-2004-67, Department of Computer Science and Engineering, Washington University, St. Louis, Missouri, Nov. 2004.

[17] M. Li, B. Ma, and S. Wang. Finding similar regions in many strings. In *STOC*, pages 473–482, 1999.

[18] C. H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981.

[19] D. Pennock, E. Horvitz, and C. L. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *AAAI*, pages 729–734, 2000.

[20] A. D. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. In *AAMAS*, pages 497–504, Hakodate, Japan, 2006.

[21] A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *IJCAI*, Hyderabad, India, 2007.

[22] 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(2):187–217, Apr. 1975.

Rob LeGrand
Washington University in St. Louis
St. Louis, Missouri, U.S.A.
Email: `legrand@cse.wustl.edu`

Evangelos Markakis
University of Toronto
Toronto ON M5S3G4, Canada
Email: `vangelis@cs.toronto.edu`

Aranyak Mehta
IBM Almaden Research Center
San Jose, CA 95120, USA
Email: `mehtaa@us.ibm.com`

Equal Representation in Two-tier Voting Systems*

Nicola Maaser and Stefan Napel

Abstract

The paper investigates how voting weights should be assigned to differently sized constituencies of an assembly. The one-person, one-vote principle is interpreted as calling for a priori equal indirect influence on decisions. The latter are elements of a one-dimensional convex policy space and may result from strategic behavior consistent with the median voter theorem. Numerous artificial constituency configurations, the EU and the US are investigated by Monte-Carlo simulations. *Penrose's square root rule*, which originally applies to preference-free dichotomous decision environments and holds only under very specific conditions, comes close to ensuring equal representation. It is thus more robust than previously suggested.

1 Introduction

The principle of "one person, one vote" is generally taken to be a cornerstone of democracy. It is not clear, however, how this principle ought to be operationalized in practice in terms of determining what are the ideal shares. This paper addresses this problem for *two-tier voting systems* that involve multiple constituencies of different population size. We concentrate on situations in which representatives of constituencies in the higher-level assembly vote as a block (as in the US Electoral College) or in which a single agent represents each constituency but is endowed with a number of votes that somehow reflect population size (as in the EU Council of Ministers). Both boil down to weighted voting.

Although it seems straightforward to allocate weights proportional to population sizes, this ignores the combinatorial properties of weighted voting, which often imply stark discrepancies between *voting weight* and actual *voting power*: In an assembly with simple majority rule and three representatives having weight 47, 43, and 10, all three possess exactly the same number of possibilities to form a winning coalition and hence the same a priori power. Moreover, direct proportionality disregards the possibly nonlinear relationship between population size and an individual's effect on the respective constituency's top-tier policy position.

The most well-known solution to this problem is the one first suggested by Penrose (1946). Starting from the ideal world in which only constituency

* This work is an abbreviated version of a paper forthcoming in *Social Choice and Welfare*. We thank M. Braham, M.J. Holler, participants of the workshop Voting Power & Procedures, Warwick, 2005, and two anonymous referees for their constructive comments.

membership[1] distinguishes voters, Penrose found that if members of any constituency are to have the same a priori chance to indirectly determine the outcome of top-tier decisions, then constituencies' voting weights need to be such that their power at the top-tier as measured by the *Penrose-Banzhaf index* (Penrose 1946; Banzhaf 1965) is proportional to the square root of the respective constituency's population size (also see Felsenthal and Machover 1998, sect. 3.4). This *square root rule* has recently become the benchmark for numerous studies of the EU Council of Ministers (see, e. g. Felsenthal and Machover 2001; 2004, Leech 2002) and it is at least a reference point for investigations concerning the US (see e. g. Gelman, Katz, and Bafumi 2004).

Applying the square root rule has, unfortunately, two weaknesses: First, Penrose's theorem critically depends on equiprobable 'yes' and 'no'-decisions by all voters (or at least a 'yes'-probability which is random and distributed independently across voters with mean exactly 0.5). If the 'yes'-probability is slightly lower or higher, or if it exhibits even minor dependence across voters – say, they are influenced by the same newspapers – then the square root rule may result in highly unequal representation (see Good and Mayer 1975 and Chamberlain and Rothschild 1981). Related empirical studies in fact have failed to confirm the predictions for average closeness of two-party elections which lie behind the square root rule (see Gelman, Katz, and Tuerlinckx 2002 and Gelman, Katz, and Bafumi 2004).

Second, rigorous justifications for using the square root rule as the benchmark have so far concerned only *preference-free binary voting*.[2] But real decisions are rarely binary, e. g., about *either* introducing a tax, building a road, accepting a candidate, introducing affirmative action, etc. *or* not. At least at intermediate levels there is a preference-driven compromise that involves *many* alternative tax levels, road attributes, suitable candidates, degrees of affirmative action, etc.

The first criticism has been addressed in the literature, at least in abstract normative terms. Namely, one can argue that constitutional design should be carried out behind a thick veil of ignorance in which no particular type of dependence or modification of equiprobability (which follows from the principle of insufficient reason) is justified. Regarding the second issue, this paper is to our knowledge the first to investigate equal representation for non-binary decisions that possibly involve strategic behavior.

We consider policy alternatives from a finite interval. Our formal model (see Section 2) imposes two key assumptions: first, the policy advocated by the top-tier representative of any given constituency coincides with the ideal point of the respective constituency's *median voter* (or the constituency's *core*).

[1] We take the constituency configuration to be given exogenously. See, e. g., Epstein and O'Halloran (1999) on constructing majority-minority voting districts along ethnic, religious, or social lines.

[2] For rigorous, very comprehensive treatments of the binary or *simple-game* world see Felsenthal and Machover (1998) or Taylor and Zwicker (1999). – The former (pp. 72ff) also justify the square root rule regarding voting weights by its minimal expected *majority deficit*.

Second, the decision taken at the top tier is the position of the *pivotal representative* (or the assembly's *core*), with pivotality determined by the weights assigned to constituencies and a 50% decision quota. The respective core is meant to capture the result of strategic interaction. As long as this is a reasonable approximation, the actual systems determining collective choices are undetermined and could even differ across constituencies.

In the benchmark case of voters with independent most-preferred policies, a given individual's chance to be pivotal at the bottom tier is inversely proportional to the respective constituency's population size. This makes it necessary and sufficient for equal representation of voters that the probability of any given constituency being pivotal at the top tier is proportional to its size.[3]

The population size of a constituency affects the distribution of its median. A given voter's chance to be doubly pivotal thus becomes a rather complex function of (the order statistics of) *differently distributed* independent random variables. This makes a neat analytical statement similar to Penrose's rule exceptionally hard and likely impossible, except for special limit situations. We therefore resort to Monte-Carlo simulation (see Section 3). Considering a vast number of randomly generated population configurations as well as recent data for the EU and the US, top-tier weights proportional to the square root of population turn out optimal for most practically relevant population configurations. Even for extreme artificial cases, the rule yields good results and becomes optimal if the number of constituencies gets large.

Our surprising main finding is thus that the square root rule is a much more robust norm for egalitarian design of two-tier voting systems than previous analysis suggests. In particular, it continues to apply in the presence of many finely graded policy alternatives and strategic interactions consistent with the median voter theorem. To the extent that this still produces independent median voters, the rule is even robust to the introduction of preference dependence within or across constituencies.

2 Model

Consider a large population of *voters* partitioned into m *constituencies* $\mathcal{C}_1, \ldots, \mathcal{C}_m$ with $n_j = |\mathcal{C}_j| > 0$ members each. Voters' preferences are single-peaked with *ideal point* λ_i^j (for $i = 1, \ldots, n_j$ and $j = 1, \ldots, m$) in a bounded convex one-dimensional *policy space* normalized to $X \equiv [0, 1]$. Assume for simplicity that all n_j are odd numbers.

For any random policy issue, let $\cdot : n_j$ denote the permutation of voter numbers in constituency $\mathcal{C}_j$ such that $\lambda^j_{1:n_j} \leq \ldots \leq \lambda^j_{n_j:n_j}$ holds. In other words, $k : n_j$ denotes the k-th leftmost voter in $\mathcal{C}_j$ and $\lambda^j_{k:n_j}$ denotes the k-th leftmost ideal point (i. e., $\lambda^j_{k:n_j}$ is the *k-th order statistic* of $\lambda^j_1, \ldots, \lambda^j_{n_j}$).

[3] If voters' utility is linear in distance, the criterion also guarantees equal expected utility, i. e., a priori *power* and expected *success* are then perfectly aligned. See Laruelle, Martínez, and Valenciano (2006) for a conceptual discussion of the latter.

A policy $x \in X$ is decided on by an *electoral college* $\mathcal{E}$ consisting of one representative from each constituency. Without going into details, we assume that the representative of $\mathcal{C}_j$, denoted by j, adopts the ideal point of his constituency's *median voter*,[4] denoted by $\lambda^j \equiv \lambda^j_{(n_j+1)/2:n_j}$. Let $\lambda^{k:m}$ denote the k-th leftmost ideal point amongst all the representatives (i. e., the k-th order statistic of $\lambda^1, \ldots, \lambda^m$).

In the top-tier assembly or electoral college $\mathcal{E}$, each constituency $\mathcal{C}_j$ has *voting weight* $w_j \geq 0$. Any subset $S \subseteq \{1, \ldots, m\}$ of representatives which achieves a combined weight $\sum_{j \in S} w_j$ above $q \equiv \frac{1}{2} \sum_{j=1}^{m} w_j$, i. e. a *simple majority* of total weight, can implement a policy $x \in X$.

Consider the random variable P defined by

$$P \equiv \min \Big\{ r \in \{1, \ldots, m\} \colon \sum_{k=1}^{r} w_{k:m} > q \Big\}.$$

Player $P\!:\!m$'s ideal point, $\lambda^{P:m}$, is the unique policy that beats any alternative $x \in X$ in a pairwise majority vote, i. e. constitutes the *core* of the voting game defined by weights and quota.[5] Without detailed equilibrium analysis of any decision procedure that may be applied in $\mathcal{E}$ (see Banks and Duggan 2000 for sophisticated non-cooperative support of policy outcomes inside or close to the core), we assume that the policy agreed by $\mathcal{E}$ is in the core, i. e. it equals the ideal point of the *pivotal representative* $P\!:\!m$.

In this setting we consider the following egalitarian norm: *Each voter in any constituency should have an equal chance to determine the policy implemented by the electoral college.* Or, more formally, there should exist a constant $c > 0$ such that

$$\forall j \in \{1, \ldots, m\} \colon \forall i \in \mathcal{C}_j \colon \Pr\big(j = P\!:\!m \wedge i = (n_j + 1)/2\!:\!n_j\big) \equiv c. \qquad (1)$$

We would like to answer the following question: which allocation of weights $w_1, \ldots, w_m$ satisfies this norm (at least approximately) for an arbitrary given partition of an electorate into m constituencies? In other words we search for an analogue of Penrose's (1946) rule, which calls for proportionality of a constituency's Penrose-Banzhaf index[6] and square root of population.

The probability of a voter's double pivotality in (1) depends on the distribution of all voters' ideal points. Though in practice ideal points in different

[4] We are aware of this not being appropriate in all contexts. – The possibility that two ideal points exactly coincide, in which case the median voter (in contrast to the median policy) is not well-defined, is ignored. This is innocuous for any continuous ideal point distribution.

[5] Things are more complicated if $q > \frac{1}{2} \sum_{j=1}^{m} w_j$ is assumed. Then, the complement of a losing coalition need no longer be winning. In this case there may not exist *any* policy $x \in X$ which beats all alternatives $x' \neq x$ despite unidimensionality of X and single-peakedness of preferences.

[6] This index equals a constituency's probability of being pivotal under equiprobable random 'yes'-or-'no' votes at the top tier. Conditions for when this is approximately the voting weight are given by Lindner and Machover (2004). In general, implementing Penrose's square root rule requires numerical solution of the *inverse problem* of finding weights which induce a desired power distribution (see e. g. Leech 2003).

constituencies may come from different distributions on X and may exhibit various dependencies, it is appealing from a normative constitutional-design point of view to presume that the ideal points of all voters in all constituencies are *independently and identically distributed* (i. i. d.).

Given that voters' ideal points in constituency $\mathcal{C}_j$ are i. i. d., each voter $i \in \mathcal{C}_j$ has the same probability to be its median. Hence,

$$\forall j \in \{1, \ldots, m\} \colon \forall i \in \mathcal{C}_j \colon \Pr\big(i = (n_j + 1)/2 \colon n_j\big) = \frac{1}{n_j}.$$

Because the events $\{i = (n_j + 1)/2 \colon n_j\}$ and $\{j = P \colon m\}$ are independent, one can thus write (1) as

$$\forall j \in \{1, \ldots, m\} \colon \frac{\Pr(j = P \colon m)}{n_j} \equiv c. \tag{2}$$

Representatives' ideal points $\lambda^1, \ldots, \lambda^m$ are independently but (except in the trivial case $n_1 = \ldots = n_m$) *not* identically distributed. If all voter ideal points come from the (arbitrary) identical distribution F with density f, then $\mathcal{C}_j$'s median position is asymptotically normally distributed (see e. g. Arnold et al. 1992) with mean $\mu^j = F^{-1}(0.5)$ and standard deviation

$$\sigma^j = \frac{1}{2\, f(F^{-1}(0.5))\sqrt{n_j}}.$$

So, the larger a constituency $\mathcal{C}_j$ is, the more concentrated is the distribution of its median voter's ideal point, λ^j, on the median of the underlying ideal point distribution (assumed to be identical for all λ_i^j). This makes the representative of a larger constituency on average more central in the electoral college and more likely to be pivotal in it for a given weight allocation.

It is important to observe that the assumption of the respective *collective preferences* having an identical a priori distribution is inconsistent with the assumption that all *individual preferences* are a priori identically distributed. We find the latter assumption considerably more fitting and will assume i. i. d. ideal points for all bottom-tier voters throughout this paper.

Probability $\Pr(j = P \colon m)$ in (2) depends both on the different distributions of representatives' ideal points (essentially the standard deviations σ^j determined by constituency sizes n_j) and the voting weight assignment. This makes computation of the probability of a given constituency $\mathcal{C}_j$ being pivotal a complex numerical task even for the most simple case of *uniform weights*, in which the representative of $\mathcal{C}_j$ with *median* top-tier ideal point is always pivotal, i. e. $P \equiv (m + 1)/2$ for odd m. Define $N^j \equiv \{1, \ldots, j - 1, j + 1, \ldots, m\}$ as the index set of all constituencies except $\mathcal{C}_j$. Then, the probability of constituency $\mathcal{C}_j$ being pivotal is

$$\begin{aligned} \Pr\big(j = (m+1)/2 \colon m\big) &= \Pr\big(\text{exactly } \tfrac{m-1}{2} \text{ of the } \lambda^k, k \neq j, \text{ satisfy } \lambda^k < \lambda^j\big) \\ &= \int \sum_{\substack{S \subset N^j, \\ |S| = (m-1)/2}} \prod_{k \in S} F_k(x) \cdot \prod_{k \in N^j \setminus S} (1 - F_k(x)) \cdot f_j(x)\, dx, \end{aligned} \tag{3}$$

where f_j and F_j denote the density and cumulative density functions of λ^j ($j = 1, \ldots, m$). It seems feasible (but is beyond the scope of this paper) to provide an asymptotic approximation for this probability as a function of constituency sizes $n_1, \ldots, n_m$ for special cases, e. g. for $n_2 = \ldots = n_m$ (hence $F_2 = \ldots = F_m$). However, we doubt the existence of a reasonable approximation for arbitrary configurations $(n_1, \ldots, n_m)$, let alone the case of weighted voting ($P \not\equiv (m+1)/2$). A purely analytical investigation of the model is therefore unlikely to produce much insight. The following section for this reason uses Monte-Carlo simulation in order to approximate the probability of any constituency $\mathcal{C}_j$ being pivotal for given partition of an electorate or *configuration* $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ and a fixed weight vector $(w_1, \ldots, w_m)$. Based on this, we try to find weights $(w_1^*, \ldots, w_m^*)$ which approximately satisfy the two equivalent equal representation conditions (1) and (2).

3 Simulation results

The probability $\pi_j \equiv \Pr(j = P\!:\!m)$ can be viewed as the *expected value* of the random variable $H_j \equiv g_j^w(\lambda^1, \ldots, \lambda^m)$ which equals 1 if $j = P : m$ holds for given weight vector w and realized median ideal points $\lambda^1, \ldots, \lambda^m$, and 0 otherwise. The *Monte-Carlo method* (Metropolis and Ulam 1949) then exploits the fact that the empirical average of s independent draws of H_j, $\bar{h}_j^s = \frac{1}{s}\sum_{l=1}^{s} h_j^l$, converges to H_j's theoretical expectation $\mathbf{E}(H_j) = \pi_j$ by the law of large numbers. The speed of convergence in s can be assessed by the sample variance of $\bar{h}_j^s$. Using the central limit theorem, it is then possible to obtain estimates of π_j with a desired precision (e. g. a 95%-confidence interval) if one generates and analyzes a sufficiently large number of realizations.

To obtain a realization h_j^l of H_j, we first draw m random numbers $\lambda^1, \ldots, \lambda^m$ from distributions $F_1, \ldots, F_m$.[7] Throughout our analysis, we take F_j to be a *beta distribution* with parameters $\big((n_j+1)/2, (n_j+1)/2\big)$. This corresponds to the median of n_j independently $[0,1]$-*uniformly distributed* voter ideal points, i. e. all individual voter positions are assumed to be distributed uniformly.[8] Second, the realized constituency positions are sorted and the pivotal position p is determined. Constituency $\mathcal{C}_{p:m}$ is thus identified as the pivotal player of $\mathcal{E}$. It follows that $h_j^l = 1$ for $j = p\!:\!m$, and 0 for all other constituencies.

The goal is to identify a simple rule for assigning voting weights to constituencies which – if it exists – approximately satisfies equal representation conditions (1) or (2) for various numbers of constituencies m and population

[7] We use a *Java* computer program. The source code is available upon request. Directly drawing the constituency medians λ^j provides a huge computational advantage. Unfortunately, it prevents statements about the population median and, e. g., its average distance to the policy outcome.

[8] The mentioned asymptotic results for order statistics imply that only F's median position and density at the median matter when constituency sizes are large. So below findings are *not* specific to the assumption of uniform distributions at the bottom tier.

configurations $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$. A natural focus is the investigation of *power laws*

$$w_j = {n_j}^{\alpha} \tag{4}$$

with $\alpha \in [0, 1]$. For big m this approximately includes Penrose's square root rule as the special case $\alpha = 0.5$ (see Lindner and Machover 2004 and **?**).[9]

For any given m and population configuration $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ under consideration, we fix α and then approximate π_j by its empirical average $\hat{\pi}_j$ in a run of 10 million iterations. This is repeated for different values of α, ranging from 0 to 1 with a step size of 0.1 or 0.01, in order to find the exponent α which comes 'closest' to implying equal representation for the given configuration.

Our criterion for evaluating distance between the (estimated) probability vector $\hat{\pi} \equiv (\hat{\pi}_1, \ldots, \hat{\pi}_m)$ realized by weights w and the ideal egalitarian vector $\pi^* \equiv (\sum_{k=1}^{m} n_k)^{-1} \cdot (n_1, \ldots, n_m)$ considers cumulative quadratic deviations between the realized and the ideal chances of an *individual*. Any voter in any constituency $\mathcal{C}_j$ would ideally determine the outcome with the same probability $1/\sum_{k=1}^{m} n_k$, but vector $\hat{\pi}$ actually gives him or her the probability $\hat{\pi}_j/n_j$ of doing so. Treating all n_j voters in any constituency $\mathcal{C}_j$ equally then amounts to looking at

$$\sum_{j=1}^{m} n_j \cdot \left(\frac{1}{\sum_{k=1}^{m} n_k} - \frac{\hat{\pi}_j}{n_j} \right)^2 . \tag{5}$$

We refer to measure (5) as *cumulative individual quadratic deviation* below.

3.1 Randomly generated configurations

Table 1 reports the optimal values of α that were obtained for four sets of configurations $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$.[10] For $m \in \{10, 15, 20, 25, 30, 40, 50\}$, constituency sizes $n_1, \ldots, n_m$ were independently drawn from a uniform distribution over $[0.5 \cdot 10^6,\ 99.5 \cdot 10^6]$. Numbers in column (I) are the optimal $\alpha \in \{0, 0.1, \ldots, 0.9, 1\} \subset [0, 1]$, where probabilities $\hat{\pi}_j$ were estimated by a simulation with 10 mio. iterations. Cumulative individual quadratic deviations for optimal α's are shown in brackets. Column (II) reports the respective values obtained for an independent second set of constituency configurations; columns (III) and (IV) do likewise but based on the finer grid $\{0, 0.01, 0.02, \ldots, 0.99, 1\}$ that contains α.[11]

[9] For comparison purposes, we also considered the exact version of Penrose's rule for a selected number of population configurations. Although there are exceptions to this, Penrose's rule tends to perform worse than (4) with the respective optimal exponent α. This extends to $\alpha = 0.5$ when this is close to being optimal. In other cases, e.g., when in fact uniform weights produce the most equal representation, Penrose's square root rule performs better at least than its approximation by $w_j = \sqrt{n_j}$. We leave a more systematic investigation of alternatives to (4) – like "w_j s.t. β_j is proportional to ${n_j}^{\alpha}$" with β_j referring to j's Penrose-Banzhaf index, as suggested by an anonymous referee – for future research.

[10] The configuration draws are independent across different values of m. Thus, the table actually reports optimal values obtained for 28 *independent* configurations.

[11] Hence columns (III) and (IV) each report on $101 \cdot 7$ simulation runs (with 10 mio. iterations each).

# const	(I)	(II)	(III)	(IV)
10	**0.5**	**0.6**	**0.39**	**0.00**
	(1.22×10^{-11})	(1.04×10^{-11})	(2.20×10^{-12})	(2.39×10^{-11})
20	**0.5**	**0.5**	**0.49**	**0.49**
	(4.80×10^{-14})	(8.59×10^{-14})	(5.66×10^{-15})	(6.91×10^{-15})
30	**0.5**	**0.5**	**0.49**	**0.49**
	(1.11×10^{-15})	(5.12×10^{-15})	(7.36×10^{-15})	(2.38×10^{-15})
50	**0.5**	**0.5**	**0.50**	**0.50**
	(3.06×10^{-15})	(4.70×10^{-15})	(3.10×10^{-15})	(3.30×10^{-15})

Table 1: Optimal value of α for uniformly distributed constituency sizes (cumulative individual squared deviations from ideal probabilities in parentheses)

While results for $m = 10$ are still inconclusive, $\alpha \approx 0.5$ emerges as the very robust ideal exponent for larger number of constituencies. The reported cumulative individual quadratic deviations are so small that even if the power laws assumed in (4) do not contain the theoretically best rule for equal representation in our median-voter context (because possibly constituencies' sizes are not the right reference point, but rather something like their Penrose-Banzhaf or Shapley-Shubik index), they allow a sufficiently good approximation for most practical purposes.

Results in Table 1 are strongly suggesting that (an approximation of) Penrose's square root rule holds also in the context of median voter-based policy decisions in $[0, 1]$. But optimality of $\alpha \approx 0.5$ could be an artifact of considering uniformly distributed constituency sizes $n_1, \ldots, n_m$, which perhaps unrealistically makes small constituencies as likely as large ones. We therefore conduct similar investigations using other distributional assumptions.

Constituency sizes seem usually a matter of history, geography, or deliberate design. In the latter case, one might expect them to be clustered around some 'ideal' intermediate level. This makes a (truncated) normal distribution around some value $\bar{n}$ a focal assumption for constituency configurations. Table 2 indicates that, in this case, $\alpha = 0.5$ is no longer the general clear winner from the considered set of parameters $\{0, 0.1, \ldots, 0.9, 1\}$. This is neither very surprising nor – from a square-root-rule point of view – very disturbing: Moderately many and more or less equally sized constituencies give rather little scope for discrimination between constituencies. Assigning slightly larger constituencies substantially more weight risks overshooting the mark, but assigning them only slightly more weight may not translate into an increased number of pivot positions at all. So, first, the optimal α can be expected to be rather sensitive to the precise constituency configuration at hand, especially when a small number of constituencies creates relatively few distinct opportunities to achieve a majority. And, second, in the wide range where extra weight to an above-the-average constituency translates into no or few extra winning coalitions, the objective function is very flat. This is nicely illustrated by Figure 1. Its minimization via Monte Carlo techniques is then particularly sensitive to remaining estimation

# const	(I)	(II)	(III)	(IV)
10	**0.0**	**0.0**	**0.0**	**0.0**
	(1.22×10^{-9})	(1.65×10^{-9})	(9.21×10^{-9})	(1.83×10^{-9})
30	**0.1**	**0.2**	**0.4**	**0.5**
	(1.07×10^{-10})	(1.07×10^{-10})	(6.94×10^{-11})	(6.76×10^{-11})
50	**0.4**	**0.2**	**0.3**	**0.3**
	(1.60×10^{-11})	(7.39×10^{-12})	(3.56×10^{-11})	(4.72×10^{-11})
100	**0.5**	**0.5**	**0.5**	**0.5**
	(1.01×10^{-13})	(2.30×10^{-12})	(1.99×10^{-13})	(3.44×10^{-13})

Table 2: Optimal value of α for normally distributed constituency sizes ($\mu =$ 1 mio., $\sigma =$ 200,000; truncated below 0)

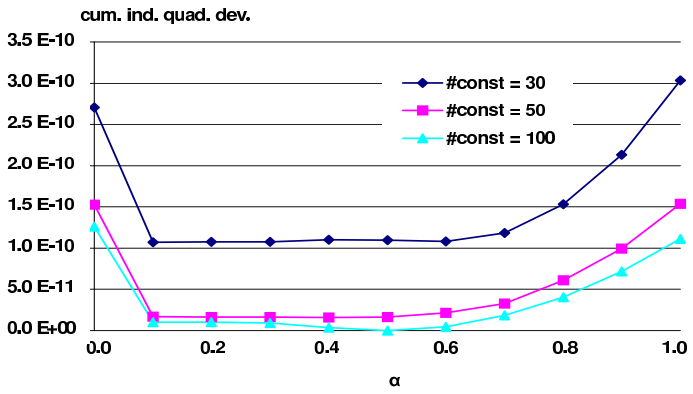


Figure 1: Cumulative individual quadratic deviation in normal-distribution runs (I) for different numbers of constituencies

errors. But note that the importance of these issues decreases as m gets large. This indicates that the applicability of the square root rule rests on enough flexibility regarding the formation of distinct winning coalitions.

When historical or geographical boundaries determine a population partition, a yet more natural distributional benchmark for n_j is a power law such as *Zipf's law* (or *zeta distribution*). In summary, simulations results with constituency sizes drawn from Pareto distributions correspond nicely to those for the uniform distribution as long as the distribution is only moderately skewed. $w_j = \sqrt{n_j}$ performs best and gets close to ensuring equal representation provided that the number of constituencies is sufficiently large. The former is no longer the case for a heavily skewed distribution of constituency sizes, i. e. when there are mostly small constituencies and only one or perhaps two large constituencies (reminiscent of atomic players in an otherwise oceanic game). Giving all constituencies equal weight does reasonably well. As in the normal-distribution case, this problem gets less severe, the greater is the total number of constituencies: For $m = 100$ or larger, $\alpha = 0.5$ turns out to be clearly optimal

even for high skewness.

The above analysis of many different population configurations reveals three things. First, as Table 1 and Figures 1 show, $\alpha = 0.5$ results in representation close to being as equal as possible for the given partition of the electorate. Second, for a moderately large number m of constituencies $\alpha \approx 0.5$ is optimal in the considered class of power laws unless all constituency sizes are very similar (e. g., n_j normally distributed with small variance) or rather similar with one or two outliers (corresponding to a heavily skewed distribution). Third, even in these extreme cases the optimal α converges to 0.5 as m gets large. We now turn to two prominent real-world two-tier voting systems.

3.2 EU Council of Ministers

Together with Commission and Parliament, the Council of Ministers is one of the European Union's chief legislative bodies. It is widely held to be the most influential amongst the three and most voting power analysis concentrates on it.[12] It consists of a national government representative from each of the EU member states, endowed with voting weight that is (weakly) increasing in share of total population.[13]

Figure 2 illustrates the probabilities that representatives from differently sized member states are pivotal in the Council assuming a 50% decision quota and assigning voting weight based on populations size via $w_j = {n_j}^{\alpha}$.[14] In line with above findings for randomly generated two-tier voting systems, $\alpha = 0.5$ performs best amongst all coefficients in $\{0, 0.1, \ldots, 1\}$. The figure shows how close the implied probability of country j being pivotal comes to the respective ideal value, which would implement a priori perfectly equal representation. Only the most populous country, Germany, would be visibly misrepresented (here: over-represented).

Note that this analysis not only puts historical voting patterns and preference similarities between some members behind a veil of ignorance but also, as do the mentioned applied studies, it disregards differences between the bottom-tier voting procedures which determine national governments. For example, the UK uses plurality rule or a "first-past-the-post" system, whilst Germany uses a roughly proportional system.[15] This difference might have a systematic effect on the respective accuracy of our median voter assumption at the constituency level. To the extent that it does not, our findings are robust.

[12] See Felsenthal and Machover (2004), and Leech (2002) for examples. Napel and Widgrén (2006) argue formally that the Commission's and Parliament's positions are nearly irrelevant in the EU25's most common *codecision procedure*.

[13] The current voting rule (based on the Treaty of Nice) is actually quite complex. In addition to standard weighted voting it involves the requirement that the majority weight supporting a policy represents a simple majority of member states and 62% of population.

[14] These and the following numbers are Monte-Carlo estimates obtained from six runs with 10 million iterations each. In case of qualified majority voting, the pivot is identified by assuming a status quo $q = 0 \in X$.

[15] Germany's system is actually complex: some members of parliament are directly elected in a first-past-the-post manner, others get seats in proportion to their party's vote.

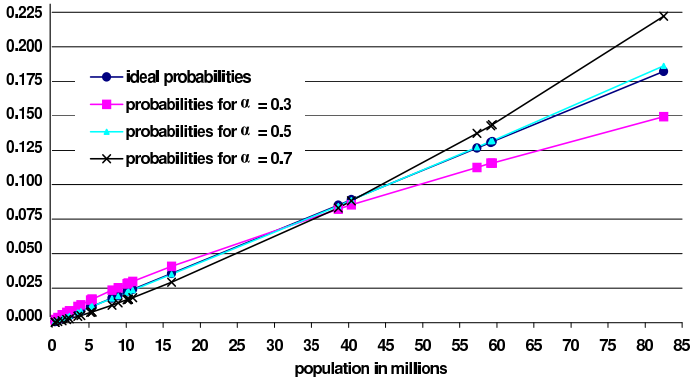


Figure 2: EU25 with weights $w_j = {n_j}^{\alpha}$ compared to ideal probabilities

Investigation of a quota variation even for a very idealized Council illustrates that the decision threshold is not only affecting the balance of 'external costs' and 'decision-making costs' (Buchanan and Tullock 1962) or challenging the so-called 'efficiency' of a decision-making body (operationalized as the probability that a random proposal is passed in the classical 0-1 setting by Felsenthal and Machover 2001 and Baldwin et al. 2001 amongst others). The quota also has important implications for equality of representation and hence the legitimacy of decisions.

3.3 US Electoral College

US citizens elect their president via an Electoral College. The 50 states and Washington DC each send representatives to it. Their number is weakly increasing in the represented share of total population. Although most Electors are not legally bound to vote in any particular way, all state representatives cast their vote for the presidential candidate who secured a plurality of the respective state's popular vote with only minor exceptions. The US Electoral College is therefore commonly treated as a weighted voting system.

Decisions in the Electoral College have in the recent past been essentially binary. The pivotal player amongst the states' median voters might, however, feature prominently in a more sophisticated model of how the two main contestants are selected. In any case, consideration of strategic policy choices in a convex space provide a useful benchmark for the preference-free dichotomous model considered by Penrose (1946) and, specifically addressing the Electoral College, Banzhaf (1968).[16] Figure 3 illustrates the result of determining (hypothetical) weights for state representatives based on current US state population data. Corroborating the findings of Penrose and Banzhaf, the square root rule

[16]Early weighted voting analysis of US presidential elections also includes Brams (1978, ch. 3).

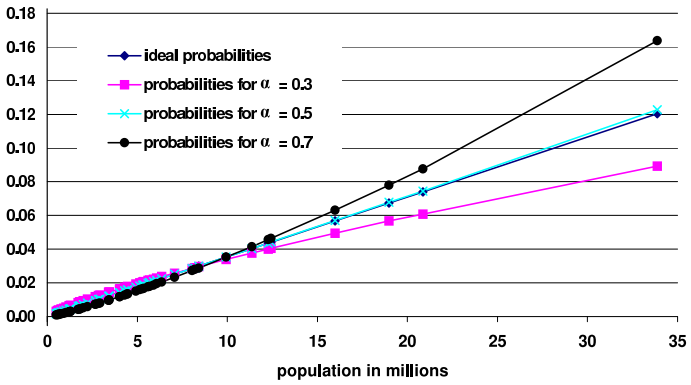


Figure 3: US Electoral College with weights $w_j = {n_j}^{\alpha}$ compared to ideal probabilities

corresponding to $\alpha = 0.5$ is again extremely successful in ensuring equal representation.

4 Concluding remarks

As highlighted, e. g., by Good and Mayer (1975) and Chamberlain and Rothschild (1981), even slight changes regarding decision making at the individual or collective level can produce very different recommendations for operationalizing the one-person, one-vote principle, interpreted here as identical (and positive) indirect expected influence on final outcomes by all voters. Apart from our 'veil of ignorance' perspective with a priori identical but independent voters, the setting considered in this paper is very remote from the preference-free binary model considered by Penrose (1946), Banzhaf (1965, 1968) and others. It is thus surprising that *voting weight proportional to square root of population*, which corresponds to Penrose's original suggestion for most practical purposes,[17] emerges as optimal for both prominent real-world examples as well as many artificial population configurations.

This result matters not only from an abstract point of view. It shows that numerous applied studies have indeed used a robust benchmark. This is also highlighted by recent work of Beisbart and Bovens (2005), which discovers optimality of the square root rule in a very different binary, utility-based egalitarian model. And at least for large constituency populations consisting of many small blocks, Barberà and Jackson (2005) produce similar conclusions in an entirely utilitarian framework. In summary, the square root rule is a simple and trustworthy norm, not an artifact of a particular objective function or setting.

[17] In fact, Penrose (1946) seems to have deliberately blurred the distinction between *voting weight* and *voting power* in his discussion of equal representation in a world assembly. Penrose was aware, however, that approximate proportionality of weight and power generally holds only for sufficiently many constituencies.

This insight can hopefully increase its effect on constitutional design in the real world.[18]

References

Arnold, B. C., N. Balakrishnan, and H. N. Nagaraja (1992). *A First Course in Order Statistics*. New York: John Wiley & Sons.

Baldwin, R. E., E. Berglöf, F. Giavazzi, and M. Widgrén (2001). *Nice Try: Should the Treaty of Nice Be Ratified?* Monitoring European Integration 11. London: Center for Economic Policy Research.

Banks, J. S. and J. Duggan (2000). A bargaining model of social choice. *American Political Science Review 94*(1), 73–88.

Banzhaf, J. F. (1965). Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review 19*(2), 317–343.

Banzhaf, J. F. (1968). One man, 3.312 votes: A mathematical analysis of the Electoral College. *Villanova Law Review 13*, 304–332.

Barberà, S. and M. O. Jackson (2005). On the weights of nations: Assigning voting weights in a heterogeneous union. mimeo, CODE, Universitat Autonoma de Barcelona and California Institute of Technology.

Beisbart, C. and L. Bovens (2005). Why degressive proportionality? An argument from cartel formation. mimeo, University of Dortmund and London School of Economics.

Brams, S. J. (1978). *The Presidential Election Game*. New Haven, CT: Yale University Press.

Buchanan, J. M. and G. Tullock (1962). *The Calculus of Consent*. Ann Arbor, MI: University of Michigan Press.

Chamberlain, G. and M. Rothschild (1981). A note on the probability of casting a decisive vote. *Journal of Economic Theory 25*, 152–162.

Epstein, D. and S. O'Halloran (1999). Measuring the electoral and policy impact of majority-minority voting districts. *American Journal of Political Science 43*(2), 367–395.

Felsenthal, D. and M. Machover (1998). *The Measurement of Voting Power – Theory and Practice, Problems and Paradoxes*. Cheltenham: Edward Elgar.

Felsenthal, D. and M. Machover (2001). The Treaty of Nice and qualified majority voting. *Social Choice and Welfare 18*(3), 431–464.

[18]The square root rule already played a significant role in the public discussion of a possible EU Constitution. See, for example, the open letter by **?**) to the EU members' governments with repercussions in various national news outlets.

Felsenthal, D. and M. Machover (2004). Analysis of QM rules in the draft Constitution for Europe proposed by the European Convention, 2003. *Social Choice and Welfare 23*(1), 1–20.

Gelman, A., J. N. Katz, and J. Bafumi (2004). Standard voting power indexes don't work: An empirical analysis. *British Journal of Political Science 34*(1133), 657–674.

Gelman, A., J. N. Katz, and F. Tuerlinckx (2002). The mathematics and statistics of voting power. *Statistical Science 17*, 420–435.

Good, I. and L. S. Mayer (1975). Estimating the efficacy of a vote. *Behavioral Science 20*, 25–33.

Laruelle, A., R. Martínez, and F. Valenciano (2006). Success versus decisiveness: Conceptual discussion and case study. *Journal of Theoretical Politics (forthcoming)*.

Leech, D. (2002). Designing the voting system for the EU Council of Ministers. *Public Choice 113*(3-4), 437–464.

Leech, D. (2003). Power indices as an aid to institutional design: The generalised apportionment problem. In M. J. Holler, H. Kliemt, D. Schmidtchen, and M. Streit (Eds.), *Yearbook on New Political Economy*, Volume 22. Tübingen: Mohr Siebeck.

Lindner, I. and M. Machover (2004). L. S. Penrose's limit theorem: Proof of some special cases. *Mathematical Social Sciences 47*, 37–49.

Metropolis, N. and S. Ulam (1949). The Monte Carlo method. *Journal of the American Statistical Association 44*, 335–341.

Napel, S. and M. Widgrén (2006). The inter-institutional distribution of power in EU codecision. *Social Choice and Welfare (forthcoming)*.

Penrose, L. S. (1946). The elementary statistics of majority voting. *Journal of the Royal Statistical Society 109*, 53–57.

Shapley, L. S. and M. Shubik (1954). A method for evaluating the distribution of power in a committee system. *American Political Science Review 48*(3), 787–792.

Taylor, A. D. and W. S. Zwicker (1999). *Simple Games*. Princeton, NJ: Princeton University Press.

Nicola Maaser
Department of Economics, University of Hamburg
20146 Hamburg, Germany
Email: `maaser@econ.uni-hamburg.de`

Stefan Napel
Department of Economics, University of Hamburg
20146 Hamburg, Germany
Email: `napel@econ.uni-hamburg.de`

The distributed negotiation of egalitarian resource allocations

Paul-Amaury Matt, Francesca Toni and Dionysis Dionysiou

Abstract

We provide a sound theory for the computation of allocations of indivisible resources amongst cooperative agents, maximising the egalitarian social welfare of the overall multi-agent system, seen as a society. Agents' preferences over resources are captured by scalar utilities that we sum up to define the agents' individual welfare. The egalitarian social welfare is defined as the minimal individual welfare across the society.
From the proposed theory we derive a mechanism of negotiation distributed over the agents. This mechanism is defined by means of a public communication protocol and a private computational policy that have the advantage of integrating efficient coordination and computational heuristics.

1 Introduction

Equity and fairness [17] are social, economic and philosophical notions that can be transposed to artificial societies and serve as a basis for the design of complex agents systems [1].

The problem of reallocating resources amongst agents within a multi-agent systems can be understood as the problem of identifying socially optimal allocations of resources amongst the agents, by interpreting multi-agent systems as societies [8]. In this setting, allocations may be understood as fair if they are *egalitarian* [8], namely if these allocations render the least "well-off" agents in the society as "better-off" as possible, in terms of the individual welfare they obtain from the resources allocated to them.

In this paper, we are concerned with the computation of fair allocations of indivisible resources amongst cooperative agents in a society, where fairness is given this egalitarian interpretation. In particular, we provide a distributed mechanism for computation of egalitarian allocations, whereby agents in a distributed platform share the burden of the computation.

Maximising the egalitarian social welfare by allocating indivisible resources is a hard global optimisation problem, characterised by a discrete domain of exponential size, on which constraints exist and in which a non-linear and non-derivable function is to be optimised. Well known global optimisation and constraint satisfaction techniques (see [15] for a recent survey) cannot be applied.

As Golovin recently put it (see [9] and references therein): 'little is known about the computational aspects of finding [...] fair allocations [...] with indivisible goods' and 'early work in operations research focused on special cases

that are tractable'. The computational aspects of fair allocations of indivisible goods have been studied by [11], but in that work fairness is achieved by minimising envy. Golovin [9] provides approximation algorithms for maximising the egalitarian social welfare, along with some complexity results (see also [2]). In the operations research community [12, 21] resources are allocated to activities instead of agents resulting in a different and simpler problem with fewer variables (linear instead of quadratic).

Endriss et al [8] prove that any sequence of strongly equitable deals (defined therein) will eventually result in an egalitarian allocation of indivisible goods. However, this is a purely theoretical result which provides no indication to agents designers on how to compute these deals and thus the allocations. Also, in the Mathematics community, advanced existence results have been provided [4] concerning fair sharing problems (where additivity of utilities of resources is not assumed). However, also these results do not address the problem of constructing optimal allocations.

In this paper we give a new negotiation mechanism for solving distributedly and without approximation the problem of allocating indivisible resources amongst cooperative agents whose preferences are modeled in terms of semi-linear utility functions. This mechanism is based upon the algorithm described in [13], and is defined in terms of a communication protocol, formalised along the lines of [7], and a communication policy, formalised along the lines of [18].

2 Preliminaries

In this paper, we refer to the agents and resources involved in a resource allocation problem as $a_1, a_2, \ldots, a_n$ and $r_1, r_2, \ldots, r_m$ respectively. The numbers of agents (n) and resources (m) are assumed to be strictly positive. We also assume that the resources are indivisible, so that each resource may be allocated to one agent at most. We will thus use the following definition of allocation of indivisible resources to agents.

Let $E_k = \{a_{i_1}, \ldots, a_{i_k}\}$ represent a group of k agents in the society [1]. An *allocation of resources* to E_k is a Boolean table of k lines and m columns:

$$A^{\{i_1,\ldots,i_k\}} = \begin{pmatrix} i_1: & A_{i_1,1} & A_{i_1,2} & \ldots & A_{i_1,m} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ i_k: & A_{i_k,1} & A_{i_k,2} & \ldots & A_{i_k,m} \end{pmatrix}$$

such that A contains at most one element=1 per column. Given $a_i \in E_k$, we say that a_i *gets* r_j if and only if $A_{i,j} = 1$.

In our framework, agents in a multi-agent systems are abstractly characterised by their own preferences concerning the resources. These preferences are given by means of a *utility table*, defined as a matrix $U = ((U_{i,j}))_{n\times m}$ with n lines and m columns of real valued, positive coefficients. For each $1 \leq i \leq n$ and $1 \leq j \leq m$, $u_{i,j}$ is referred to as the *utility* of resource r_j for agent a_i,

[1] Note that the entire society is given by E_n.

measuring the contribution of the resource to the agent's welfare. Each agent need only be aware of its own preferences, namely its own line in the utility table.

A reasonable and convenient assumption is to consider that the welfare of an agent resulting from an allocation of resources is semi-linearly distributed over the resources, as given by the following definition: for any $1 \leq i \leq n$, the *welfare of agent* a_i *resulting from allocation* A is given by the equation:

$$w_i(A) = c_i + \sum_{j=1}^{m} u_{i,j} A_{i,j}$$

where c_i is a real valued, positive coefficient, representing the welfare of agent a_i prior to any allocation of resources.

Let us now introduce an optimality criterion on allocations, borrowed from the areas of social choice theories [1, 19, 14] and welfare economics [17, 10, 6] and having an *egalitarian* flavour. We are after allocations that maximise the egalitarian social welfare, defined metaphorically as the welfare of the "unhappiest" or least "well-off" agent in the system. Formally, the *egalitarian social welfare* of an allocation A to the entire society E_n is:

$$sw_e(A) = Min\{w_i(A) | i = 1, \ldots, n\}$$

An *egalitarian allocation* is an allocation A^* maximising the egalitarian social welfare.

When building an egalitarian allocation, two problems need to be solved at once: a) finding the value sw_e^* of the optimal egalitarian social welfare and b) actually finding an egalitarian allocation, with social welfare sw_e^*. To solve the first problem, one can perform a dichotomous search. To solve the second problem, the agents will have to reason about sets of allocations, that we encode using *fuzzy allocations*, defined below. A *fuzzy allocation* F *to* E_k is a table with k lines, m columns and whose coefficients $f_{i,j}$ belong to $\{-1, 0, 1\}$:

$$F = \begin{pmatrix} i_1 : & f_{i_1,1} & f_{i_1,2} & \ldots & f_{i_1,m} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ i_k : & f_{i_k,1} & f_{i_k,2} & \ldots & f_{i_k,m} \end{pmatrix}$$

A fuzzy allocation F to E_k encodes the set of allocations to E_k according to which each agent a_i in the group gets r_j if $f_{i,j} = 1$ and does not get r_j if $f_{i,j} = -1$. The coefficients equal to 0 correspond to unspecified information about the allocation of the corresponding resources, and are the reason why fuzzy allocation do not simply denote singletons, but really sets.

We also define the *signature* $s(F)$ *of a fuzzy allocation* F as the allocation in the set encoded by F that allocates fewest resources. This allocation is obtained by replacing in F all the coefficients equal to -1 by 0.

The *social welfare corresponding to a fuzzy allocation* F, denoted $w(F)$, is the egalitarian social welfare of the signature of F, defined over E_k.

3 Computational strategy

In this section we revise the method of [13] that a) uses dichotomous search for finding the value sw_e^* of the optimal egalitarian social welfare and b) uses frugal reductions of allocations and fuzzy allocations for actually finding an egalitarian allocation, with social welfare sw_e^*.

Dichotomy is a simple and elegant mechanism guaranteeing arbitrary precision and enabling fast estimation of the optimal social welfare. In our dichotomous search, lower (L) and upper (U) bounds for this optimal value are updated iteratively. The upper bound corresponds to an allocation where after the allocation the unhappiest agent is given all the resources and the lower bound corresponds to an allocation where after the allocation the unhappiest agent is given no resource. Clearly, the value of the optimal egalitarian social welfare lies somewhere between those bounds. These are initialised as follows:

$$L = Min\{c_i \mid i = 1 \dots n\},\ U = Min\{c_i + \sum_{j=1}^{m} u_{i,j} \mid i = 1 \dots n\}$$

Assuming agents are endowed with an appropriate mechanism for checking the non-emptiness of the set of allocations with social welfare higher than an arbitrary value (the mean of the bounds), dichotomous search algorithm 1 can be used to determine in finite time the exact value of sw_e^*. Our only assumption here is that all agents internally represent their preferences $u_{i,j}$ with d digits of precision. The optimal egalitarian social welfare is rapidly found after

Algorithm 1 Dichotomous search. Inputs: precision d in digits, lower and upper bounds for sw_e^*. Output: optimal social welfare sw_e^*.

1: **repeat**
2: **if** $\{A|sw_e(A) \geq (L+U)/2\} \neq \emptyset$ **then**
3: $L \leftarrow (L+U)/2$
4: **else**
5: $U \leftarrow (L+U)/2$
6: **end if**
7: **until** $U - L < 10^{-d}$
8: **return** round (L+U)/2 with d digits

`floor`$(\log_2 \frac{U-L}{10^{-d}}) + 1$ cycles only.

The check at line 2 of the algorithm is highly complex, as the space of possible allocations is of exponential size $(n+1)^m$. We now discuss how to best handle this check. Basically, our idea is to use a space reduction operator that both eliminates inefficient allocations and redundancies. Indeed, after all, given L and U, all the agents need to do is find out if they can come up with some allocation A such that $sw_e(A) \geq (L+U)/2$.

The operator's definition is based on a special binary relation between pairs of allocations. Let A and B be two allocations to E_k. We say that B *minors* A

$$F(\{\begin{pmatrix}1&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}0&1&0&0\\0&0&1&0\\1&0&0&0\end{pmatrix}\begin{pmatrix}0&0&1&0\\0&1&0&0\\1&0&0&0\end{pmatrix}\})$$
$$=\{\begin{pmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}0&1&0&0\\0&0&1&0\\1&0&0&0\end{pmatrix}\}$$

Figure 1: The frugal reduction operator F eliminates both redundancies (superfluous agreements) and inefficient allocations (over-consuming resources). The agents save memory and computational time and the society manages its resources better (here either resource r_1 or r_4 can be preserved).

and write $B \preceq A$ if and only if

$$\forall j \in \{1, ..., m\} : \sum_{i \in E_k} B_{i,j} \leq \sum_{i \in E_k} A_{i,j}$$

The intuitive meaning of $B \preceq A$ is that whatever resource is allocated according to B, it is also allocated according to A. When considering sets of allocations for all of which $w(A) \geq (L + U)/2$ holds, the agents may perfectly treat non-minimal allocations as superfluous. Also, when two such allocations minor each other, one can be eliminated to avoid redundancy. This defines our reduction operator. Let S be a set of allocations for E_k. A *frugal reduction* $F(S)$ of S is a subset of S such that

- any allocation in S is minored by an allocation of $F(S)$
- no two allocations in $F(S)$ minor each other.

Note that frugal reductions are not guaranteed to be unique, but the frugal reduction operator has a remarkable property: $S \neq \emptyset \Leftrightarrow F(S) \neq \emptyset$.

Intuitively, we can forsee that $F(S)$ is statistically much smaller than S itself (cf figure 1), so in a way, using frugal reductions simplifies the search process for allocations. Moreover, frugal reductions can be computed using an incremental negotiation mechanism summarised in algorithm 2. At each step k, one new agent joins a group E_k, forcing a revision of the set of agreements amongst these prior agents. The newly formed group then eliminates superfluous agreements using a frugal reduction or abandons the search step when no agreements can be found (fail).

In order to build the minimal collection of agreements for a group to which a new agent has just been added, we consider a forest (set of trees). In each phase, specific leaves (termed *positive leaves*) of the trees in a forest constitute a collection (not yet minimal) of agreements for the group. The roots of the trees constituting the forest of a phase are simply the signatures of the positive leaves of the forest in the previous phase.

Algorithm 2 Incremental construction of a frugal reduction $F_n(x)$ of $\{A|sw_e(A) \geq x\}$. Input: x. Output: $F_n(x)$.

$E_0 \leftarrow \{\}; k \leftarrow 0; F_0(x) \leftarrow \{\}$
repeat
  $k \leftarrow k + 1$
  $E_k \leftarrow E_{k-1} \cup \{k\}$
  a_k tries to find a consensus Ext_k with the prior agents (E_{k-1}) of welfare at least equal to x
  if $Ext_k \neq \emptyset$ (i.e. a consensus can be found) **then**
    $F_k(x) = F(Ext_k)$ (reduce the set frugally)
  else
    return $\emptyset$ (failure)
  end if
until $k = n$
return $F_n(x)$

Suppose an agent $a_{i'}$ wants to join a group $G = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$ to form the group $G' = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}, a_{i'}\}$. The group G then starts constructing a new forest whose trees' nodes N are pairs of the form $(F, w(F))$ where F is a fuzzy allocation for G'.

The root of any tree in the constructed forest at iteration $k+1$ is a pair $(F, w(F))$ where the first k lines of F take their values in $Ag(G)$ (the minimal collection of agreements for G), where G is the group of agents at iteration k (consisting of k agents), and all the coefficients in the last line (corresponding to the newly added agent $a_{i'}$) are equal to zero. The trees are constructed top-down from their root and all have a strictly binary structure.

A node $(F, w(F))$ in a tree is called

- *positive* iff F is satisfying, i.e. $w(F) \geq (L + U)/2$
- *open* iff it is not positive but the allocation in the set encoded by F in which all the resources not used by an agent in G are used by the new agent $a_{i'}$ is satisfying
- *negative* iff it is neither positive nor open.

Negative and positive nodes have no children, only open nodes do.

Consider an open node $N = (F, w(F))$. Let j_0 be the index of a resource r_j that $a_{i'}$ could use, i.e. $f_{i',j_0} = 0$. Such an index exists since the node is open. Then the left and right children of N, denoted $(F_L, w(F_L))$ and $(F_R, w(F_R))$, are defined as follows:

$$\begin{cases} f_{L;i',j_0} = 1, \\ f_{R;i',j_0} = -1, \\ \forall j \neq j_0 : \ f_{L;i',j} = f_{R;i',j} = f_{i',j} \end{cases}$$

The agents build the tree by constructing the descendants of all the open nodes cf figure 2. The process terminates finitely because their is a finite number of resources. In fact, the depth of a tree is equal to the number of resources $a_{i'}$ can use.

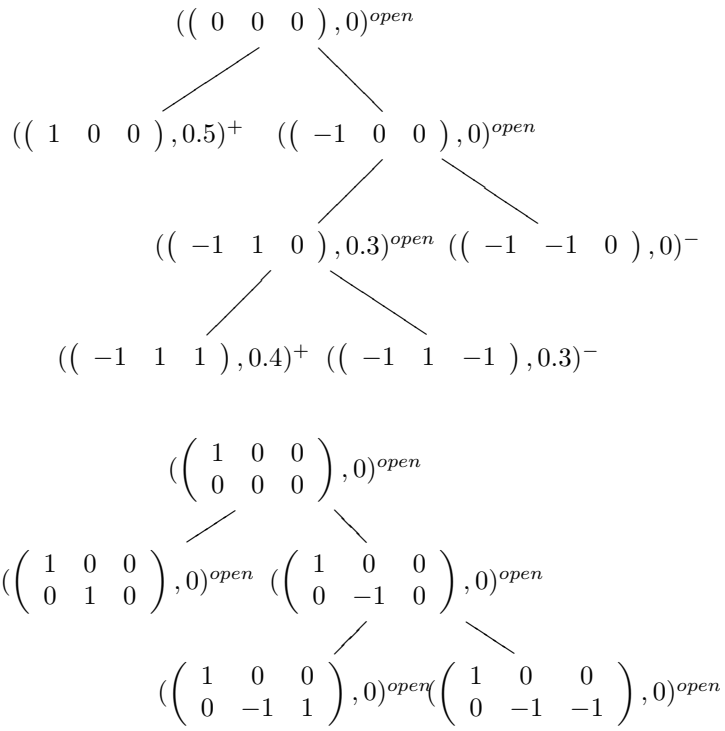


Figure 2: Agent a_1 finds two agreements (top tree). From the first one, agent a_2 derives (bottom tree) two possible agreements that satisfy them both. The agent pair finds a consensus in which a_1 gets r_1 and a_2 either r_2 or r_3.

The trees thus constructed have the interesting property that the frugal reduction of the set of satisfying sub-allocations in a fuzzy sub-allocation F is included in the set of signatures of the frugal tree whose root is F.

Applying the frugal reduction operator after having collected a tree's signatures is advocated as it enables the agents to ignore any superfluous agreements. The reason why we do not loose any useful agreement by working only on the positive nodes signatures is slightly technical and justified by the following result, where the role of the signatures set is played by A and the satisfying set of allocation in the root is played by Σ:

if $F(\Sigma) \subseteq A \subseteq \Sigma$ then $F(A) = F(\Sigma)$ (namely, a frugal reduction of A is also one of Σ).

The order in which the agents join in the group is an order that coordinates

group negotiations. We have noticed in [13] that *social orders*, i.e. orders derived from welfare metrics, can have a strong (positive) impact on the time complexity of the negotiations. In particular, it is important to order the agents in increasing level of initial welfare. When the unhappiest agents think first about the resources they need, the detection of impossibility to find a common consensus is made earlier thus saving negotiation time. Also, since unhappiest agents tend to consume more resources than the others, they leave the others with a more restricted choice, which simplifies their reasoning task. We refer to this heuristic as LW.

When an agent constructs a search tree, it is important to minimise the depth of the tree. A good heuristic for that consists in splitting open nodes by thinking about the most useful resource that remains available to the agent. Indeed, this increases the probability that the left sub-tree is simply a positive leaf. When this heuristic (that we refer to as MU) and the earlier LW heuristic are combined, the total negotiation time is reduced by a factor almost equal to 30 (cf figure 3) in comparison with negotiations where no heuristics are applied and the agents negotiate in a random order, do not prioritise resources, and have initial welfare and preferences uniformly distributed in the interval $[0, 1]$. A precise description of the settings used for the corresponding experiments can be found in [13].

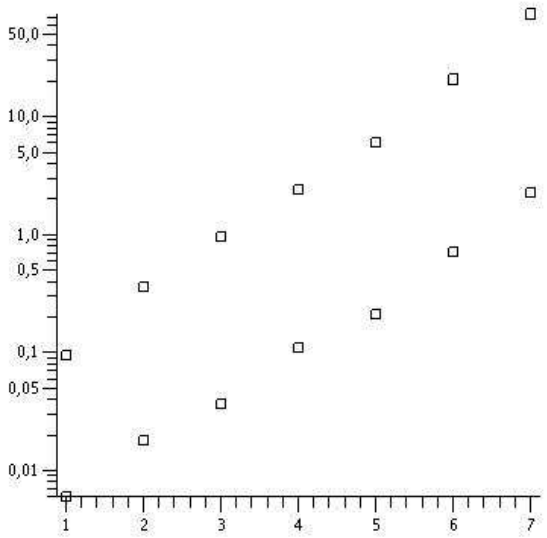


Figure 3: Negotiation time (in seconds) using the combined LW-MU heuristic (bottom) compared to a random strategy (up). The negotiation speed is multiplied by 30 when using this heuristic.

4 Protocol and Policy

The resource allocation problem can be solved distributedly by means of negotiation amongst the agents. The description of this process can be given by defining a public communication protocol, agreed by all agents, and private computational policies, held by the individual agents. The protocol defines what agents are allowed to say and how they should react (by means of their internal policy) to messages they receive. Giving a protocol is a necessary requirement for the definition of a suitable semantics of an agent communication language [16, 20]. In order to render the negotiation mechanism unambiguous, each policy needs to conform to the protocol.

We present here a public communication protocol derived from algorithms 1 and 2, and encapsulating our efficient methods for reasoning about agreements between groups of agents over resources allocations. The protocol is presented in the form of a deterministic finite state automaton (DFA) (see figure 4), in the flavour of [7]. The DFA consists of $n+1$ states: one state a_k per agent and a final state f. Each of the states a_k is characterised by the values of three variables: current lower bound L and upper bound U of the optimal egalitarian social welfare and the set $F_k(x)$ (for $x = (L+U)/2$) of agreements for the current group $E_k = \{a_1, ..., a_k\}$. The initial state is a_1 with variables assigned to the values $L_0, U_0, \emptyset$. The empty set means that the first agent does not need to take into account agreements reached by the other agents. The syntax [3] of our negotiation protocol is given by the language [18] $\mathcal{L}$ consisting of instances of the *tell* predicate which has the following five arguments: sender X, receiver Y, message M, lower bound L and upper bound U of the optimal egalitarian social welfare. A message M may take three forms, i) a non-empty set A of agreements, ii) failure due to the absence of possible consensus, iii) success when a consensus can be found, and iv) solution for publishing an egalitarian allocation A^*, solution of the problem. The language is then defined as

$$\mathcal{L} = \{\texttt{tell}(X, Y, M, L, U) | X \in S, Y \in S \texttt{ or } Y = S, (L, U) \in \mathbb{R}^2, 0 \leq L \leq U\},$$

where $M \in \{\texttt{agreements}(A), \texttt{failure}, \texttt{success}, \texttt{solution}(A^*)\}$ and S stands for the socially ordered variant of the agent system $\{a_1, a_2, \ldots a_n\}$ such that $c_1 \leq c_2 \leq ... \leq c_n$.

The DFA's transition function maps pairs of states and elements of the input alphabet to states. In the context of communication protocols, elements of the input alphabet are dialogue moves and states are the possible stages of the interaction. The transition function consequently gives a clear semantics [3] to our protocol. We introduce the *next* function that transforms a_i into a_{i+1} for $i < n$ and a_n into a_1 so as to enable looping in the negotiations.

The transition function δ is then defined as the union of the following rules where i ranges from 1 to n:

- $\delta(a_i^{L,U,F_i}, \texttt{tell}(a_i, a_{next(i)}, \texttt{agreements}(F_i), L, U)) = a_{next(i)}^{L,U,F_{next(i)}}$
- $\delta(a_i^{L,U,\emptyset}, \texttt{tell}(a_i, a_1, \texttt{failure}, L, \frac{L+U}{2})) = a_1^{L,\frac{L+U}{2},F_1}$

- $\delta(a_n^{L,U,F_n}, \texttt{tell}(a_n, a_1, \texttt{success}, \frac{L+U}{2}, U)) = a_1^{\frac{L+U}{2},U,F_1}$
- $\delta(a_n^{L,U,F_n}, \texttt{tell}(a_n, S, \texttt{solution}(A^*), \texttt{Round}(\frac{L+U}{2}, d), \texttt{Round}(\frac{L+U}{2}, d)) = f$

In each state a_k, agent a_k has to revise the set of agreements found by the prior agents $\{a_1, \ldots a_{k-1}\}$ and communicated by a_{k-1}. The graph loops back to the first agent either when no consensus can be found or when all the agents have found a consensus and in both cases the lower or upper bounds are updated accordingly to the dichotomous update, pessimistically in the first case and optimistically in the second one. The last agent a_n is responsible for detecting the final dichotomous step and does so by checking if $U - L < 10^{-d}$ holds. If it is the case, it chooses arbitrarily an egalitarian allocation and sends it to them. The negotiation stops and the agents can go and pick up their resources accordingly to the solution.

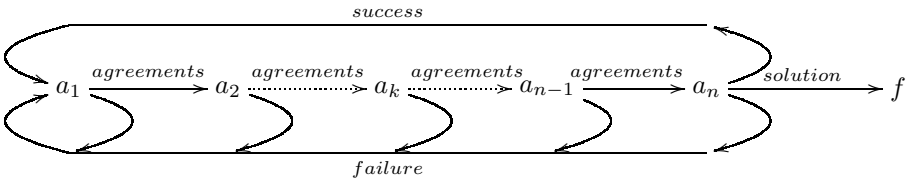


Figure 4: Public communication protocol.

A policy that conforms to the protocol and encapsulates the computational techniques is now given. Following [18], policies are expressed as dialogue constraints of the form $p_i \wedge C \Rightarrow p_{i+1}$, where p_i and p_{i+1} are dialogue moves. The dialogue constraints are constructed so as to associate unambiguously to each agent and message received a (unique) dialogue move [2]. Those policies give a pragmatics [3] that is easy to implement and execute in a distributed architecture.

- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge (F_Y(A) = \emptyset) \Rightarrow \texttt{tell}(Y, a_1, \texttt{failure}, L, (L + U)/2)$
- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge \neg((Y = a_n) \wedge ((U - L) < 10^{-d})) \wedge (F_Y(A) \neq \emptyset) \Rightarrow \texttt{tell}(Y, \texttt{next}(Y), \texttt{agreements}(F_Y(A)), L, U)$
- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge (Y = a_n) \wedge ((U - L) \geq 10^{-d})) \wedge (F_Y(A) \neq \emptyset) \Rightarrow \texttt{tell}(Y, a_1, \texttt{success}, (L + U)/2, U)$
- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge (Y = a_n) \wedge ((U - L) < 10^{-d}) \wedge (F_Y(A) \neq \emptyset) \Rightarrow \texttt{tell}(Y, S, \texttt{solution}(\texttt{OneOf}(F_Y(A))), \texttt{Round}((L + U)/2, d), \texttt{Round}((L + U)/2, d)$
- $\texttt{tell}(X, Y, \texttt{failure}, L, U) \wedge (F_Y(\emptyset) \neq \emptyset) \Rightarrow \texttt{tell}(Y, \texttt{next}(Y), \texttt{agreements}(F_1()), L, U)$
- $\texttt{tell}(X, Y, \texttt{failure}, L, U) \wedge (F_1(\emptyset) = \emptyset) \Rightarrow \texttt{tell}(Y, Y, \texttt{failure}, L, (L + U)/2)$

[2] All variables in the given dialogue constraints are implicitly universally quantified from the outside.

- $\texttt{tell}(X, Y, \texttt{solution}(A^*), sw_e^*, sw_e^*) \Rightarrow$ COLLECT RESOURCES

We assume that each agent is equipped with this policy. By definition, a dialogue move p is legal with respect to a state s if and only if there exists a state s' such that $\delta(s, p) = s'$. In order to make sure that the policy conforms to the protocol and is well formed, the reader can check that:

- for any (legal) message Msg received by an agent Y, the agent can compute a unique state (L, U, F) (determined by the protocol) and that state satisfies the constraints of a unique policy rule amongst those whose p_i match Msg (policy rules exhaustivity and independence). Consequently:
- i) agents never utter any illegal move (weak protocol conformance)
- ii) agents utter at least one legal output move for any legal input they receive (exhaustive protocol conformance)

5 Conclusion

We presented a sound method that guarantees agents to find an allocation that exactly maximises the egalitarian social welfare of the society they constitute. The method relies upon a dichotomous search terminating after a small number of steps. In the search process, agents examine and update the value of the optimal egalitarian social welfare that can be collectively achieved given their personal preferences, which can be kept secret. Our method uses binary search trees and forests of Boolean fuzzy allocations as well as a frugal reduction operator that simplifies the reasoning process of the agents by eliminating opportunistically any superfluous agreements they might come up with. The solutions are efficient as far as they never over-consume resources. We proved empirically that the agents reason collectively much faster when thinking in priority about the most useful resources and could efficiently coordinate the sequence of their negotiations by using the monotonic increasing social order. Finally, the negotiation mechanism has been distributed over the agents engaged in the allocation process using a protocol and a policy conforming to it which implements the dichotomous search and encapsulates the efficient consensus search algorithm here-presented. The overall mechanism has been implemented on a JADE platfrom [5]. Part of our future work will be dedicated to a theoretical and experimental study of the frugal reduction's efficiency. We will also propose other ways of modelling an agent's preferences that will enable to solve the allocation problem in polynomial time and show how the mechanism can be used for negotiating the allocation of markets supervised by fair trade organisations. Finally, we would like to make the mechanism strategy-proof.

Acknowledgements

This work was partially funded by the Sixth Framework IST programme of the EC, under the 035200 ARGUGRID project. The second author has also been supported by a UK Royal Academy of Engineering/Leverhulme Trust senior fellowship.

References

[1] Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, New York, 2 edition, 1963.

[2] Sylvain Bouveret, Michel Lemaître, Hélène Fargier, and Jérôme Lang. Allocation of indivisible goods: a general model and some complexity results. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 1309–1310. ACM, 2005.

[3] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, 2004.

[4] Marco Dall'Aglio and Fabio Maccheroni. Fair division without additivity. *AMM: The American Mathematical Monthly*, 112, 2005.

[5] Dionysis Dionysiou. *Egalitarian Resource Allocations in Multi-Agent Systems*. MSc Thesis, Imperial College London, 2006.

[6] Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Resource allocation in egalitarian agent societies, 2003.

[7] Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Logic-based agent communication protocols. 2004.

[8] Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315–348, 2006.

[9] Daniel Golovin. Max-min fair allocation of indivisible goods. *Technical Report CMU-CS-05-144, Carnegie Mellon University*, 2005.

[10] John C. Harsanyi. Can the maximin principle serve as a basis for morality? *American Political Science Review*, 86(2):269–357, 1996.

[11] Richard Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *CECOMM: ACM Conference on Electronic Commerce*, 2004.

[12] Hanan Luss. On equitable resource allocation problems: A lexicographic minimax approach. *Operations Research*, 47(3):361–378, 1999.

[13] Paul-Amaury Matt and Francesca Toni. Egalitarian allocations of indivisible resources: Theory and computation. In *Cooperative Information Agents (CIA'06)*. Lecture Notes in Computer Science, Springer Verlag, 2006.

[14] Hervé Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[15] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13:271–369, 2004.

[16] Jeremy Pitt and Abe Mamdani. A protocol-based semantics for an agent communication language. In Dean Thomas, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol1)*, pages 486–491, S.F., July 31–August 6 1999. Morgan Kaufmann Publishers.

[17] John Rawls. *A Theory of Justice*. Harvard University Press, 1971.

[18] Fariba Sadri, Francesca Toni, and Paolo Torroni. An abductive logic programming architecture for negotiating agents. *Lecture Notes in Computer Science*, 2424, 2002.

[19] Amartya K. Sen. *Collective Choice and Social Welfare*. Holden Day, 1970.

[20] Munindar P. Singh. Agent communication languages: Rethinking the principles. *IEEE Computer*, 31(12):40–47, 1998.

[21] Gang Yu. On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44:407–415, 1996.

Paul-Amaury Matt, Francesca Toni and Dionysis Dionysiou
Department of Computing, Imperial College London,
South Kensington Campus, Huxley Building,
London SW7 2AZ, United Kingdom
Email: {pmatt,ft,dd205}@doc.ic.ac.uk

Anonymous Voting and Minimal Manipulability

Stefan Maus, Hans Peters and Ton Storcken

Abstract

We compare the manipulability of different choice rules by considering the number of manipulable profiles. We establish the minimal number of such profiles for tops-only, anonymous, and surjective choice rules, and show that this number is attained by unanimity rules with status quo.

Keywords: Anonymity, voting, minimal manipulability

JEL Classification Numbers: D72

1 Introduction

In choosing new parliamentary representatives most democracies apply voting procedures that select among the top-ranked candidates reported by the voters. It is well known that such procedures are vulnerable to manipulation. For example, if there is an electoral threshold, then votes for a small party might be reconsidered and cast on a (second best) larger party which with high probability will meet the threshold. Also in a district dependent procedure, a voter might opt for the second best if his best candidate has only small support, and in that way prevent a third (worse) candidate to be elected as district representative. In this paper we study voting procedures with respect to this kind of manipulability. Using a natural measure of manipulation, we show that unanimity rules with status quo are the least vulnerable among all reasonable procedures.

We consider a framework in which voting procedures are modelled as choice rules assigning alternatives (from a set of at least three alternatives) to profiles of individual preferences. These choice rules are assumed to be tops-only, meaning that they only depend on the top-ranked alternatives of the voters. Additionally, two standard and natural conditions are imposed: anonymity and unanimity. Anonymity is an egalitarian principle, saying that the names of the voters do not matter. Unanimity is a minimal sovereignty principle: it means that if all voters have the same top candidate, then this candidate is elected. To this setting, however, the well-known result of Gibbard [7] and Satterthwaite [21] applies, and therefore any choice rule satisfying the three mentioned conditions is vulnerable to manipulation. This means that, for any such rule, there exist a profile and a voter who, by changing his preference, can induce a new profile resulting in an outcome which is better for him. This kind of manipulation may be undesirable for several reasons. First, the manipulating voter may benefit on the expense of others. Second, in order to obtain a good

outcome, the right input should be given to the voting mechanism. Finally, the impossibility of manipulation simplifies the decision process for the voters because they only have to know their own preferences.

There are several strands of research dealing with this manipulability issue. One concerns relaxations of the conditions on the rules at hand. Often, stronger or similar impossibility results are found. See e.g. Pattanaik [19], [20] and Ehlers *et al.* [5]. A second strand of literature is based on a stochastic approach, again often resulting in similar impossibilities. See, e.g., Gibbard [8], [9] and Dutta [4]. A third strand imposes preference domain restrictions, often to single-peaked preferences. If the space of alternatives is one-dimensional, preferences are single-peaked, and the number of voters is odd, then a Condorcet winner exists, which is then a non-manipulable choice. See, e.g., Black [2] or Moulin [18]. If the space of alternatives is more dimensional, then a Condorcet winner usually fails to exist. Depending on the domain of admissible preferences, non-manipulable choice rules may or may not exist. See, e.g., Kim and Roush [13], Border and Jordan [3], and Zhou [22]. (Of course, the given references are far from constituting a complete list.)

In this paper, we take a different approach. Since all choice rules are manipulable, a natural question is which choice rules are performing best in this respect, i.e., are the least manipulable. To answer this question we need a measure of manipulability. An intuitive measure is to count the number of profiles at which a given choice rule is manipulable: the larger this number the more manipulable the choice rule is. This measure was introduced by Kelly [10]. He found the minimal number of manipulable profiles for choice rules which are unanimous and non-dictatorial in the case of two agents[1] and three alternatives. See also Kelly [11], [12]. In Fristrup and Keiding [6] this minimal number was found for an arbitrary number of alternatives and two agents. Maus *et al.* [14] obtain a general result for arbitrary numbers of agents and alternatives: almost dictatorial rules are the least vulnerable to manipulation among all non-dictatorial and unanimous rules. In Maus *et al.* [15] the minimal degree of manipulation for surjective and anonymous choice rules is determined. In Maus *et al.* [16] this degree is found for unanimous and anonymous choice rules for the case of three alternatives and an arbitrary number of agents. By enumeration and simulation techniques, Aleskerov and Kurbanov [1] determine the minimal number of manipulable profiles for twenty six well-known choice rules such as Borda and plurality. They also discuss other measures for manipulation.

The present paper is different since we confine ourselves to tops-only choice rules—often called voting rules. We show that among all unanimous and anonymous voting rules, the unanimity rule with status quo is doing best with respect to manipulability. This rule chooses a given fixed alternative (the status quo) unless all voters have the same best alternative, possibly different from the status quo. We derive this result under the assumption that the number of agents exceeds the number of alternatives. The fraction of manipulable profiles for

[1] In general we use the term "agent" rather than "voter".

this rule turns out to be of order $n \cdot m^{2-n}$, where n is the number of agents and m the number of alternatives. So this rule is among the few choice rules which are not highly manipulable in the terminology of Kelly [12]. Clearly, this choice rule is only occasionally used, for instance in the Council of the European Union, and its rigidity makes it hardly applicable in elections. Therefore, the result presented here is an exploring step setting an absolute lower bound on the measure of manipulation in voting rules, rather than a recommendation to use unanimity with status quo rules. Moreover, we do not know if it holds true if manipulation is measured differently (see Aleskerov and Kurbanov [1]). On the other hand, this lower bound makes it possible to compare the level of manipulation of a given rule to what is achievable in this respect.

Our proof of this result is based on combinatorial arguments which have no bite if the number of agents does not exceed the number of alternatives. For the latter case some partial results can be found in [17] and in the final section of this paper. It turns out that for two agents unanimity rules with status quo are not necessarily minimally manipulable. Also for two agents, we obtain a characterization of all minimally manipulable rules under the stronger condition of Pareto optimality instead of unanimity.

The paper is organized as follows. Section 2 contains preliminaries and introduces unanimity rules with status quo. Section 3 presents the main result, and Section 4 concludes.

2 Unanimity rules with status quo

Throughout we consider a finite set A of m alternatives and a set $N = \{1, 2, ..., n\}$ of agents. Unless stated otherwise we assume $n > 2$. The agents have linear preferences over the alternatives, i.e. (strongly) complete, antisymmetric and transitive relations on A. Let $L(A)$ denote the set of all these preferences. A choice rule is a function f from $L(A)^N$ to A, where $L(A)^N$ denotes the set of profiles p of linear orderings. At a profile p the preference of agent $i \in N$ is denoted by $p(i)$. Let a, b and c be three alternatives in A. Then $...a...b... = p(i)$ means that a is preferred to b at $p(i)$ and $c... = p(i)$ means that c is best at $p(i)$; in that case we also write $\text{top}(p(i)) = c$. For a profile p in $L(A)^N$ the function top(p) in A^N is defined by $\text{top}(p)(i) = \text{top}(p(i))$ for all agents $i \in N$. Also, $\text{topset}(p) = \{\text{top}(p(i)) : i \in N\}$ is the set of alternatives that are at least once at the top of an agent's preference in p. For a profile $p \in L(A)^N$ and an alternative a in A let $N(a, p) = \{i \in N : \text{top}(p(i)) = a\}$ and $n(a, p) = |N(a, p)|$, where $|S|$ denotes the cardinality of the set S.

A choice rule f is called *anonymous* if it is symmetric in its arguments. It is called *surjective* if (as usual) $f(L(A)^N) = A$. Here, for all $V \subseteq L(A)^N$, the image of V under f is denoted by $f(V)$. A slightly stronger condition than surjectivity is *unanimity*: this means that for profiles p, if $\text{topset}(p) = \{a\}$ for some alternative a, then $f(p) = a$. So, if all agents order alternative a best, then it is chosen. A choice rule f is called *tops-only* if $f(p) = f(q)$ for all profiles

p and q with $\text{top}(p) = \text{top}(q)$. So the outcome of a tops-only choice rule at a profile is completely determined by the best alternatives of the agents: such a rule is usually called a voting rule.

For an agent $i \in N$, profiles p and q are i-deviations if $p(j) = q(j)$ for all $j \neq i$. A choice rule f is *manipulable at profile p by agent i via profile q* if p and q are i-deviations, $f(p) \neq f(q)$ and $...f(q)...f(p)... = p(i)$. In such a case agent i can benefit at profile p by reporting $q(i)$ in stead of $p(i)$. Let M_f denote the set of all profiles at which choice rule f is manipulable. Then $|M_f|$ measures the manipulability of choice rule f. If $|M_f|$ is equal to zero, then at every profile the choice rule is not manipulable, in which case it is said to be *strategy-proof*. If there are at least three alternatives, then only dictatorial rules are strategy-proof and surjective: this is the well-known result of Gibbard [7] and Satterthwaite [21]. Let F denote the class of all anonymous, surjective and tops-only choice rules. Then the Gibbard-Satterthwaite result implies $\min\{|M_f| : f \in F\} > 0$ since dictatorial rules are tops-only and surjective but not anonymous.

For an alternative a we define the *unanimity rule with status quo a*, denoted by u_a, as follows. Let p be a profile. Then $u_a(p) := x$ if $\{x\} = \text{topset}(p)$ for some $x \in A$, and $u_a(p) = a$ in all other cases. So an alternative x different from a is chosen only if all agents consider it best. The main result of this paper is that unanimity rules with status quo are the minimally manipulable rules among all anonymous, surjective, and tops-only rules, provided $n > m \geq 3$. The number of manipulable profiles $|M_{u_a}|$ can be computed as follows. Consider a profile $p \in M_{u_a}$. Then for some agent i and some i-deviation q, $u_a(p) \neq u_a(q)$ and $...u_a(q)...u_a(p)... = p(i)$. Clearly $u_a(p) = a$ and $u_a(q) \neq a$. So, $u_a(q)... = p(j)$ for all agents $j \in N \setminus \{i\}$. As $u_a(p) \neq u_a(q)$ it follows that $\text{top}(p(i)) \neq u_a(q)$. Since there are $\frac{m!}{2}$ preferences $p(i)$ with $u_a(q)$ ranked above a but $(m-1)!$ of these have $u_a(q)$ on top, it follows that there are $\frac{m!}{2} - (m-1)!$ preferences $p(i)$ which result in a manipulable profile. Since we can choose i from a set of n agents, $u_a(q) \neq a$ from $m-1$ alternatives, and the other alternatives can be ordered by the other agents in $((m-1)!)^{n-1}$ ways, we find altogether that

$$\begin{aligned} |M_{u_a}| &= n \cdot (m-1) \cdot (\frac{m!}{2} - (m-1)!) \cdot ((m-1)!)^{(n-1)} \\ &= \frac{1}{2} n(m-1)(m-2)((m-1)!)^n. \end{aligned} \tag{1}$$

We end this section with a combinatorial observation which is used extensively in the following two sections.

Remark 1 Let $m \geqslant 3$ and let p be a profile with $\text{topset}(p) = \{x_1, x_2, ..., x_k\}$. Let the anonymous and tops-only choice rule f be manipulable at profile p by agent $i \in N(x_1, p)$ via profile q. Then, obviously, $f(p) \neq x_1$. There are

$$\frac{n!}{n(x_1,p)! \cdot n(x_2,p)! \cdot \ldots \cdot n(x_k,p)!}((m-1)!)^n \tag{2}$$

profiles r which by anonymity and tops-onliness yield the same outcome as p under f. As $f(p) \neq x_1$, at most

$$\frac{n!}{n(x_1,p)! \cdot n(x_2,p)! \cdot \ldots \cdot n(x_k,p)!}((m-1)!)^{n-n(x_1,p)} \cdot \left(\frac{(m-1)!}{2}\right)^{n(x_1,p)} \quad (3)$$

of these profiles are such that all agents in $N(x_1,p)$ prefer $f(p)$ to $f(q)$, and therefore are not manipulable by such an agent at p via q. Subtracting (3) from (2), we obtain

$$|M_f| \geqslant \frac{n!}{n(x_1,p)! \cdot n(x_2,p)! \cdot \ldots \cdot n(x_k,p)!}((m-1)!)^n(1 - \left(\frac{1}{2}\right)^{n(x_1,p)}). \quad (4)$$

3 Minimal manipulation with three or more alternatives

In this section, we prove the following theorem, which is the main result of this paper.

Theorem 1 *Let $n > m \geqslant 3$. Let $f \in F$. Then $|M_f| \leqslant |M_g|$ for all $g \in F$ if and only if f is a unanimity rule with status quo.*

So the theorem says that among all surjective, anonymous and tops-only choice rules only unanimity rules with status quo are minimally manipulable, provided that $n > m \geqslant 3$. In the concluding Section 4 we briefly discuss the case of two agents.

Let $f \in F$ such that $|M_f| \leqslant |M_{u_a}|$. For $1 \leqslant k \leqslant m$ let $B_k = \{p \in L(A)^N : |\text{topset}(p)| \geqslant k\}$. So B_k is the set of profiles at which there are at least k different top alternatives. The proof of Theorem 1 is based on a series of lemmas about f. The first lemma says that non-manipulability of f on profiles with at least three top elements implies that f is constant on such profiles.

Lemma 1 *Let $n > m \geqslant 3$ and let $k \geqslant 3$. Suppose $B_k \cap M_f = \emptyset$. Then there is an alternative a such that $f(B_k) = \{a\}$.*

Proof. Let $p, q \in B_k$ and $i \in N$ such that p and q are i-deviations. It is sufficient to prove that $f(p) = f(q)$. To the contrary assume that $f(p) = a \neq b = f(q)$. As neither p nor q are in M_f it follows that $...f(p)...f(q)... = p(i)$ and $...f(q)...f(p)... = q(i)$.

Suppose $\text{top}(p(i)) = c \neq f(p)$. Then for an i-deviation r of p such that $r(i) = c...f(q)...f(p)...$ we would have, by tops-onliness: $f(r) = f(p)$, hence i could manipulate at r via q. Since $r \in B_k$, this contradicts $B_k \cap M_f = \emptyset$. Hence $\text{top}(p(i)) = f(p) = a$. Similarly it follows that $\text{top}(q(i)) = f(q) = b$. So, $n(a,p) = n(a,q) + 1$ and $n(b,p) + 1 = n(b,q)$. Since p and q are i-deviations in B_k and $k \geqslant 3$, there is an alternative $c \in A \setminus \{a,b\}$ and an individual

$j \in N(c,p) \cap N(c,q)$. Consider profiles v and w such that v is a j-deviation of p with $b... = v(j)$ and w satisfies $v(i) = w(j)$, $v(j) = w(i)$, and $v(l) = w(l)$ for all $l \neq i,j$. Note that q and w are j-deviations. Suppose $f(v) \neq a$. Then by tops-onliness we may assume without loss of generality that $...f(v)...a... = p(j)$. But then f is manipulable at p by j via v, a contradiction since $p \in B_k$ and therefore $p \notin M_f$. So $f(v) = a$. Then, by anonymity, $f(w) = a$. Because of tops-onliness we may assume without loss of generality that $...a...b... = q(j)$. This makes f manipulable at q by j via w, which yields a contradiction since $q \in B_k$ and therefore $q \notin M_f$. Hence, $f(p) = f(q)$. ■

In the next three lemmas we assume $n > m \geqslant 4$. We first show that B_4 is disjoint from M_f; then that f is constant on B_3; finally that f is constant on B_2.

Lemma 2 *Let $n > m \geqslant 4$. Then $B_4 \cap M_f = \emptyset$.*

Proof. Let $p \in B_4$ and suppose that f is manipulable at p. Let $\text{topset}(p) = \{x_1, x_2, \ldots, x_k\}$, where $k \geqslant 4$. By (4) there is an alternative x_1 such that

$$|M_f| \geqslant \frac{n!}{n(x_1,p)! \cdot n(x_2,p)! \cdot \ldots \cdot n(x_k,p)!}((m-1)!)^n(1 - \left(\frac{1}{2}\right)^{n(x_1,p)}) \,. \quad (5)$$

Note that for arbitrary natural numbers c and d we have $c!\,d! \leqslant (c+d-1)!$. Repeated application of this inequality yields

$$\begin{aligned} n(x_1,p)! \cdot n(x_2,p)! \cdot \ldots \cdot n(x_k,p)! &\leqslant \left(\sum_{j=1}^{k} n(x_j,p) - (k-1)\right)! \\ &= (n-(k-1))! \\ &\leqslant (n-3)! \end{aligned}$$

Here, the last inequality follows since $k \geqslant 4$. Observing moreover that $1 - \left(\frac{1}{2}\right)^{n(x_1,p)} \geqslant \frac{1}{2}$, (5) implies

$$|M_f| \geqslant \frac{n!}{(n-3)!} \cdot \frac{1}{2} \cdot ((m-1)!)^n > |M_{u_a}| \,,$$

where the final inequality follows by (1) and $n > m$. This is a contradiction, which completes the proof. ■

Lemma 3 *Let $n > m \geqslant 4$. There is an alternative a such that $f(B_3) = \{a\}$.*

Proof. Lemma 2 implies that $B_4 \cap M_f = \emptyset$. So Lemma 1 implies that there is an alternative $a \in A$ such that $f(B_4) = \{a\}$. Let $p \in B_3 \setminus B_4$. It is sufficient to prove that $f(p) = a$. To the contrary suppose $f(p) \neq a$. Since $p \in B_3 \setminus B_4$ it follows that $|\text{topset}(p)| = 3$, say $\text{topset}(p) = \{x_1, x_2, x_3\}$.

First we show that $A \subseteq \{x_1, x_2, x_3, a\}$. To the contrary suppose that $b \in A \setminus \{x_1, x_2, x_3, a\}$. Since $n > m \geqslant 4$ we may without loss of generality assume that $n(x_1, p) \geqslant 2$. Let $i \in N(x_1, p)$ and consider an i-deviation q from p such that $b... = q(i)$ and $...f(p)...a... = q(i)$. Since $q \in B_4$, $f(q) = a$. As f is manipulable at profile q by i via p we have a contradiction with Lemma 2. Hence, $A \subseteq \{x_1, x_2, x_3, a\}$. In particular, $m = 4$ and $a \notin \{x_1, x_2, x_3\}$. We have also proved that $f(r) = a$ for any profile $r \in B_3 \setminus B_4$ such that $a \in \text{topset}(r)$.

Since $f(p) \neq a$, by tops-onliness we may assume without loss of generality that $f(p) = x_1$ and $...a...f(p)... = p(i)$ for some $i \in N(x_2, p) \cup N(x_3, p)$. Consider an i-deviation q of p with $a... = q(i)$. We claim that $f(q) = a$. Indeed, if $q \in B_4$ then this follows from $f(B_4) = \{a\}$, and if $q \in B_3 \setminus B_4$ this follows from the observation in the last sentence of the previous paragraph. But now, f is manipulable at p by i via q. Thus, by applying Remark 1 to p for an agent i in $N(x_2, p)$ and also for an agent i in $N(x_3, p)$ we obtain

$$\begin{aligned} |M_f| &\geqslant \frac{n!}{n(x_1,p)!n(x_2,p)!n(x_3,p)!} \cdot ((m-1)!)^n \cdot (1 - \left(\frac{1}{2}\right)^{n(x_2,p)}) \\ &\quad + \frac{n!}{n(x_1,p)!n(x_2,p)!n(x_3,p)!} \cdot ((m-1)!)^n \cdot (1 - \left(\frac{1}{2}\right)^{n(x_3,p)}) \\ &\geqslant \frac{n!}{n(x_1,p)!n(x_2,p)!n(x_3,p)!} \cdot ((m-1)!)^n. \end{aligned}$$

Hence

$$\frac{|M_f|}{|M_{u_a}|} \geqslant \frac{n!}{(n-2)!} \cdot \frac{2}{n(m-1)(m-2)} = \frac{(n-1)}{3} > 1,$$

where the equality follows since $m = 4$. This is a contradiction, so $f(p) = a$ and the proof is complete. ■

Lemma 4 *Let $n > m \geqslant 4$ and let $f(B_3) = \{a\}$ for some $a \in A$. Then $f(B_2) = \{a\}$.*

Proof. Let $p \in B_2 \setminus B_3$. It is sufficient to prove that $f(p) = a$. Let x and y be two alternatives and S and T be two non-empty subsets of N, such that $S = N(x, p)$, $T = N(y, p)$ and $S \cup T = N$. Let $s = |S|$ and $t = |T|$, such that $s \geqslant t$. Suppose $f(p) \neq a$.

First suppose that $t \geqslant 2$. Consider profiles $q \in B_3$ which are i-deviations of p for some $i \in N$ such that $z...f(p)...a... = q(i)$ for some alternative $z \in A \setminus \{x, y, a\}$. Because of tops-onliness we may assume that for some $j \in N$ we have $...a...f(p)... = p(j)$, where $j \in S$ if $f(p) \neq x$ and $j \in T$ if $f(p) \neq y$. So, since $f(q) = a$ it follows that f is manipulable both at p by j via q and at q by i via p. So, by applying Remark 1 to profiles q and p we have

$$|M_f| \geqslant (m-3) \cdot \frac{n!}{(s-1)!t!} \cdot ((m-1)!)^n \cdot (1 - \left(\frac{1}{2}\right)^1)$$

$$
\begin{aligned}
&+(m-3)\cdot\frac{n!}{s!(t-1)!}\cdot((m-1)!)^n\cdot(1-\left(\frac{1}{2}\right)^1)\\
&+\frac{n!}{s!t!}\cdot((m-1)!)^n\cdot(1-\left(\frac{1}{2}\right)^t)\\
=\;&(m-3)\cdot\Bigg((\frac{n!}{(s-1)!t!}+\frac{n!}{s!(t-1)!})\cdot\frac{1}{2}\\
&+\frac{n!}{s!t!}\cdot(1-\left(\frac{1}{2}\right)^t)\Bigg)\cdot((m-1)!)^n\\
=\;&(m-3)\cdot\left(\frac{n\cdot n!}{s!t!}\cdot\frac{1}{2}+\frac{n!}{s!t!}\cdot(1-\left(\frac{1}{2}\right)^t)\right)\cdot((m-1)!)^n. \qquad (6)
\end{aligned}
$$

Here, the first two terms after the inequality sign relate to manipulations at profiles q via p: there are $m-3$ possible choices for z, in the first term $i \in S$, and in the second term $i \in T$; and the last term relates to manipulations at p via profiles q. From (6), as $t \geqslant 2$,

$$
|M_f| > (m-3)\cdot\left(\frac{n\cdot n!}{2\cdot s!t!}+\frac{n!}{2\cdot s!t!}\right)\cdot((m-1)!)^n
$$

and

$$
\begin{aligned}
\frac{|M_f|}{|M_{u_a}|} &> \frac{(m-3)(n+1)\cdot n!}{2\cdot s!t!}\cdot\frac{2}{n(m-1)(m-2)}\\
&\geqslant \frac{(m-3)(n+1)(n-1)!}{2!(n-2)!(m-1)(m-2)}\\
&\geqslant \frac{(m-3)(n+1)(n-1)}{2(m-1)(m-2)}\\
&\geqslant \frac{(m-3)(m+2)m}{2(m-1)(m-2)}\\
&> 1,
\end{aligned}
$$

where the last inequality follows since $m \geq 4$. This contradicts our assumption $|M_f| \leqslant |M_{u_a}|$. Hence, $f(p) = a$ if $t \geqslant 2$.

Now let $t = 1$. Consider i-deviations q for $i \in S$ such $z... = q(i)$ for $z \in A \setminus \{a, x\}$ and $...f(p)...a... = q(i)$. Because $q \in B_3$ in case $z \neq y$ or $n(z, q) = 2$ in case $z = y$, we have $f(q) = a$ and therefore that f is manipulable at q. Hence, by applying Remark 1 to profiles q for cases where $z \neq y$ and for cases where $z = y$ we have that

$$
\begin{aligned}
|M_f| &\geqslant (m-3)\cdot\frac{n!}{(n-2)!}\cdot(1-(\frac{1}{2})^1)\cdot((m-1)!)^n\\
&\quad+\frac{n!}{(n-2)!2}\cdot(1-(\frac{1}{2})^2)\cdot((m-1)!)^n\\
&= ((m-3)\frac{1}{2}+\frac{3}{8})\cdot\frac{n!}{(n-2)!}\cdot((m-1)!)^n.
\end{aligned}
$$

Hence

$$\begin{aligned}\frac{|M_f|}{|M_{u_a}|} &\geqslant \frac{((m-3)\frac{1}{2}+\frac{3}{8})\cdot\frac{n!}{(n-2)!}}{\frac{1}{2}\cdot n\cdot(m-1)\cdot(m-2)}\\ &= \frac{(m-3)(n-1)+\frac{3}{4}(n-1)}{(m-1)(m-2)}\\ &\geqslant \frac{m(m-2\frac{1}{4})}{(m-1)(m-2)}\\ &> 1,\end{aligned}$$

where the last inequality follows since $m \geqslant 4$. This contradicts our assumption $|M_f| \leqslant |M_{u_a}|$ and therefore completes the proof. ■

The next two lemmas deal with the case $n > m = 3$.

Lemma 5 *Let $n > m = 3$. Then $B_3 \cap M_f = \emptyset$.*

Proof. Let $p \in B_3$ and suppose that f were manipulable at p by some agent, say i in $N(x_1, p)$. Remark 1 then implies that

$$|M_f| \geqslant \frac{n!}{n(x_1,p)!\cdot n(x_2,p)!n(x_3,p)!}((m-1)!)^n(1-\left(\frac{1}{2}\right)^{n(x_1,p)}).$$

So, $|M_f| \geqslant \frac{1}{2}\cdot n(n-1)\cdot((m-1)!)^n$. As $n \geqslant 4$ it follows that $|M_f| > n((m-1)!)^n = |M_{u_a}|$. This contradiction completes the proof. ■

Remark 2 By Lemmas 1 and 5 there is an $a \in A$ such that $f(B_3) = \{a\}$.

Lemma 6 *Let a be an alternative such that $f(B_3) = \{a\}$. Then $f(B_2) = \{a\}$.*

Proof. Let p be a profile in $B_2 \setminus B_3$. It is sufficient to prove that $f(p) = a$. Let x and y be two alternatives and S and T be two non-empty subsets of N, such that $S = N(x, p)$, $T = N(y, p)$ and $S \cup T = N$. Let $s = |S|$ and $t = |T|$, and assume $s \geqslant t$.

First we show that, if $a \in \{x, y\}$, then $f(p) = a$. So assume that $a \in \{x, y\}$. Suppose $f(p) \neq a$. Then there is a $z \in A \setminus \{x, y\}$, an $i \in S$, and an i-deviation q of p such that $z... = q(i)$ and $...f(p)...a... = q(i)$. Since $q \in B_3$, by assumption $f(q) = a$ and therefore f is manipulable at q by i via p. This contradicts Lemma 5. Hence, for all profiles $r \in B_2$ with $a \in \text{topset}(p)$, $f(r) = a$.

Next suppose $a \notin \{x, y\}$. First consider the case $t \geqslant 2$. Suppose $f(p) \neq a$. Let $z \in \{x, y\} \setminus \{f(p)\}$. Since f is tops-only we may assume that $z... = p(i)$ and $...a...f(p)... = p(i)$ for some $i \in N(z, p)$. Let v be an i-deviation of p such that $a... = v(i)$. As $\text{topset}(v) = \{x, y, a\}$, Lemma 5 implies $f(v) = a$. Hence, f is

manipulable at p by i via v. Remark 1 implies $|M_f| \geqslant \frac{n!}{s!t!}((m-1)!)^n(1-\left(\frac{1}{2}\right)^t)$. So

$$\begin{aligned}\frac{|M_f|}{|M_{u_a}|} &\geqslant \frac{n!}{2(n-2)!}\cdot\frac{3}{4}\cdot\frac{1}{n}\\ &\geqslant \frac{3(n-1)}{8}\\ &> 1,\end{aligned}$$

where the last inequality follows since $n \geqslant 4$. This is a contradiction and therefore $f(p) = a$.

Finally, consider the case $t = 1$ (and still $a \notin \{x, y\}$). Suppose $f(p) \neq a$. Consider, for $i \in S$, an i-deviation w of p with $y... = w(i)$ and $...f(p)...a... = w(i)$. By the previous paragraph $f(w) = a$ and therefore f is manipulable at w by i via p. By Remark 1 applied to the profile w it follows that $|M_f| \geqslant \frac{n!}{(n-2)!2!}((m-1)!)^n(1-\left(\frac{1}{2}\right)^2)$, and similarly as above this implies that $|M_f| > |M_{u_a}|$. This is a contradiction and therefore $f(B_2) = \{a\}$. ■

We are now sufficiently equipped to prove Theorem 1.

Proof of Theorem 1. Assume that $|M_f| \leqslant |M_g|$ for all $g \in F$. It is sufficient to show that f is a unanimity rule with status quo. By Lemmas 3 and 4, and Remark 2 and Lemma 6 there is an alternative $a \in A$ such that $f(B_2) = \{a\}$. For every $x \in A$ let p_x denote a profile such that $\text{topset}(p_x) = \{x\}$. By tops-onliness it is sufficient to prove that $f(p_x) = x$, for then $f = u_a$, the unanimity rule with status quo a. Let

$A_1 = \{x \in A \setminus \{a\} : f(p_x) = x\}$,
$A_2 = \{x \in A \setminus \{a\} : f(p_x) = y \text{ for some } y \notin \{x, a\}\}$,
$A_3 = \{x \in A \setminus \{a\} : f(p_x) = a\}$, and
$A_4 = \{x \in A \setminus \{a\} : f(p_a) = x\}$.

Let $m_i = |A_i|$ for $i \in \{1, 2, 3, 4\}$. Then $m_4 \in \{0, 1\}$ and, by $f(B_2) = \{a\}$ and surjectivity, $m_3 \in \{0, 1\}$ and $m_3 = 1 \Rightarrow m_4 = 1$. Hence, $m_4 \geqslant m_3$ and since $m_1 + m_2 + m_3 = m - 1$, we have

$$m_1 + m_2 + m_4 \geqslant m - 1. \tag{7}$$

By a similar argument as the one resulting in (1), there are exactly $\frac{1}{2}n(m-2)\left((m-1)!\right)^n$ manipulable profiles for each $x \in A_1$, hence in total

$$m_1 \cdot \frac{1}{2}n(m-2)\left((m-1)!\right)^n. \tag{8}$$

Now consider $x \in A_2$. The total number of profiles of the format p_x is equal to $((m-1)!)^n$. These profiles are manipulable unless $f(p_x)$ is ranked above a for each agent (since $f(B_2) = \{a\}$). This results in $n[((m-1)!)^n - ((m-1)!/2)^n] =$

$n[((m-1)!)^n \cdot (1-(1/2)^n)]$ manipulable profiles. Furthermore, if q is an i-deviation of p_x such that $...f(p_x)...a... = q(i)$ and $x... \neq q(i)$, then f is manipulable at q by i via p_x since $f(q) = a$. This results in another $n \cdot (1/2) \cdot (m! - (m-1)!) \cdot ((m-1)!)^{n-1}$ manipulable profiles, namely all such deviations with x not on top for exactly one agent and $f(p_x)$ ranked above a for the same agent. In total, this adds

$$m_2 \cdot n \left((1 - \left(\frac{1}{2}\right)^n) + \frac{1}{2}(m-1) \right) ((m-1)!)^n \tag{9}$$

manipulable profiles.
Next, consider $x \in A_4$, hence $x = f_p(a)$ and $x \neq a$. Consider an i-deviation q of p_a such that $...a...f(p_a)... = q(i)$ and $...a \neq q(i)$. Then, since $f(q) = a$, f is manipulable at p_a via q. This yields $n\,((m-1)!)^n$ manipulable profiles, namely all profiles of the format p_a. On the other hand, for an i-deviation q of p_a with $...f(p_a)...a... = q(i)$ we have that f is manipulable by i at q via p_a. Since there are $m!/2$ preferences where $f(p_a)$ is ranked above a, this results in another $(m!/2) \cdot n \cdot ((m-1)!)^{n-1} = \frac{1}{2}nm\,((m-1)!)^n$ manipulable profiles. So to the total this adds

$$m_4 \cdot n(\frac{1}{2}m + 1)\,((m-1)!)^n \tag{10}$$

manipulable profiles. Combining (1) with (8)–(10), we obtain

$$\begin{aligned}
&\tfrac{1}{2}n(m-1)(m-2)\,((m-1)!)^n \\
&\qquad \geqslant \quad |M_f| \\
&\qquad \geqslant \quad m_1 \cdot \frac{1}{2}n(m-2)\,((m-1)!)^n \\
&\qquad\qquad +m_2 \cdot n \left((1 - (\frac{1}{2})^n) + \frac{1}{2}(m-1) \right) ((m-1)!)^n \\
&\qquad\qquad +m_4 \cdot n \left(\frac{1}{2}m + 1 \right) ((m-1)!)^n . \qquad (11)
\end{aligned}$$

If $m_2 \neq 0$ or $m_4 \neq 0$ then the right-hand side of (11) is strictly larger than

$$\begin{aligned}
&\frac{1}{2}n\,((m-1)!)^n \cdot [m_1(m-2) + m_2(m-2) + m_4(m-2)] \\
&\qquad\qquad \geqslant \quad \frac{1}{2}n(m-1)(m-2)\,((m-1)!)^n ,
\end{aligned}$$

where we use (7) for the last inequality. This contradicts (11), hence $m_2 = m_4 = m_3 = 0$ and $m_1 = m - 1$. Thus, $f(p_x) = x$ for all $x \in A$. This completes the proof. ■

Since, under the conditions of Theorem 1, unanimity rules with status quo are the minimally manipulable ones among all rules in F, they are also the

minimally manipulable ones among the unanimous rules in F. Therefore, the following consequence of Theorem 1 is immediate.

Corollary 1 *Let $n > m \geqslant 3$. Let $f \in F$. Then $|M_f| \leqslant |M_g|$ for all unanimous $g \in F$ if and only if f is a unanimity rule with status quo.*

4 Conclusion

In Theorem 1 we have characterized all minimally manipulable tops-only, surjective and anonymous social choice rules—hence all minimally manipulable surjective and anonymous voting rules—under the assumption that there are more agents (voters) than alternatives (candidates). Although this covers many cases of interest, it is also worthwhile to investigate the case where the number of agents is not larger than the number of alternatives. The combinatorial arguments used to derive the results in the preceding sections can no longer be used since they depend on the assumption $n > m$.

In Maus *et al.* [17] some results for the case of two agents are established. It turns out, indeed, that unanimity rules with status quo are no longer *per se* the minimally manipulable ones among all tops-only, surjective (or even unanimous) and anonymous social choice rules. We do not have a complete characterization for this case. We do, however, have a complete characterization (for $n = 2$) if we strengthen unanimity to Pareto optimality. Call, as usual, an alternative Pareto dominated in a profile of preferences if there is another alternative that is ranked higher by all agents. A choice rule is *Pareto optimal* if it never picks a Pareto dominated alternative.

Let $R = a_1 a_2 ... a_m$ be a linear ordering of the alternatives. Let the choice rule $f_R : L(A)^{\{1,2\}} \to A$ assign to every profile p the element of topset(p) which is ranked higher under R, i.e., the element with the lower number. Obviously, f_R is tops-only, anonymous, and Pareto optimal. See [17] for a proof of the following theorem.

Theorem 2 *Let $n = 2$ and $m \geqslant 3$. Let f be a Pareto optimal, tops-only and anonymous choice rule. Then $|M_f| \leqslant |M_g|$ for all Pareto-optimal, tops-only and anonymous choice rules g if and only if $f = f_R$ for some linear ordering R of A.*

Since unanimity rules with status quo are not Pareto optimal, Theorem 1 entails that Pareto optimality is not implied by—and in fact inconsistent with—minimal manipulability among all surjective, anonymous and tops-only rules for $n > m \geqslant 3$. Since Pareto optimality is still a normatively weak requirement, it is worthwhile to investigate minimal manipulability under this stronger condition for the case of more than two agents as well. This is left to future research.

References

[1] F. Aleskerov, E. Kurbanov, Degree of manipulability of social choice procedures, in: Proceedings of the Third International Meeting of the Society for the Advancement of Economic Theory, Springer-Verlag, Berlin/Heidelberg/New York, 1999.

[2] D. Black, On the rationale of group decision making, J. Polit. Economy 56 (1948), 23–34.

[3] K.C. Border, S.J. Jordan, Straightforward elections, unanimity and phantom voters, Rev. Econ. Stud. 50 (1983), 153–170.

[4] B. Dutta, Strategic voting in a probabilistic framework, Econometrica 48 (1980), 447–456.

[5] L. Ehlers, H. Peters, T. Storcken, Threshold strategy-proofness: On manipulability in large voting systems, Games Econ. Behav. 49 (2004), 103–116.

[6] P. Fristrup, H. Keiding, Minimal manipulability and interjacency for two-person social choice functions, Soc. Choice Welfare 15 (1998), 455–467.

[7] A. Gibbard, Manipulation of voting schemes: A general result, Econometrica 41 (1973), 587–601.

[8] A. Gibbard, Manipulation of schemes that mix voting and chance, Econometrica 45 (1977), 665–681.

[9] A. Gibbard, Straightforwardness of game forms with lotteries as outcomes, Econometrica 46 (1978), 595–614.

[10] J.S. Kelly, Minimal manipulability and local strategy-proofness, Soc. Choice Welfare 5 (1988), 81–85.

[11] J.S. Kelly, Interjacency, Soc. Choice Welfare 6 (1989), 331–355.

[12] J.S. Kelly, Almost all social choice rules are highly manipulable, but a few aren't, Soc. Choice Welfare 10 (1993), 161–175.

[13] H.K. Kim, F.W. Roush, Non-manipulability in two dimensions, Math. Soc. Sci. 8 (1984), 29–43.

[14] S. Maus, H. Peters, T. Storcken, Minimal manipulability: Unanimity and non-dictatorship, working paper, RM/04/006, University of Maastricht (2004).

[15] S. Maus, H. Peters, T. Storcken, Minimal manipulability: Anonymity and surjectivity, working paper, RM/04/007, University of Maastricht (2004).

[16] S. Maus, H. Peters, T. Storcken, Minimal manipulability: Anonymity and unanimity, working paper, RM/04/0026, University of Maastricht (2004).

[17] S. Maus, H. Peters, T. Storcken, Anonymous voting and minimal manipulability, working paper, RM/05/0012, University of Maastricht (2005).

[18] H. Moulin H., On strategy-proofness and single peakedness, Public Choice 35 (1980), 437–455.

[19] P.K. Pattanaik, Counter-threats strategic manipulation under voting schemes, Rev. Econ. Stud. 43 (1976), 191–204.

[20] P.K. Pattanaik, Threats and counter-threats and strategic voting, Econometrica 44 (1976), 91–104.

[21] M.A. Satterthwaite, Strategy-proofness and Arrow's conditions: Existence and correspondence theorem for voting procedures and social welfare functions, J. Econ. Theory 10 (1975), 187–217.

[22] L. Zhou, Impossibility of strategy-proof mechanisms in economies with pure public goods, Rev. Econ. Stud. 58 (1990), 107–119.

Stefan Maus
Department of Quantitative Economics
University of Maastricht
6200 MD Maastricht, The Netherlands
Email: S.Maus@ke.unimaas.nl

Hans Peters (*corresponding author*)
Department of Quantitative Economics
University of Maastricht
6200 MD Maastricht, The Netherlands
Email: H.Peters@ke.unimaas.nl

Ton Storcken
Department of Quantitative Economics
University of Maastricht
6200 MD Maastricht, The Netherlands
Email: T.Storcken@ke.unimaas.nl

Approximability of Dodgson's Rule

John C. M^{c}Cabe-Dansted, Geoffrey Pritchard, Arkadii Slinko

Abstract

It is known that Dodgson's rule is computationally very demanding. Tideman [15] suggested an approximation to it but did not investigate how often his approximation selects the Dodgson winner. We show that under the Impartial Culture assumption the probability that the Tideman winner is the Dodgson winner tends to 1. However we show that the convergence of this probability to 1 is slow. We suggest another approximation — we call it Dodgson Quick — for which this convergence is exponentially fast.

1 Introduction

Condorcet proposed that a winner of an election is not legitimate unless a majority of the population prefer that alternative to all other alternatives. A number of voting rules have been proposed which select the Condorcet winner if it exists, and otherwise selects an alternative that is in some sense closest to being a Condorcet Winner. A prime example of such as rule was the one proposed by Dodgson [7].

Bartholdi et al. [2] proved that finding the Dodgson winner is, unfortunately, an NP-hard problem. Hemaspaandra et al. [8] refined this result by proving that it is Θ_2^p-complete and hence is not NP-complete unless the polynomial hierarchy collapses. As Dodgson's rule is hard to compute, a number of numerical studies have used approximations [14, 10]. The worst case time required to compute the Dodgson winner from a voting situation is sublinear for a fixed number of alternatives [10], however this algorithm is non-trivial to implement and its running time may grow quickly with the number of alternatives.

We investigate the asymptotic behaviour of simple approximations to the Dodgson rule as the number of agents gets large. Tideman [15] suggested an approximation but did not investigate its convergence to Dodgson. We prove that under the assumption that all votes are independent and each type of vote is equally likely, the probability that the Tideman [15] approximation picks the Dodgson winner asymptotically converges to 1, but not exponentially fast. Although the Simpson rule frequently picks the Dodgson winner [11], it does not converge to Dodgson's rule [10] and is not included in this paper.

We propose a new social choice rule, which we call Dodgson Quick. The Dodgson Quick approximation does exhibit exponential convergence to Dodgson. We may quickly verify that a particular profile has the property that forces the DQ-winner to be the Dodgson winner.

Despite its simplicity, our approximation picked the correct winner in all of 1,000,000 elections with 85 agents and 5 alternatives [10], each generated

randomly according to the Impartial Culture assumption. Our approximation can also be used to develop an algorithm to determine the Dodgson winner with $\mathcal{O}(\ln n)$ expected running time for a fixed number of alternatives and n agents.

A result independently obtained by Homan and Hemaspaandra [9] has a lot in common with our result formulated in the previous paragraph, but there are important distinctions as well. They developed a "greedy" algorithm that, given a profile, finds the Dodgson winner with certain probability. Under the Impartial Culture assumption this probability also approaches 1 as we increase the number of agents. However the Dodgson Quick rule is simpler and, unlike their algorithm, the Dodgson Quick rule requires only the information in the weighted majority relation. This makes the Dodgson Quick rule easier to study and compare with other simple rules such as the Tideman rule.

2 Preliminaries

Let A and $\mathcal{N}$ be two finite sets of cardinality m and n respectively. The elements of A will be called alternatives, the elements of $\mathcal{N}$ agents. We assume that the agents have preferences over the set of alternatives represented by (strict) linear orders. By $\mathcal{L}(A)$ we denote the set of all linear orders on A. The elements of the Cartesian product

$$\mathcal{L}(A)^n = \mathcal{L}(A) \times \cdots \times \mathcal{L}(A) \quad (n \text{ times})$$

are called **profiles**. Let $\mathcal{P} = (P_1, P_2, \ldots, P_n)$ be a profile. The linear order P_i represents the preferences of the i^{th} agent; by aP_ib, we denote that this agent prefers a to b. We define n_{xy} to be the number of linear orders in $\mathcal{P}$ that rank x above y, i.e. $n_{xy} = \#\{i \mid xP_iy\}$. A function $W^{\mathcal{P}} \colon A \times A \to \mathbb{Z}$ given by $W^{\mathcal{P}}(a,b) = n_{ab} - n_{ba}$ for all $a, b \in A$, will be called the **weighted majority relation** on $\mathcal{P}$. It is obviously skew symmetric, i.e. $W^{\mathcal{P}}(a,b) = -W^{\mathcal{P}}(b,a)$ for all $a, b \in A$.

Many of the rules to determine the winner use the numbers

$$\operatorname{adv}(a,b) = \max(0, n_{ab} - n_{ba}) = (n_{ab} - n_{ba})^+,$$

which will be called **advantages**. Note that $\operatorname{adv}(a,b) = \max(0, W(a,b)) = W(a,b)^+$ where W is the weighted majority relation on $\mathcal{P}$.

A **Condorcet winner** is an alternative a for which $\operatorname{adv}(b,a) = 0$ for all other alternatives b.

The **Dodgson score** [7, 4, 15], which we denote as $Sc_{\mathbf{d}}(a)$, of an alternative a is the minimum number of neighbouring alternatives that must be swapped to make a a Condorcet winner. We call the alternative(s) with the lowest Dodgson score the **Dodgson winner**(s).

The **Tideman score** [15] $Sc_{\mathbf{t}}(a)$ of an alternative a is

$$Sc_{\mathbf{t}}(a) = \sum_{b \neq a} \operatorname{adv}(b,a).$$

We call the alternative(s) with the lowest Tideman score the **Tideman winner**(s). Tideman [15] suggested the rule based on this score as an approximation to Dodgson.

The **Dodgson Quick (DQ) score** $Sc_{\mathbf{q}}(a)$ of an alternative a, which we introduce in this paper, is

$$Sc_{\mathbf{q}}(a) \quad = \quad \sum_{b \neq a} F(b,a), \text{ where } F(b,a) = \left\lceil \frac{\mathrm{adv}(b,a)}{2} \right\rceil .$$

We call the alternative(s) with the lowest DQ-score the **Dodgson Quick winner**(s) or **DQ-winner**.

The **Impartial Culture** assumption (IC) stipulates that all possible profiles $\mathcal{P} \in \mathcal{L}(A)^n$ are equally likely to represent the collection of preferences of an n-element society of agents $\mathcal{N}$. This assumption does not accurately reflect the voting behaviour of most voting societies and the choice of probability model can affect the similarities between approximations to the Dodgson rule [11]. However the IC is the most simplifying assumption available. As noted by Berg [3], many voting theorists have chosen to focus their research upon the IC. Thus an in depth study of the approximability of Dodgson's rule under the Impartial Culture is a natural first step.

The IC leads to the following $m!$-dimensional multinomial distribution. Let us enumerate all $m!$ linear orders in some way. Let $\mathcal{P} \in \mathcal{L}(A)^n$ be a random profile. Let then X be a vector where each X_i, for $i = 1, 2, \ldots, m!$, represents the number of occurrences of the i^{th} linear order in the profile $\mathcal{P}$. Then, under the IC, the vector X is $(n, k, \mathbf{p})$-multinomially distributed with $k = m!$ and $\mathbf{p} = \mathbf{1}_k/k = (\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k})$.

Definition 2.1 *A **weighted tournament** on a set A is any function $W\colon A \times A \to \mathbb{Z}$ satisfying $W(a,b) = -W(b,a)$ for all $a, b \in A$.*

We call $W(a,b)$ the **weight** of an ordered pair of distinct elements (a,b). One can view weighted tournaments as complete directed graphs whose edges are assigned integers characterising the intensity and the sign of the relation between the two vertices that this particular edge connects. The only condition is that if an edge from a to b is assigned integer z, then the edge from b to a is assigned the integer $-z$.

Weighted majority relation $W^{\mathcal{P}}$ on a profile $\mathcal{P}$ defined earlier in this paper is a prime example of a weighted tournament. We say that a profile $\mathcal{P}$ **generates** a weighted tournament W if $W = W^{\mathcal{P}}$. We note that $\mathrm{adv}(a,b) = W^{\mathcal{P}}(a,b)^+$, where $x^+ = \max(0,x)$. Similarly $W^{\mathcal{P}}(a,b) = \mathrm{adv}(a,b) - \mathrm{adv}(b,a)$.

The following theorem generalises the famous M$^{\text{c}}$Garvey theorem [12].

Theorem 2.2 *Let W be a weighted tournament. Then there exists a profile that generates a weighted tournament W if and only if all weights in W have the same parity [5, 13].*

3 Dodgson Quick, A New Approximation

In this section we work under the Impartial Culture assumption.

Definition 3.1 *Let $\mathcal{P} = (P_1, P_2, \ldots, P_n)$ be a profile. We say that the i^{th} agent ranks b* ***directly above*** *a if and only if aP_ib and there does not exist c different from a, b such that aP_ic and cP_ib. We define $D(b, a)$ as the number of agents who rank b directly above a.*

Lemma 3.2 *The probability that $D(x, a) > F(x, a)$ for all x converges exponentially fast to 1 as the number of agents n tends to infinity.*

Proof. As n_{ba} and $D(b, a)$ are binomially distributed with means of $n/2$ and n/m, respectively, from Chomsky's large deviation theorem [6], we know that for a fixed number of alternatives m there exist $\beta_1 > 0$ and $\beta_2 > 0$ s.t.

$$P\left(\frac{D(b,a)}{n} < \frac{1}{2m}\right) \leq e^{-\beta_1 n}, \qquad P\left(\frac{n_{ba}}{n} - \frac{1}{2} > \frac{1}{4m}\right) \leq e^{-\beta_2 n}.$$

We can rearrange the second equation to involve $F(b, a)$,

$$P\left(\frac{n_{ba}}{n} - \frac{1}{2} > \frac{1}{4m}\right) = P\left(\frac{n_{ba} - n_{ab}}{n} > \frac{1}{2m}\right) = P\left(\frac{\text{adv}(b,a)}{n} > \frac{1}{2m}\right).$$

Since $\text{adv}(b, a) \geq F(b, a)$,

$$P\left(\frac{n_{ba}}{n} - \frac{1}{2} > \frac{1}{4m}\right) \quad \geq \quad P\left(\frac{F(b,a)}{n} > \frac{1}{2m}\right).$$

From the law of probability $P(A \vee B) \leq P(A) + P(B)$ it follows that

$$P\left(\frac{F(b,a)}{n} > \frac{1}{2m}\right) \leq e^{-\beta_2 n}, \qquad P\left(\frac{D(b,a)}{n} < \frac{1}{2m}\right) \leq e^{-\beta_1 n},$$

and so for $\beta = \min(\beta_1, \beta_2)$ we obtain

$$P\left(\frac{F(b,a)}{n} > \frac{1}{2m} \text{ or } \frac{D(b,a)}{n} < \frac{1}{2m}\right) \leq e^{-\beta_1 n} + e^{-\beta_2 n} \leq 2e^{-\beta n}.$$

Hence

$$P\left(\exists_x \ \frac{F(x,a)}{n} > \frac{1}{2m} \text{ or } \frac{D(x,a)}{n} < \frac{1}{2m}\right) \quad \leq \quad 2me^{-\beta n}.$$

Using $P(\bar{E}) = 1 - P(E)$, we find that

$$P\left(\forall_x \ \frac{F(x,a)}{n} < \frac{1}{2m} < \frac{D(x,a)}{n}\right) \quad \geq \quad 1 - 2me^{-\beta n}.$$

□

Lemma 3.3 *The DQ-score $Sc_{\boldsymbol{q}}(a)$ is a lower bound for the Dodgson Score $Sc_{\boldsymbol{d}}(a)$ of a.*

Proof. Let $\mathcal{P}$ be a profile and $a \in A$. Suppose we are allowed to change linear orders in $\mathcal{P}$, by repeatedly swapping neighbouring alternatives. Then to make a a Condorcet winner we must reduce $\mathrm{adv}(x, a)$ to 0 for all x and we know that $\mathrm{adv}(x, a) = 0$ if and only if $F(x, a) = 0$. Swapping a over an alternative b ranked directly above a will reduce $n_{ba} - n_{ab}$ by two, but this will not affect $n_{ca} - n_{ac}$ where $a \neq c$. Thus swapping a over b will reduce $F(b, a)$ by one, but will not affect $F(c, a)$ where $b \neq c$. Therefore, making a a Condorcet winner will require at least $\Sigma_b F(b, a)$ swaps. This is the DQ-Score $Sc_{\mathbf{q}}(a)$ of a. □

Lemma 3.4 *If $D(x, a) \geq F(x, a)$ for every alternative x, then the DQ-Score $Sc_{\boldsymbol{q}}(a)$ of a is equal to the Dodgson Score $Sc_{\boldsymbol{d}}(a)$ and the DQ-Winner is equal to the Dodgson Winner.*

Proof. If $D(b, a) \geq F(b, a)$, we can find at least $F(b, a)$ linear orders in the profile where b is ranked directly above a. Thus we can swap a directly over b, $F(b, a)$ times, reducing $F(b, a)$ to 0. Hence we can reduce $F(x, a)$ to 0 for all x, making a a Condorcet winner, using $\Sigma_x F(x, a)$ swaps of neighbouring alternatives. In this case, $Sc_{\mathbf{q}}(a) = \Sigma_b F(b, a)$ is also an upper bound for the Dodgson Score $Sc_{\mathbf{d}}(a)$ of a. Hence $Sc_{\mathbf{q}}(a) = Sc_{\mathbf{d}}(a)$. □

Theorem 3.5 *The probability that the DQ-Score $Sc_{\boldsymbol{q}}(a)$ of an arbitrary alternative a equals the Dodgson Score $Sc_{\boldsymbol{d}}(a)$, converges to 1 exponentially fast.*

Proof. From Lemma 3.4, if $D(x, a) \geq F(x, a)$ for all alternatives x then $Sc_{\mathbf{q}}(a) = Sc_{\mathbf{d}}(a)$. From Lemma 3.2, the probability of this event converges exponentially fast to 1 as $n \to \infty$. □

Corollary 3.6 *The probability that the DQ-Winner is the Dodgson Winner converges to 1 exponentially fast as we increase the number of agents.*

Corollary 3.7 *Suppose that the number of alternatives m is fixed. Then there exists an algorithm that computes the Dodgson score of an alternative a taking as input the frequency of each linear order in the profile $\mathcal{P}$ with expected running time logarithmic with respect to the number of agents (i.e. is $\mathcal{O}(\ln n)$).*

Proof. The are at most $m!$ distinct linear orders in the profile. Hence for a fixed number of alternatives the number of distinct linear orders is bounded. Hence we may find the DQ-score and check whether $D(x, a) \geq F(x, a)$ for all alternatives x using a fixed number of additions. Additions can be performed in time linear with respect to the number of bits and logarithmic with respect

to the magnitude of the operands. So we have used an amount of time that is at worst logarithmic with respect to the number of agents.

If $D(x, a) \geq F(x, a)$ for all alternatives x, we know that the DQ-score is the Dodgson score and we do not need to go further. From Lemma 3.2 we know that the probability that we need go further declines exponentially fast, and, if this happens, we can still find the Dodgson score in time polynomial with respect to the number of agents [2]. □

4 Tideman's Rule

In this section we focus our attention on the Tideman rule which was defined in Section 2. We continue to assume the IC.

Lemma 4.1 *Given an even number of agents, the Tideman winner and the DQ-winner will be the same.*

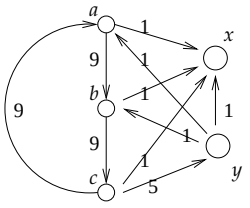
Proof. Since n is even, all weights in the majority relation W are even. Since $\text{adv}(a, b) \equiv W(a, b)^+$ it is clear that all advantages will also be even. Since $\text{adv}(a, b)$ will always be even, $\lceil \text{adv}(a, b)/2 \rceil$ will be exactly half $\text{adv}(a, b)$ and so the DQ-score will be exactly half the Tideman score. Hence the DQ-winner and the Tideman winner will be the same. □

Corollary 4.2 *Let $\mathcal{P}$ be a profile for which the Tideman winner is not the DQ-winner. Then all non-zero advantages are odd.*

Proof. As we must have an odd number of agents, all weights in the majority relation $W^{\mathcal{P}}$ must be odd. Since $\text{adv}(a, b) = W^{\mathcal{P}}(a, b)^+$ the advantage $\text{adv}(a, b)$ must be zero or equal to the weight $W^{\mathcal{P}}(a, b)$. □

Note 4.3 *There are no profiles with three alternatives where the set of DQ-winners and Tideman winners differ. There are profiles with four alternatives where the set of tied winners differ, but no such profile has a unique DQ-winner that differs from the unique Tideman winner [10].*

Example 4.4 *There exist profiles with five alternatives where there is a unique Tideman winner that differs from the unique DQ-winner. By Theorem 2.2, we know we may construct a profile whose weighted majority relation has the following advantages:*



Scores	*a*	*b*	*c*	*x*	*y*
Tideman	*10*	*10*	*9*	*4*	*5*
DQ	*6*	*6*	*5*	*4*	*3*

Here x is the sole Tideman winner, but y is the sole DQ-winner.

Theorem 4.5 *For any $m \geq 5$ there exists a profile with m alternatives and an odd number of agents, where the Tideman winner is not the DQ-winner.*

Example 4.4 demonstrates the existence of a profile with $m = 5$ alternatives for which the Tideman winner is not the Dodgson Quick winner. To extend this example for larger numbers of alternatives, we may add additional alternatives who lose to all of a, b, c, x, y. From Theorem 2.2 there exists a profile with an odd number of agents that generates that weighted majority relation.

Theorem 4.6 *If the number of agents is even, the probability that all of the advantages are 0 does not converge to 0 faster than $\mathcal{O}(n^{-\frac{m!}{4}})$.*

Proof. Let $\mathcal{P}$ be a random profile, $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{m!}\}$ be an ordered set containing all $m!$ possible linear orders on m alternatives, and X be a random vector, with elements X_i representing the number of occurrences of $\mathbf{v}_i$ in $\mathcal{P}$. Under the Impartial Culture assumption, X is distributed according to a multinomial distribution with n trials and $m!$ possible outcomes. Let us group the $m!$ outcomes into $m!/2$ pairs $S_i = \{\mathbf{v}_i, \bar{\mathbf{v}}_i\}$. Denote the number of occurrences of $\mathbf{v}$ as $n(\mathbf{v})$. Let the random variable Y_i^1 be $n(\mathbf{v}_i)$ and Y_i^2 be $n(\bar{\mathbf{v}}_i)$. Let $Y_i = Y_i^1 + Y_i^2$.

It is easy to show that, given $Y_i = y_i$ for all i, each Y_i^1 is independently binomially distributed with $p = 1/2$ and y_i trials. It is also easy to show that for an arbitrary integer $n > 0$, a $(2n, 0.5)$-binomial random variable X has a probability of at least $\frac{1}{\sqrt{2n}}$ of equaling n; thus if y_i is even then the probability that $Y_i^1 = Y_i^2$ is at least $\frac{1}{2\sqrt{y_i}}$. Combining these results we get

$$P(\forall_i Y_i^1 = Y_i^2 \mid \forall_i Y_i = y_i \in 2\mathbb{Z}) \quad \geq \quad \prod_i \frac{1}{2\sqrt{y_i}} \geq \prod_i \frac{1}{2\sqrt{n}} = 2^{-\frac{m!}{2}} n^{-\frac{m!}{4}}.$$

It is easy to show that for any k-dimensional multinomially distributed random vector, the probability that all k elements are even is at least 2^{-k+1}; hence the probability that all X_i are even is at least 2^{-k+1} where $k = m!/2$. Hence

$$P(\forall_i X_{i,1} = X_{i,2}) \quad \geq \quad \left(2^{-\frac{m!}{2}+1}\right)\left(2^{-\frac{m!}{2}} n^{-\frac{m!}{4}}\right) = 2^{1-m!} n^{-\frac{m!}{4}}.$$

If for all i, $X_{i,1} = X_{i,2}$ then for all i, $n(\mathbf{v}_i) = n(\bar{\mathbf{v}}_i)$, i.e. the number of each type of vote is the same as its complement. Thus

$$n_{ba} \quad = \quad \sum_{\mathbf{v} \in \{\mathbf{v}: b\mathbf{v}a\}} n(v) = \sum_{\bar{\mathbf{v}} \in \{\bar{\mathbf{v}}: a\bar{\mathbf{v}}b\}} n(\bar{v}) = \sum_{\mathbf{v} \in \{\mathbf{v}: a\mathbf{v}b\}} n(v) = n_{ab},$$

so $\mathrm{adv}(b, a) = 0$ for all alternatives b and a. □

Lemma 4.7 *The probability that the Tideman winner is not the DQ-winner does not converge to 0 faster than $\mathcal{O}(n^{-\frac{m!}{4}})$ as the number of agents n tends to infinity.*

Let $\mathcal{P}$ be a random profile from $\mathcal{L}(A)^n$ for some odd number n. Let $|C|$ be the size of the profile from Theorem 4.5. Let us place the first $|C|$ agents from profile $\mathcal{P}$ into sub-profile C and the remainder of the agents into sub-profile D. There is a small but constant probability that C forms the example from Theorem 4.5, resulting in the Tideman winner of C differing from its DQ-winner. As n, $|C|$ are odd, $|D|$ is even. Thus from Theorem 4.6 the probability that the advantages in D are zero does not converge to 0 faster than $\mathcal{O}(n^{-\frac{m!}{4}})$. If all the advantages in D are zero then adding D to C will not affect the Tideman or DQ-winners. Hence the probability that the Tideman winner is not the DQ-winner does not converge to 0 faster than $\mathcal{O}(n^{-\frac{m!}{4}})$.

Theorem 4.8 *The probability that the Tideman winner is not the Dodgson winner does not converge to 0 faster than $\mathcal{O}(n^{-\frac{m!}{4}})$ as the number of agents n tends to infinity.*

Proof. From Corollary 3.6 the DQ-winner converges to the Dodgson winner exponentially fast. However, the Tideman winner does not converge faster than $\mathcal{O}(n^{-\frac{m!}{4}})$ to the DQ-winner, and hence also does not converge faster than $\mathcal{O}(n^{-\frac{m!}{4}})$ to the Dodgson winner. □

Our next goal is to prove that under the IC the probability that the Tideman winner and Dodgson winner coincide converges asymptotically to 1.

Definition 4.9 *We define the adjacency matrix M, of a linear order $\mathbf{v}$, as follows:*

$$M_{ij} \quad = \quad \begin{cases} 1 & \text{if} \quad i\mathbf{v}j \\ -1 & \text{if} \quad j\mathbf{v}i \\ 0 & \text{if} \quad i = j \end{cases} .$$

Lemma 4.10 *Suppose that $\mathbf{v}$ is a random linear order chosen from the uniform distribution on $\mathcal{L}(A)$. Then its adjacency matrix M is an m^2-dimensional random variable satisfying $E[M] = 0$ and for all $i, j, r, s \in A$:*

$$cov(M_{ij}, M_{rs}) = \begin{cases} 1 & \text{if} \quad i = r \neq j = s, \\ 1/3 & \text{if} \quad i = r, \text{ but } i, j, s \text{ distinct} \vee j = s, \text{ others distinct}, \\ -1/3 & \text{if} \quad i = s, \text{ others distinct} \vee j = r, \text{ others distinct}, \\ 0 & \text{if} \quad i, j, r, s \text{ distinct} \vee i = j = r = s, \\ -1 & \text{if} \quad i = s \neq j = r. \end{cases}$$

Proof. Clearly, $E[M_{ij}] = \frac{(1)+(-1)}{2} = 0$. It is well known [1] that $\text{cov}(X, Y) = E[XY] - E[X]E[Y]$ so it follows that $\text{cov}(M_{ij}, M_{rs})E[M_{ij}M_{rs}]$.

Note that for all $i \neq j$ we know that $M_{ii}M_{ii} = 0$, $M_{ij}M_{ij} = 1$, and $M_{ij}M_{ji} = -1$. If $i = r$ and i, j, s are all distinct then the sign of $M_{ij}M_{is}$ for each permutation of i, j and s is as shown below.

	i	i	j	j	s	s
	j	s	i	s	i	j
	s	j	s	i	j	i
M_{ij}	$+$	$+$	$-$	$-$	$+$	$-$
M_{is}	$+$	$+$	$+$	$-$	$-$	$-$
$M_{ij}M_{is}$	$+$	$+$	$-$	$+$	$-$	$+$

Thus, $E[M_{ij}M_{rs}] = \frac{+1+1-1+1-1+1}{6} = \frac{1}{3}$.

If i, j, r, s are all distinct then there are six linear orders $\mathbf{v}$ where $i\mathbf{v}j$ and $r\mathbf{v}s$, six linear orders $\mathbf{v}$ where $i\mathbf{v}j$ and $s\mathbf{v}r$, six linear orders $\mathbf{v}$ where $j\mathbf{v}i$ and $r\mathbf{v}s$, and six linear orders $\mathbf{v}$ where $j\mathbf{v}i$ and $s\mathbf{v}r$. Hence,

$$E[M_{ij}M_{rs}] \quad = \frac{6(1)(1)+6(1)(-1)+6(-1)(1)+6(-1)(-1)}{24} = 0 \quad .$$

We may prove the other cases for $\text{cov}(M_{ij}, M_{rs})$ in much the same way. □

We note that as $\text{var}(X) = \text{cov}(X, X)$ we also have, $\text{var}(M_{ij}) = 1$ if $i \neq j$, and $\text{var}(M_{ij}) = 0$ if $i = j$.

Define Y to be a collection of random normal variables indexed by i, j for $1 \leq i < j \leq m$ each with mean of 0, and covariance matrix Ω, where

$$\Omega_{ij,rs} \quad = \quad \text{cov}(Y_{ij}, Y_{rs}) = \text{cov}(M_{ij}, M_{rs}),$$

We may use the fact that $i < j$, $r < s$ implies $i \neq j$, $r \neq s$, $(s = i \Rightarrow r \neq j)$ and $(r = j \Rightarrow s \neq i)$ to simplify the definition of Ω as shown below:

$$\Omega_{ij,rs} \quad = \quad \begin{cases} 1 & \text{if} \quad (r,s) = (i,j), \\ 1/3 & \text{if} \quad r = i, s \neq j \text{ or } s = j, r \neq i, \\ -1/3 & \text{if} \quad s = i \text{ or } r = j, \\ 0 & \text{if} \quad i, j, r, s \text{ are all distinct.} \end{cases}$$

Lemma 4.11 *Let $\mathcal{P} = (P_1, P_2, \ldots, P_n)$ be a profile chosen from the uniform distribution on $\mathcal{L}(A)^n$. Let M_i be the adjacency matrix of P_i. Then, as n approaches infinity, $\sum_{i=1}^{n} M_i/\sqrt{n}$ converges in distribution to*

$$\begin{bmatrix} 0 & Y_{12} & Y_{13} & \cdots & Y_{1m} \\ -Y_{12} & 0 & Y_{23} & \cdots & Y_{2m} \\ -Y_{13} & -Y_{23} & 0 & \cdots & Y_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -Y_{1m} & -Y_{2m} & -Y_{3m} & \cdots & 0 \end{bmatrix},$$

where Y is a collection of random normal variables indexed by i, j for $1 \leq i < j \leq m$ each with mean of 0, and covariance matrix Ω, where

$$\Omega_{ij,rs} \quad = \quad cov(Y_{ij}, Y_{rs}) = cov(M_{ij}, M_{rs}).$$

Proof. As $M_1, M_2, \ldots, M_n$ are independent identically-distributed (i.i.d.) random variables, we know from the multivariate central limit theorem [1, p81] that $\sum_{i=1}^{n} M_i/\sqrt{n}$ converges in distribution to the multivariate normal distribution with the same mean and covariance as the random matrix M from Lemma 4.10. As $M^T = -M$ and $M_{ii} = 0$, we have the result. □

Lemma 4.12 *Ω is non-singular.*

Proof. Consider Ω^2 with elements

$$(\Omega^2)_{ij,kl} = \sum_{1 \le r < s \le m} \Gamma_{ij,kl}(r,s),$$

where $\Gamma_{ij,kl}(r,s) = \Omega_{ij,rs}\Omega_{rs,kl}$.

If i,j,r,s distinct, then $\Gamma_{ij,ij}(i,j) = 1$ and $\Gamma_{ij,ij}(r,s) = 0$. For $(r,j),(i,s),(r,i),(j,s)$ the function $\Gamma_{ij,ij}$ evaluates to $1/9$.

Let us consider the case $(i,j) = (k,l)$. If $(i,j) = (k,l)$ then

$$\Gamma_{ij,ij}(r,s) = \Omega_{ij,rs}\Omega_{rs,ij} = \begin{cases} (1)^2 & \text{if} \quad (r,s) = (i,j), \\ (1/3)^2 & \text{if} \quad r = i, s \neq j \text{ or } s = j, r \neq i, \\ (-1/3)^2 & \text{if} \quad s = i, (r \neq j) \text{ or } r = j, (s \neq i), \\ 0 & \text{if} \quad i,j,r,s \text{ are all distinct.} \end{cases}$$

Recall that $r < s$, $i < j$ and $r, s \in [1, m]$. Let us consider for how many values of (r,s) each of the above cases occur:

- $(r,s) = (i,j)$: This occurs for exactly one value of (r,s).
- $r = i, s \neq j$: Combining the fact that $r < s$ and $r = i$ we get $i < s$. Thus $s \in (i,j) \cup (j,m]$, and there are $(j-i-1)+(m-j) = (m-i-1)$ possible values of s. As there is only one possible value of r this means that there are also $(m-i-1)$ possible values of (r,s).
- $s = j, r \neq i$: Combining the fact that $r < s$ and $s = j$ we get $r < j$. Thus $r \in [1,i) \cup (i,j)$, and there are $(i-1)+(j-i-1) = (j-2)$ possible values of (r,s).
- $s = i$: Here we want $r \neq j$, however $r < s = i < j$, so explicitly stating $r \neq j$ is redundant. Combining the fact that $r < s$ and $s = i$ we get $r < i$. Hence $r \in [1,i]$ and there are $i-1$ possible values for (r,s).
- $r = j$: Here we want $s \neq i$, however $i < j = r < s$, so explicitly stating that $r \neq j$ is redundant. From here on we will not state redundant inequalities. Combining the fact that $r < s$ and $r = j$ we get $j < s$. Hence $s \in (j,m]$ and there are $m-j$ possible values for (r,s).

Hence,

$$\begin{aligned}\sum_{1\le r<s\le m}\Gamma_{ij,ij}(r,s) &= 1+(m+j-i-3)\left(\frac{1}{9}\right)+(m+i-j-1)\left(\frac{1}{9}\right)\\ &= (9+(m+j-i-3)+(m+i-j-1))/9=\frac{2m+5}{9}.\end{aligned}$$

Let us consider now the case $i=k$, $j\neq l$. Then

$$\Gamma_{ij,il}(r,s) = \begin{cases} (1)(1/3) = 1/3 & \text{if} \quad (i,j)=(r,s),\\ (1/3)(1) = 1/3 & \text{if} \quad r=i, s=l\neq j,\\ (1/3)(1/3) = 1/9 & \text{if} \quad r=i, s\neq j, s=\neq l,\\ (1/3)(0) = 0 & \text{if} \quad s=j\neq l, r\neq i,\\ (-1/3)(-1/3) = 1/9 & \text{if} \quad s=i,\\ (-1/3)(1/3) = -1/9 & \text{if} \quad r=j, s=l,\\ (-1/3)(0) = 0 & \text{if} \quad r=j, s\neq l,\\ 0 = 0 & \text{if} \quad i,j,r,s \text{ are all distinct,}\end{cases}$$

hence,

$$\begin{aligned}\sum_{1\le r<s\le m}\Gamma_{ij,il}(r,s) &= \frac{1}{3}+\frac{1}{3}+\sum_{1\le r<s\le m, r=i, s\neq j, s=\neq l}\frac{1}{9}+\sum_{1\le r<s\le m, s=i}\frac{1}{9}-\frac{1}{9}\\ &= \frac{1}{3}+\frac{1}{3}+\sum_{i<s\le m}\frac{1}{9}-\frac{2}{9}+\sum_{1\le r<i}\frac{1}{9}-\frac{1}{9}\\ &= \frac{1}{3}+(m-i)\frac{1}{9}+(i-1)\frac{1}{9}=\frac{m+2}{9}.\end{aligned}$$

Similarly for $i\neq k$, $j=l$, we may show $(\Omega^2)_{ij,kj}=\frac{m+2}{9}$. If $j=k$ then

$$(\Omega^2)_{ij,kl} = -\frac{1}{3}-\frac{1}{3}+\frac{1}{9}-\sum_{1\le r<i, r\neq i}\frac{1}{9}-\sum_{j<s\le m, s\neq l}\frac{1}{9}=-\frac{m+2}{9},$$

similarly for $l=i$. If i,j,k,l are all distinct, $(\Omega^2)_{ij,kl}$ equals 0. Consequently

$$\Omega^2 = \left(\frac{m+2}{3}\right)\Omega-\left(\frac{m+1}{9}\right)I.$$

Since the matrix Ω satisfies $\Omega^2=\alpha\Omega+\beta I$ with $\beta\neq 0$ it has an inverse, hence Ω is not singular. □

Theorem 4.13 *The probability that the Tideman winner and Dodgson winner coincide converges asymptotically to* 1 *as* $n\to\infty$.

Proof. We will prove that the Tideman winner asymptotically coincides with the Dodgson Quick winner. The Tideman winner is the alternative $a\in A$ with the minimal value of

$$G(a) = \sum_{b\in A}\mathrm{adv}(b,a),$$

while the DQ-winner has minimal value of

$$F(a) \quad = \quad \sum_{b \in A} \left\lceil \frac{\mathrm{adv}(b,a)}{2} \right\rceil .$$

Let a_T be the Tideman winner and a_Q be the DQ-winner. Note that $G(c) - m \leq 2F(c) \leq G(c)$ for every alternative c. If for some b we have $G(b) - m > G(a_T)$, then $2F(b) \geq G(b) - m > G(a_T) \geq 2F(a_T)$ and so b is not a DQ-winner. Hence, if $G(b) - m > G(a_T)$ for all alternatives b distinct from a_T, then a_T is also the DQ-winner a_Q. Thus,

$$P(a_T \neq a_Q) \leq P(\exists_{a \neq b} \, |G(a) - G(b)| \leq m) = P\left(\exists_{a \neq b} \left| \frac{G(a) - G(b)}{\sqrt{n}} \right| \leq \frac{m}{\sqrt{n}} \right).$$

It follows that for any $\epsilon > 0$ and sufficiently large n, we have

$$P(a_T \neq a_Q) \quad \leq \quad P\left(\exists_{a \neq b} \left| \frac{G(a) - G(b)}{\sqrt{n}} \right| \leq \epsilon \right).$$

We will show that the right-hand side of the inequality above converges to 0 as n tends to ∞. All probabilities are non-negative so $0 \leq P(a_T \neq a_Q)$. From these facts and the sandwich theorem it will follow that $\lim_{n \to \infty} P(a_T \neq a_Q) = 0$.

Let

$$G_j \quad = \quad \sum_{i<j} (Y_{ij})^+ + \sum_{k>j} (-Y_{jk})^+ ,$$

where variables Y_{ij} come from the matrix (1) to which $\sum_{i=1}^{n} M_i / \sqrt{n}$ converges by Lemma 4.11. Thus,

$$\lim_{n \to \infty} P\left(\exists_{a \neq b} \left| \frac{G(a) - G(b)}{\sqrt{n}} \right| \leq \epsilon \right) \quad = \quad P\left(\exists_{i \neq j} \, |G_i - G_j| \leq \epsilon \right)$$

Since $\epsilon > 0$ is arbitrary,

$$\lim_{n \to \infty} P(a_T \neq a_Q) \quad \leq \quad P(\exists_{i \neq j} G_i = G_j).$$

For fixed $i < j$ we have

$$G_i - G_j = -Y_{ij} + \sum_{k<i} (-Y_{ki})^+ + \sum_{k>i, k \neq i} (Y_{ik})^+ - \sum_{k<j, k \neq i} (Y_{kj})^+ - \sum_{k>j} (-Y_{jk})^+ .$$

Define v so that $G_i - G_j = -Y_{ij} + v$. Then $P(G_i = G_j) = P(Y_{ij} = v) = E[P(Y_{ij} = v \,|\, v)]$. Since Y has a multivariate normal distribution with a non-singular covariance matrix Ω, it follows that $P(Y_{ij} = v \,|\, v) = 0$. That is, $P(G_i = G_j) = 0$ for any i, j where $i \neq j$. Hence $P(\exists_{i \neq j} G_i = G_j) = 0$. As discussed previously in this proof, we may now use the sandwich theorem to prove that $\lim_{n \to \infty} P(a_T \neq a_Q) = 0$. □

5 Conclusion

In this paper we showed that, under the Impartial Culture assumption, the Tideman rule converges to the Dodgson's rule when the number of agents tends to infinity. However we discovered that a new rule, which we call Dodgson Quick, approximates Dodgson's rule much better and converges to it much faster. The Dodgson Quick rule is computationally very simple, however in our simulations [10] it picked the Dodgson winner in all of 1,000,000 elections with 85 agents and 5 alternatives.

These results, the simplicity of Dodgson Quick's definition and the ease with which its winner can be computed make Dodgson Quick a highly effective tool for theoretical and numerical study of Dodgson's rule under the Impartial Culture assumption. Despite the popularity of the Impartial Culture as a simplifying assumption, it is highly unrealistic and our theorems do not apply if the slightest deviation from impartiality occurs. Our previous numerical results [11] suggest that introduction of homogeneity into the random sample may cause these approximations to diverge from the Dodgson rule. The most interesting question for further research, that this paper rises, is whether or not the Dodgson Quick rule approximates Dodgson's rule under the Impartial Anonymous Culture assumption and other models for the population.

While there is no significant difference in the difficulty of computing the Dodgson Quick winner or the Tideman winner, the Tideman rule can be easier to reason with in some circumstances. We find that the Tideman rule is often useful to study properties of the Dodgson rule where rapid convergence is not required.

References

[1] Anderson, T. W. *An Introduction to Multivariate Statistical Analysis*. John Wileys and Sons, Brisbane, 2nd edition, 1984.

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

[3] Berg, S. Paradox of voting under an urn model: The effect of homogeneity. *Public Choice*, 47:377–387, 1985.

[4] Black, D. *Theory of committees and elections*. Cambridge University Press, Cambridge, 1958.

[5] Debord, B. *Axiomatisation de procédures d'agrégation de préférences*. Ph.D. thesis, Université Scientifique, Technologique et Médicale de Grenoble, 1987. `http://tel.archives-ouvertes.fr/tel-00010237/en/`.

[6] Dembo, A. and Zeitouni, O. *Large deviations techniques*. Johns and Barlett, 1993.

[7] Dodgson, C. L. *A method for taking votes on more than two issues*. Clarendon Press, Oxford, 1876. Reprinted in (Black, 1958) with discussion.

[8] Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. 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.

[9] Homan, C. M. and Hemaspaandra, L. A. Guarantees for the success frequency of an algorithm for finding dodgson-election winners. Technical Report Technical Report TR-881, Department of Computer Science, University of Rochester, Rochester, NY, 2005. `https://urresearch.rochester.edu/retrieve/4794/tr881.pdf`.

[10] M^{c}Cabe-Dansted, J. C. *Feasibility and Approximability of Dodgson's rule*. Master's thesis, Auckland University, 2006. `http://dansted.org/thesis06`.

[11] M^{c}Cabe-Dansted, J. C. and Slinko, A. Exploratory analysis of similarities between social choice rules. *Group Decision and Negotiation*, 15:1–31, 2006. `http://dx.doi.org/10.1007/s00355-005-0052-4`.

[12] M^{c}Garvey, D. C. A theorem on the construction of voting paradoxes. *Econometrica*, 21:608–610, 1953.

[13] Ratliff, T. C. A comparison of dodgson's method and kemeny's rule. *Social Choice and Welfare*, 18(1):79–89, 2001. ISSN 0176-1714 (Print) 1432-217X (Online). `http://dx.doi.org/10.1007/s003550000060`.

[14] Shah, R. *Statistical Mappings of Social Choice Rules*. Master's thesis, Stanford University, 2003.

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

John C. M^{c}Cabe-Dansted
School of Computer Science and Software Engineering
M002 The University of Western Australia
35 Stirling Highway, Crawley 6009, Western Australia
Email: `gmatht@gmail.com`

Dr Geoffrey Pritchard
Department of Statistics, Auckland University
Private Bag 92019, Auckland 1142, New Zealand
Email: `geoff@stat.auckland.ac.nz`

Dr Arkadii Slinko
Department of Mathematics, Auckland University
Private Bag 92019, Auckland 1142, New Zealand
Email: `a.slinko@auckland.ac.nz`

Simulating the Effects of Misperception on the Manipulability of Voting Rules

Johann Mitlöhner, Daniel Eckert, and Christian Klamler

Abstract

The fact that rank aggregation rules are susceptible to manipulation by varying degrees has long been known. In this work we study the effect of noise on manipulation i.e. we assume that individuals are not able to perceive the preferences of others without distortion. To study the frequency of various outcomes we simulate a large number of rank aggregations and manipulations on random profiles with the help of a software package developed by the authors in the Python language and discuss some preliminary results.

1 Motivation

The extent to which various aggregation rules are susceptible to manipulation has been studied in a number of investigations, including simulation studies [6]. Manipulation here means to strategically misrepresent one's true preferences in order to change the election outcome to a personally more favorable one [4]. This strategic misrepresentation is based on knowledge of the other voters' preferences. In this work we relax the assumption of perfect information. We study situations where individuals manipulate while perceiving a noisy version of the other voters' preferences. Such manipulations with noisy information have the interesting property that while they change the outcome to a more favourable one given the noisy information they may fail to do so given the true preferences of the other voters, leading to situations where the manipulator is worse off than without manipulation. This can be interpreted as a form of punishment for lying; the extent to which various aggregation rules produce this effect is the subject of this paper.

For our simulation study we have implemented a set of well-known voting rules [3] in the programming language Python.[1] These rules are applied to a large number of random profiles in a setting with misperception which we describe in the next section. The details of the implementation are outlined after that, and the paper concludes with the discussion of the simulation results.

[1] At the website http://prefrule.sourceforge.net the complete package is available for download, and http://balrog.wu-wien.ac.at/∼mitloehn/prefrule provides an interactive web interface to the system.

2 Manipulation and Misperception

We assume that each voter has complete and strict preferences over the set of candidates. A profile p is a set of n strict orders over the set of candidates C, e.g. $p = ((a \succ b \succ c \succ d), (b \succ c \succ a \succ d), (c \succ a \succ b \succ d))$ denotes a profile with $n = 3$ voters and the set of $m = 4$ candidates $C = \{a, b, c, d\}$. A rank aggregation rule R applied to a profile p derives an aggregate ranking $p_* = R(p)$ which is either a strict or a weak order.

The distance d of the aggregate ranking p_* and some order p_i is measured by taking the positional difference of the winner of p_* in p_i, e.g. with $p_i = (a \succ b \succ c)$ and $p_* = (b \succ a \succ c)$ the distance is $d(p_i, p_*) = 1$.[2] A manipulation is successful if voter i is able to decrease the distance d by stating manipulated preferences p'_i, e.g. if with $p'_i = (a \succ c \succ b)$ the aggregate ranking becomes $p'_* = ((a = b) \succ c)$ then $d(p'_i, p'_*) = 0.5$.

The assumption that any voter i has perfect knowledge of the remaining profile p_{-i} i.e. the preferences of all other voters is somewhat unrealistic. In this work we explore a setting where the manipulating voter is mistaken in the perception of the remaining profile p_{-i} by a certain amount of error i.e. instead of the true p_{-i} the noisy p^e_{-i} is perceived.[3] We define the error e as the number of pairwise exchanges in adjacent pairs of candidates in some ranking(s) p_j where $j > 1$, e.g. with $p_{-i} = ((b \succ c \succ a \succ d), (a \succ c \succ b \succ d))$ and $p^e_{-i} = ((b \succ c \succ a \succ d), (c \succ a \succ b \succ d))$ we have an error of $e = 1$ since there is one switch of a and c in the last voter.

With faulty perceptions manipulations can result in a distance increase instead of a decrease. This can be viewed as a punishment for lying. The situation is described in eqs. 1 and 2.

$$d(p_i, R(p'_i, p^e_{-i})) < d(p_i, R(p_i, p^e_{-i})) \tag{1}$$

$$d(p_i, R(p'_i, p_{-i})) > d(p_i, R(p_i, p_{-i})) \tag{2}$$

Voter $i = 1$ perceives the noisy profile p^e_{-i} and based on this observation choses manipulated preferences p'_i that decrease the distance d as shown in eq. 1. However, when the aggregation rule R is applied to the manipulated preferences p'_i and the *true* remaining profile p_{-i} the result is shown in eq. 2: the distance is increased i.e. voter i has not gained but lost by manipulating.

This type of punishment would be an attractive quality of rank aggregation rules since if it occurs frequently enough it incites voters to state their true preferences and refrain from manipulation. The question remains whether a situation of this type is a rare exceptional case that has little meaning for the evaluation and comparison of aggregation rules in this respect, or a phenomenon common enough for quantitative analysis.

[2] In the case of more than one winner the average distance is used.

[3] In our simulations the manipulator is always voter $i = 1$.

In order to answer this question we define the expected benefit from manipulation $E(\Delta d)$ as the weighted sum of distance changes for the fractions of successful and failed manipulations:

$$E(\Delta d) = \frac{|S|}{|S| + |F|} \sum_{p \in S} (d^m - d^0) + \frac{|F|}{|S| + |F|} \sum_{p \in F} (d^m - d^0) \tag{3}$$

Here d^0 denote the distance without manipulation, and d^m is the distance achieved with manipulation, both as calculated when the aggregation rule is applied to the manipulated preferences p'_i of voter $i = 1$ and the true preferences p_{-i} of the remaining voters. S is the set of profiles where voter $i = 1$ successfully manipulated i.e. where $d^m < d^0$, while F is the set of profiles where manipulation failed i.e. it resulted in punishment.

3 Simulations

In order to study the frequency of the punishment effect we have implemented a number of well-known voting rules in a software package developed in the Python programming language. The simulation generates a stated number of random profiles for n voters and m candidates where rankings are independent i.e. anonymous culture. The set of rules to explore is another parameter, since some rules are computationally much more expensive than others and for that reason may be excluded in some simulation runs. The rules implemented are: Borda (BO), Copeland (CO), Kemeny (KE), Plurality (PL), Antiplurality (AP), Transitive Closure (TC), Maximin (MM), Slater (SL), Nanson (NA), Young (YO), and Dodgson (DO).

Since voters' preferences are assumed to be complete and strict a profile is implemented as a nested list with integers for the candidates, e.g. [[0,1,2],[2,0,1],[0,2,1]] for $((a \succ b \succ c), (c \succ a \succ b), (a \succ c \succ b))$. Aggregate relations are encoded as binary matrices denoting weak preference i.e. if $r_{i,j} = 1$ and $r_{j,i} = 0$ then $c_i \succ c_j$; if $r_{i,j} = r_{j,i} = 1$ than $c_i = c_j$. Therefore, the nested list [[1,1,1],[1,1,1],[0,0,1]] denotes the aggregate ranking $((a = b) \succ c)$. For printing the rankings and aggregate relations are transformed into more readable versions using plain text symbols, such as $a > b > c$. Table 1 shows a sample random profile generated by the system and the corresponding aggregate rankings resulting from various voting rules. This profile was selected for variety of results; in a typical sample the aggregate relations are much more similar [2].

The Python code is about ten to twenty times slower than a comparable version of a subset of the code written in the C programming language. However, in contrast to low level languages like C and Java used in earlier work [1] the Python language provides more clarity and elegance of syntax. Therefore it is less error-prone and saves programmer time instead of execution time. The

Table 1: Sample random profile and aggregate rankings.

```
pr: abcd,cadb,cabd,bdca,bcda
BO: [[1,0,0,1],[1,1,0,1],[1,1,1,1],[0,0,0,1]]    c>b>a>d
CO: [[1,1,1,1],[1,1,1,1],[1,1,1,1],[0,0,0,1]]    a=b=c>d
TC: [[1,1,1,1],[1,1,1,1],[1,1,1,1],[0,0,0,1]]    a=b=c>d
NA: [[1,0,0,1],[1,1,1,1],[1,0,1,1],[0,0,0,1]]    b>c>a>d
MM: [[1,0,0,1],[1,1,1,1],[1,1,1,1],[1,0,0,1]]    b=c>a=d
KE: [[1,0,0,1],[1,1,1,1],[1,0,1,1],[0,0,0,1]]    b>c>a>d
SL: [[1,1,1,1],[0,1,1,1],[0,0,1,1],[0,0,0,1]]    a>b>c>d
YO: [[1,0,0,1],[1,1,1,1],[1,1,1,1],[1,0,0,1]]    b=c>a=d
DO: [[1,0,0,1],[1,1,1,1],[1,1,1,1],[0,0,0,1]]    b=c>a>d
PL: [[1,0,0,1],[1,1,1,1],[1,1,1,1],[0,0,0,1]]    b=c>a>d
AP: [[1,0,0,1],[1,1,0,1],[1,1,1,1],[1,0,0,1]]    c>b>a=d
```

Python language has been termed "executable pseudo-code"; fig. 1 shows an example.

Table 2 shows timings taken on a Dual Core 3.2 GHz Intel Pentium D running Debian Linux. The data show that where execution time t as a function of m is concerned the positional rules Borda, Plurality, and Antiplurality, together with Copeland, Maximin, and Nanson form the most efficient group which allows them to be applied to a wide range of parameter values. The Transitive Closure rule and the Young rule form an intermediate group, while the Kemeny, Slater, and Dodgson rules show significant increases with m in execution time even in the small parameter range tabulated: the Kemeny and Slater rules with an $O(m!)$ term for trying all permutations of candidates; and the Dodgson rule with $O((nm)!)$ for trying pairwise exchanges of adjacent candidates.

4 Results

Using the software package described the manipulation with misperception has been simulated with $n = 5$ voters and $m = 4$ candidates. The error level was $e = 1$ i.e. a single misperception modelled as a pairwise exchange of adjacent candidates in the the remaining profile p_{-i} as perceived by voter $i = 1$. Table 3 shows the results.

These results are preliminary due to their limited parameter range; as such they indicate that Copeland shows the punishment effect to a much higher degree than Borda and Kemeny. Success and punishment, if they materialize at all, are pronounced most strongly in Kemeny and Slater, the only rules amoung the set investigated that always produce strict aggregate preferences. The lowest expected change in distance occurs with Transitive Closure, the rule that tends to produce a high number of indifferences in the aggregate relations.

Figure 1: This function takes a vector of scores and constructs the corresponding binary relation. The function call after the >>> prompt shows how the code can be tested in the interactive environment provided by the Python interpreter, a feature that is very useful in program development.

```
def scorel(sc):
  m=len(sc)
  r=mat(m,m)
  for i in range(m):
    for j in range(m):
      if sc[i]>=sc[j]: r[i][j]=1
  return r

>>> scorel([5,9,8,2])
[[1, 0, 0, 1], [1, 1, 1, 1], [1, 0, 1, 1], [0, 0, 0, 1]]
```

Table 2: Execution times in seconds for 1000 random profiles with $n = 9$ voters and $m = 4, 5, 6, 7, 8$ candidates.

Rule	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$
BO	0.07	0.07	0.08	0.09	0.10
CO	0.09	0.09	0.10	0.11	0.13
PL	0.07	0.07	0.08	0.09	0.10
AP	0.07	0.07	0.08	0.09	0.10
MM	0.08	0.10	0.11	0.13	0.14
NA	0.08	0.09	0.10	0.11	0.13
TC	0.15	0.25	0.43	0.72	1.21
YO	1.21	1.93	2.97	3.71	5.15
KE	0.14	0.52	3.58	31.61	318.46
SL	0.15	0.47	3.03	26.20	253.14
DO	2.32	12.31	51.98	160.56	464.17

Table 3: Results of 100000 random profiles with $n = 5$ voters and $m = 4$ candidates with error level $e = 1$. Explanation of headings: M: number of profiles with noise manipulable by some voter; M_1: number of profiles with noise manipulable by voter one; S: number of profiles with successful manipulation without noise i.e. where voter one succeeded in decreasing the distance; Δd_S: average distance decrease for successful manipulation; F: number of profiles where manipulation failed i.e. it resulted in punishment; Δd_F: average distance increase for failure i.e. punishment; $E(\Delta d)$: expected change in distance resulting from manipulation as defined in eq. 3.

Rule	M	M_1	S	Δd_S	F	Δd_F	$E(\Delta d)$
BO	59731	28551	13960	-0.715	2704	0.714	-0.484
CO	32358	8942	4185	-0.545	1430	0.635	-0.244
KE	27171	7727	3260	-1.327	1073	1.232	-0.693
PL	57534	16430	12052	-0.718	1884	0.835	-0.508
AP	52331	25976	21194	-0.675	1694	0.715	-0.572
TC	22909	7356	3853	-0.422	881	0.995	-0.158
NA	25710	9469	3621	-0.938	1349	1.068	-0.393
MM	28052	9101	4727	-0.445	828	0.805	-0.259
SL	27321	7731	3601	-1.334	1057	1.266	-0.744
YO	28052	9151	4731	-0.446	787	0.805	-0.267
DO	33698	10437	4721	-0.579	1576	0.625	-0.278

Apart from TC the Copeland rule shows the strongest punishment effect.

The data also show that the punishment effect for manipulation with misperception is not a rare exceptional case. It occurs frequently enough to provide an additional dimension for the evaluation and comparison of voting rules.

5 Conclusions

This paper described the prefrule software package for preference aggregation and reported the results of simulations of various voting rules on a large number of random profiles. Specifically, the manipulability and the effect of misperception of preferences of other voters was investigated. It has been shown that manipulators can lose rather than gain from manipulation in a setting with misperception. The susceptibility of various rank aggregation rules to these effects has been explored in simulation runs. Future work will test the validity of these results for a wider range of parameters and expand the range of applications of the software package developed for the simulations.

References

[1] Daniel Eckert, Christian Klamler, and Johann Mitlöhner. Condorcet efficiency, information costs, and the performance of scoring rules. *Central European Journal of Operations Research*, 2005:1.

[2] Daniel Eckert, Christian Klamler, Johann Mitlöhner, and Christian Schlötterer. A distance-based comparison of basic voting rules. *Proc. Joint Workshop on Decision Support Systems, Experimental Economics and e-Participation*, Graz, Austria, 2005.

[3] Peter C. Fishburn. Condorcet Social Choice Functions, *SIAM Journal of Applied Mathematics*, 33:469–489, 1977.

[4] Donald Saari. Susceptibility to manipulation. *Public Choice*, 64:21–41, 1990.

[5] Donald Saari. *Decisions and Elections - Explaining the Unexpected*. Cambridge University Press, 2001

[6] David Smith. Manipulability Measures of Common Social Choice Functions. *Social Choice and Welfare*, 16:639–661, 1999.

Johann Mitlöhner
Vienna University of Economics and Business Administration
Augasse 2–6, A-1090 Vienna, Austria
Email: `mitloehn@wu-wien.ac.at`

Daniel Eckert
University of Graz
Universitätsstr. 15/E4, A-8010 Graz, Austria
Email: `daniel.eckert@uni-graz.at`

Christian Klamler
University of Graz
Universitätsstr. 15/E4, A-8010 Graz, Austria
Email: `christian.klamler@uni-graz.at`

Weak Monotonicity and Bayes-Nash Incentive Compatibility

Rudolf Müller, Andrés Perea, Sascha Wolf

Abstract

An allocation rule is called Bayes-Nash incentive compatible, if there exists a payment rule, such that truthful reports of agents' types form a Bayes-Nash equilibrium in the direct revelation mechanism consisting of the allocation rule and the payment rule. This paper provides a characterization of Bayes-Nash incentive compatible allocation rules in social choice settings where agents have multi-dimensional types, quasi-linear utility functions and interdependent valuations. The characterization is derived by constructing complete directed graphs on agents' type spaces with cost of manipulation as lengths of edges. Weak monotonicity of the allocation rule corresponds to the condition that all 2-cycles in these graphs have non-negative length. For the case that type spaces are convex and the valuation for each outcome is a linear function in the agent's type, we show that weak monotonicity of the allocation rule together with an integrability condition is a necessary and sufficient condition for Bayes-Nash incentive compatibility.

1 Introduction

This paper is concerned with the characterization of Bayes-Nash incentive compatible allocation rules in social choice settings where agents have independently distributed, multi-dimensional types and quasi-linear utility functions, that is, utility is the valuation of an allocation minus a payment. We allow for interdependent valuations across agents. The central task addressed in this paper is the following: given such type distributions and valuations, characterize precisely those allocation rules for which there exists a payment rule such that truthful reporting of agent's types forms a Bayes-Nash equilibrium in the direct revelation mechanism consisting of the allocation rule combined with the payment rule. In addition, we aim for a framework that lets us construct a payment rule, if any, which makes a particular allocation rule Bayes-Nash incentive compatible. For example, given an allocation rule which decides in a combinatorial auction for each set of bids for each agent which set of items he wins, we want to be able to decide whether there exists a pricing scheme for winning bids that makes truthful bidding a Bayes-Nash equilibrium. If the answer is yes, we would like to have means to construct such a pricing scheme.

1.1 Related Work

An allocation rule is dominant strategy incentive compatible, if there exists a payment rule such that for any report of the other agents an agent maximizes

his own utility by reporting truthfully his type. Roberts (1979) implicitly uses a monotonicity condition on the allocation rule in order to derive his characterization of dominant strategy incentive compatible mechanisms in terms of affine maximizers for unrestricted preference domains. For a selection of restricted preference domains, Bikhchandani et al. (2003) and Lavi et al. (2003) characterize dominant strategy incentive compatibility directly in terms of a monotonicity condition on the allocation rule. Gui et al. (2004) extend these results to larger classes of preference domains by making a link to network theory. The most general results are by Saks and Yu (2005), who show that previous results extend to any convex multi-dimensional type space.

The environment considered by Saks and Yu (2005) features quasi-linear utilities and multi-dimensional types. The allocation rule maps agents' type reports into a finite set of m possible outcomes. An agent's type is a vector in $\mathbb{R}^m$ reflecting his valuation of the different possible outcomes, that is, the agent's valuation of some outcome a is given by the a^{th} element of his type vector. Agents' type spaces are assumed to be convex. Saks and Yu (2005) show that dominant strategy incentive compatible allocation rules in this setting can be characterized in terms of *weak monotonicity*, a term introduced by Lavi et al. (2003). In order to derive this result they construct complete directed graphs in the following way: Take some agent and fix a profile of type reports for the others. Now, a directed graph is constructed by associating a node with each outcome and putting a directed edge between each ordered pair of nodes. Take two outcomes a and b. Consider the difference of the valuation of a and the valuation of b with respect to every type for which truthfully reporting this type yields outcome a. The length of the network edge from a to b is defined as the infimum of all these differences. In this fashion a graph is constructed for every agent and every possible report profile of the other agents. Weak monotonicity states that for any two different outcomes a and b, the sum of the two edge lengths from a to b and from b to a is non-negative.

Earlier, Rochet (1987) characterized dominant strategy implementation in cases where the set of outcomes is not necessarily finite; an assumption that is crucial to the work of Saks and Yu (2005). He considers a setting where agents have multi-dimensional, convex type spaces and valuation functions which are linear w.r.t. their own true types. Making some additional differentiability assumptions, Rochet (1987) shows that in this case dominant strategy incentive compatibility can be characterized in terms of a monotonicity condition on the allocation rule plus an integrability condition.

Monotonicity has also been used to characterize Bayes-Nash incentive compatible allocation rules. Jehiel et al. (1999) and Jehiel and Moldovanu (2001) develop characterizations for social choice settings where agents have multi-dimensional, convex type spaces and valuation functions which are linear w.r.t. their true types. Their characterizations of Bayes-Nash incentive compatibility include a monotonicity condition on the allocation rule as well as an integrability condition comparable to the one presented by Rochet (1987).

1.2 Our Contribution

Similar to the network approach of Gui et al. (2004) and Saks and Yu (2005) we construct graphs. If an allocation rule is Bayes-Nash incentive compatible, then there exists a payment rule such that an agent's expected utility for truthfully reporting his type t is at least as high as his expected utility for misreporting some type s. Similarly, an agent's expected utility for truthfully reporting type s is at least as high as his expected utility for misreporting type t. From combining these two conditions we get a weak monotonicity condition on the allocation rule. This condition is the expected utility equivalent of the monotonicity condition mentioned in the context of dominant strategy incentive compatible allocation rules. Weak monotonicity is a necessary condition for Bayes-Nash incentive compatibility. It expresses that the expected gain in valuation for truthfully reporting t instead of misreporting s should be at least as big as the expected gain in valuation for misreporting t instead of truthfully reporting s.

Recognizing that the constraints inherent in the definition of Bayes-Nash incentive compatibility have a natural network interpretation we build complete directed graphs for agents' type spaces. To do so we associate a node with each type and put a directed edge between each ordered pair of nodes. The length of the edge going from the node associated with type s to the node associated with type t is defined as the cost of manipulation, that is, the expected difference in an agent's valuation for truthfully reporting t instead of misreporting s. Note that unlike the network approach of Gui et al. (2004) and Saks and Yu (2005) (see description above) we construct only *one* graph for each agent since we work in terms of expectations and do not consider each possible type profile of the other agents separately. Furthermore, each of these graphs contains an infinite number of nodes as we associate a node with each possible type of the agent. One could also construct outcome based graphs (as done by Gui et al., 2004; Saks and Yu, 2005) by associating a node with each possible probability distribution over outcomes. However, these graphs also contain an infinite number of nodes whenever the different possible type reports of an agent induce an infinite number of probability distributions over outcomes.

The outline of the paper is as follows: In Section 2 we state some basic assumptions and definitions. Throughout the paper we assume that agents have quasi-linear utility functions and independently distributed, privately known, multi-dimensional types. Furthermore, we allow for interdependent valuations. We do not put any restrictions on the number of possible outcomes.

In Section 3 we show that an allocation rule is Bayes-Nash incentive compatible if and only if the graphs described above contain no finite, negative length cycles. Rochet (1987) shows that dominant strategy incentive compatibility can be characterized in terms of the absence of finite, negative length cycles in similar graphs. Our result is the Bayes-Nash equivalent for his finding.

In Section 4 agents' type spaces are assumed to be convex and their valuation functions are assumed to be linear w.r.t. to their own true types. Even under

these restrictions, weak monotonicity alone is not sufficient for Bayes-Nash incentive compatibility. However, we show that weak monotonicity together with an integrability condition is both necessary and sufficient for Bayes-Nash incentive compatibility. The setting of a single-item auction with externalities considered in Jehiel et al. (1999) and the social choice setting considered in Jehiel and Moldovanu (2001) are special cases of the framework presented in this section. Compared to their settings, our multi-dimensional framework allows for a broader class of possible interdependencies between agents' valuations.

The main contribution of this paper is thus to derive for the setting described above a complete characterization of Bayes-Nash incentive compatibility in terms of weak monotonicity and an additional integrability condition. Thereby we achieve a characterization that depends purely on the valuations and the allocation rule. The characterization resembles the one derived by Rochet (1987) for dominant strategy incentive compatibility. However, our result does not follow from Rochet (1987) immediately, as we cover interdependent valuations.

2 The Model and Basic Definitions

There is a set of agents $N = \{1, \ldots, n\}$. Each agent i has a type $t^i \in T^i$ with $T^i \subseteq \mathbb{R}^k$. T denotes the set of all type profiles $t = \left(t^1, \ldots, t^n\right)$, and T^{-i} denotes the set of all type profiles $t^{-i} = \left(t^1, \ldots, t^{i-1}, t^{i+1}, \ldots, t^n\right)$. A *payment rule* is a function

$$P : T \mapsto \mathbb{R}^n,$$

so given a report profile r^{-i} of the others, reporting a type r^i results in a payment $P_i\left(r^i, r^{-i}\right)$ for agent i. Denoting the set of outcomes by Γ, an *allocation rule* is a function

$$f : T \mapsto \Gamma.$$

We allow for interdependent valuations across agents, that is, agents' valuations do not only depend on their own types but on the types of all agents. As an example one can think of an auction for a painting (see Klemperer, 1999) where agents' types reflect how much they like the painting. An agent's valuation for owning the painting depends on the types of the others as they affect the possible resale value of the painting and the owner's prestige. Take agent i having true type t^i and reporting r^i while the others have true types t^{-i} and report r^{-i}. The value that agent i assigns to the resulting allocation is denoted by $v^i\left(f\left(r^i, r^{-i}\right) \mid t^i, t^{-i}\right)$. Utilities are quasi-linear, that is, an agent's utility is his valuation of an allocation minus his payment.

Agents' types are independently distributed. Let π^i denote the probability density on T^i. The joint density π^{-i} on T^{-i} is then given by

$$\pi^{-i}\left(t^{-i}\right) = \prod_{\substack{j \in N \\ j \neq i}} \pi^j\left(t^j\right).$$

Assume that agent i believes all other agents to report truthfully. If agent i has true type t^i, then his expected utility for making a report r^i is given by

$$\begin{aligned} U^i(r^i \mid t^i) &= \int_{T^{-i}} \left(v^i\left(f\left(r^i, t^{-i}\right) \mid t^i, t^{-i}\right) - P_i\left(r^i, t^{-i}\right)\right) \pi^{-i}\left(t^{-i}\right) dt^{-i} \\ &= E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid t^i, t^{-i}\right) - P_i\left(r^i, t^{-i}\right)\right]. \end{aligned} \tag{1}$$

We assume $E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid t^i, t^{-i}\right)\right]$ to be finite $\forall r^i, t^i \in T^i$.

An allocation rule f is *Bayes-Nash incentive compatible* if there exists a payment rule P such that $\forall i \in N$ and $\forall r^i, \tilde{r}^i \in T^i$:

$$\begin{aligned} & E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid r^i, t^{-i}\right) - P_i\left(r^i, t^{-i}\right)\right] \\ \geq\ & E_{-i}\left[v^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid r^i, t^{-i}\right) - P_i\left(\tilde{r}^i, t^{-i}\right)\right]. \end{aligned} \tag{2}$$

Symmetrically, we have also

$$\begin{aligned} & E_{-i}\left[v^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid \tilde{r}^i, t^{-i}\right) - P_i\left(\tilde{r}^i, t^{-i}\right)\right] \\ \geq\ & E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid \tilde{r}^i, t^{-i}\right) - P_i\left(r^i, t^{-i}\right)\right]. \end{aligned} \tag{3}$$

By adding (2) and (3) we get the following monotonicity condition:[1]

Definition 1 (Weak Monotonicity) *An allocation rule f satisfies weak monotonicity if $\forall i \in N$ and $\forall r^i, \tilde{r}^i \in T^i$:*

$$\begin{aligned} & E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid r^i, t^{-i}\right) - v^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid r^i, t^{-i}\right)\right] \\ \geq\ & E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid \tilde{r}^i, t^{-i}\right) - v^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid \tilde{r}^i, t^{-i}\right)\right]. \end{aligned}$$

This condition is the expected utility equivalent to the weak monotonicity (W-MON) condition of Lavi et al. (2003), the non-decreasing in marginal utility condition (NDMU) of Bikhchandani et al. (2003) and the 2-cycle inequality of Gui et al. (2004). The rationale for naming the above condition weak monotonicity becomes evident once we consider valuation functions that are linear with respect to agents' types in Section 4. Obviously, weak monotonicity is a necessary condition for Bayes-Nash incentive compatibility. In Section 4 we present a setting where weak monotonicity together with an integrability condition is also a sufficient condition.

3 A Network Interpretation

We begin this section by briefly reviewing a well-known result from the field of network flow theory.[2] Let $X = \{x_1, \ldots, x_k\}$ be a finite set of variables. Consider the following system of constraints:

$$x_i - x_j \leq w_{ij} \qquad \forall i, j \in \{1, \ldots, k\}, \tag{4}$$

[1] Expected payments cancel since we work under the assumption of independently distributed types.

[2] A comprehensive introduction to network flows can be found in Ahuja et al. (1993).

where w_{ij} is some constant specific to the ordered pair (i, j). The system can be associated with a network by constructing a directed, weighted graph whose nodes correspond to the variables. A directed edge is put between each ordered pair of nodes. The length of the edge from the node corresponding to x_i to the node corresponding to x_j is given by w_{ij}.

It is a well-known result (see e.g. Shostak, 1981) that the system of linear inequalities in (4) is feasible, that is, there exists an assignment of real values to the variables such that the constraints in (4) are satisfied, if and only if there is no negative length cycle in the associated network. Furthermore, if the system is feasible then one feasible solution is to assign to each x_i the length of a shortest path from the node associated with x_i to some arbitrary source node.[3]

In order to see that the constraints in (2) have a natural network interpretation it is useful to rewrite (2) as follows:

$$\begin{aligned} & E_{-i}\left[P_i\left(r^i, t^{-i}\right) - P_i\left(\tilde{r}^i, t^{-i}\right)\right] \\ \leq \quad & E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid r^i, t^{-i}\right) - v^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid r^i, t^{-i}\right)\right]. \end{aligned} \tag{5}$$

Considering a specific allocation rule, the right-hand side of (5) is a constant. Thus, we have a system of difference constraints as described in (4) (except that we are now dealing with a potentially infinite number of variables).

Given this observation, we associate the system of inequalities (5) with a network in the same way as is described above. For each agent we build a complete directed graph T_f^i. A node is associated with each type and a directed edge is put between each ordered pair of nodes. For agent i the length of an edge directed from r^i to $\tilde{r}^i$ is denoted $l^i(r^i, \tilde{r}^i)$ and is defined as the *cost of manipulation*:

$$l^i\left(r^i, \tilde{r}^i\right) = E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid r^i, t^{-i}\right) - v^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid r^i, t^{-i}\right)\right]. \tag{6}$$

Given our previous assumptions, the edge length is finite. For technical reasons we allow for loops. However, note that an edge directed from r^i to r^i has length $l^i(r^i, r^i) = 0$.

Using this definition of the edge lengths, the weak monotonicity condition can be written as

$$l^i\left(r^i, \tilde{r}^i\right) + l^i\left(\tilde{r}^i, r^i\right) \geq 0 \qquad \forall i \in N, \forall r^i, \tilde{r}^i \in T^i.$$

So weak monotonicity corresponds to the absence of negative length 2-cycles in the graphs described above.

Rochet (1987) observed that dominant strategy incentive compatibility can be characterized in terms of the absence of finite, negative length cycles in similar graphs. Using the same proof technique, we can derive such a characterization for Bayes-Nash incentive compatibility as well.

[3] In order to be consistent with the existing literature we defined the system of constraints as in (4). However, in network theory the constraints are commonly defined as $x_j - x_i \leq w_{ij}$. In this case, if the system is feasible then one feasible solution is to assign to each x_i the length of a shortest path from some arbitrary source node to the node associated with x_i.

Theorem 1 *An allocation rule f is Bayes-Nash incentive compatible if and only if there is no finite, negative length cycle in T_f^i, $\forall i \in N$.*

Proof (Adapted from Rochet, 1987.)
Take some agent i and let $C = \left(r_1^i, \ldots, r_m^i, r_{m+1}^i = r_1^i\right)$ denote a finite cycle in T_f^i. Let us assume that f is Bayes-Nash incentive compatible. This implies, using (5) and the edge length definition (6), that for every $j \in \{1, \ldots, m\}$,

$$E_{-i}\left[P_i\left(r_j^i, t^{-i}\right) - P_i\left(r_{j+1}^i, t^{-i}\right)\right] \leq l^i\left(r_j^i, r_{j+1}^i\right).$$

Adding up these inequalities yields

$$0 \leq \sum_{j=1}^{m} l^i\left(r_j^i, r_{j+1}^i\right),$$

so C has non-negative length.

Conversely, let us assume that there exists no finite, negative length cycle in T_f^i, $\forall i \in N$. For each agent i we pick an arbitrary source node $r_0^i \in T^i$ and define $\forall r^i \in T^i$

$$p^i\left(r^i\right) = \inf \sum_{j=1}^{m} l^i\left(r_j^i, r_{j+1}^i\right),$$

where the infimum is taken over all finite paths $A = \left(r_1^i = r^i, \ldots, r_{m+1}^i = r_0^i\right)$ in T_f^i, that is, all finite paths that start at r^i and end at r_0^i. Absence of finite, negative length cycles implies that $p^i\left(r_0^i\right) = 0$. Furthermore, $\forall r^i \in T^i$ we have

$$p^i\left(r_0^i\right) \leq p^i\left(r^i\right) + l^i\left(r_0^i, r^i\right)$$

which implies that $p^i\left(r^i\right)$ is finite. For every pair $r^i, \tilde{r}^i \in T^i$ we also have

$$p^i\left(r^i\right) \leq p^i\left(\tilde{r}^i\right) + l^i\left(r^i, \tilde{r}^i\right).$$

Thus, by setting[4] $P_i\left(r^i, t^{-i}\right) = p^i\left(r^i\right)$, $\forall t^{-i} \in T^{-i}$, and using (6) we get

$$\begin{aligned} & E_{-i}\left[P_i\left(r^i, t^{-i}\right) - P_i\left(\tilde{r}^i, t^{-i}\right)\right] \\ \leq \; & E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid r^i, t^{-i}\right) - v^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid r^i, t^{-i}\right)\right]. \end{aligned}$$

Hence, the constraints in (5) are satisfied and f is Bayes-Nash incentive compatible.

□

Let us conclude this section with a condition for the costs of manipulation that is used in the derivation of the characterization theorem presented in the following section.

[4]Note that it is sufficient if P is set such that $E_{-i}\left[P_i\left(r^i, t^{-i}\right)\right] = p^i\left(r^i\right) + c$. This allows for a variety of payment rules yielding the same expected payments up to an additive constant.

Definition 2 (Decomposition Monotonicity) *The costs of manipulation are decomposition monotone if* $\forall \underline{r}^i, \bar{r}^i \in T^i$ *and* $\forall r^i \in T^i$ *s.t.* $r^i = (1-\alpha)\underline{r}^i + \alpha \bar{r}^i$, $\alpha \in (0,1)$ *we have*

$$l^i\left(\underline{r}^i, \bar{r}^i\right) \geq l^i\left(\underline{r}^i, r^i\right) + l^i\left(r^i, \bar{r}^i\right).$$

So looking at a pair of nodes, if decomposition monotonicity holds then the direct edge between those nodes is at least as long as any path connecting the same two nodes via nodes lying on the line segment between them.

4 Weak Monotonicity and Path Independence

In this section we restrict the rather general setting presented in Section 2. We assume that T^i is convex for each agent i. Furthermore, we now assume that an agent's valuation function is linear in his own true type. So if agent i has true type t^i and reports r^i while the others have true types t^{-i} and report r^{-i}, his valuation for the resulting allocation is

$$v^i\left(f\left(r^i, r^{-i}\right) \mid t^i, t^{-i}\right) = \alpha^i\left(f\left(r^i, r^{-i}\right) \mid t^{-i}\right) + \beta^i\left(f\left(r^i, r^{-i}\right) \mid t^{-i}\right) t^i. \quad (7)$$

Note that $\alpha^i : \Gamma \times T^{-i} \mapsto \mathbb{R}$ and $\beta^i : \Gamma \times T^{-i} \mapsto \mathbb{R}^k$, i.e. α^i assigns to every $\left(\gamma, t^{-i}\right) \in \Gamma \times T^{-i}$ a value in $\mathbb{R}$, whereas β^i assigns to every $\left(\gamma, t^{-i}\right) \in \Gamma \times T^{-i}$ a vector in $\mathbb{R}^k$. Similarly, assuming he believes all other agents to report truthfully, agent i's expected valuation for reporting r^i while having true type t^i is

$$\begin{aligned} & E_{-i}\left[v^i\left(f\left(r^i, t^{-i}\right) \mid t^i, t^{-i}\right)\right] \\ = \; & E_{-i}\left[\alpha^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right] + E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right] t^i. \end{aligned} \quad (8)$$

Using (8), the weak monotonicity condition becomes: $\forall i \in N$, $\forall r^i, \tilde{r}^i \in T^i$

$$E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right) - \beta^i\left(f\left(\tilde{r}^i, t^{-i}\right) \mid t^{-i}\right)\right]\left(r^i - \tilde{r}^i\right) \geq 0. \quad (9)$$

In this restricted setting weak monotonicity implies that the costs of manipulation are decomposition monotone:

Lemma 1 *Suppose that every agent i has a valuation function which is linear in his true type: If f satisfies weak monotonicity then the costs of manipulation are decomposition monotone.*

Proof
Take some agent i and let $\underline{r}^i, \bar{r}^i \in T^i$. Let $r^i \in T^i$ such that $r^i = (1-\alpha)\underline{r}^i + \alpha \bar{r}^i$ for some $\alpha \in (0,1)$. Weak monotonicity implies that

$$E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right) - \beta^i\left(f\left(\bar{r}^i, t^{-i}\right) \mid t^{-i}\right)\right]\left(r^i - \bar{r}^i\right) \geq 0.$$

Note that $\underline{r}^i - r^i$ is proportional to $r^i - \bar{r}^i$, specifically $\underline{r}^i - r^i = \frac{\alpha}{1-\alpha}\left(r^i - \bar{r}^i\right)$. Since $\alpha \in (0,1)$, the above inequality implies that

$$E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right) - \beta^i\left(f\left(\bar{r}^i, t^{-i}\right) \mid t^{-i}\right)\right]\left(\underline{r}^i - r^i\right) \geq 0.$$

Adding $E_{-i}\left[\beta^i\left(f\left(\underline{r}^i,t^{-i}\right)\mid t^{-i}\right)-\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]\underline{r}^i$ on both sides of the latter inequality and rearranging terms yields

$$\begin{aligned}
& E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)-\beta^i\left(f\left(\bar{r}^i,t^{-i}\right)\mid t^{-i}\right)\right]\underline{r}^i \\
& +E_{-i}\left[\beta^i\left(f\left(\underline{r}^i,t^{-i}\right)\mid t^{-i}\right)-\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]\underline{r}^i \\
\geq\quad & E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)-\beta^i\left(f\left(\bar{r}^i,t^{-i}\right)\mid t^{-i}\right)\right]r^i \\
& +E_{-i}\left[\beta^i\left(f\left(\underline{r}^i,t^{-i}\right)\mid t^{-i}\right)-\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]\underline{r}^i.
\end{aligned}$$

Notice that the first and the last term on the left-hand side of the inequality cancel. Hence, using (6), the above can be written as

$$l^i\left(\underline{r}^i,\bar{r}^i\right)\geq l^i\left(\underline{r}^i,r^i\right)+l^i\left(r^i,\bar{r}^i\right),$$

so the costs of manipulation are decomposition monotone.

□

It can be shown (Müller et al. 2005) that if agents' type spaces are one-dimensional then weak monotonicity is a sufficient condition for Bayes-Nash incentive compatibility. Unfortunately, if type spaces are multi-dimensional then weak monotonicity alone is not sufficient anymore (as is illustrated in Müller et al. 2005). However, in the following we are going to show that weak monotonicity together with an integrability condition is sufficient.

Definition 3 (Path Independence) *Let ψ: $T^i\mapsto\mathbb{R}^k$ be a vector field. ψ is called path independent if for any two $\underline{r}^i,\bar{r}^i\in T^i$ the path integral of ψ from $\underline{r}^i$ to $\bar{r}^i$*

$$\int_{\underline{r}^i,S}^{\bar{r}^i}\psi$$

is independent of the path of integration S.

Note that $E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]$ is a vector field $T^i\mapsto\mathbb{R}^k$.

Theorem 2 *Suppose that every agent i has a convex type space and a valuation function which is linear in his true type. Then the following statements are equivalent:*
1) f is Bayes-Nash incentive compatible.
2) f satisfies weak monotonicity and for every agent i, $E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]$ is path independent.[5]

Proof
(1)⇒(2): Let us assume that f is Bayes-Nash incentive compatible. As mentioned in Section 2, the necessity of weak monotonicity follows trivially. Furthermore, from Theorem 1 it follows that for every agent i the graph T_f^i has no

[5]That weak monotonicity of f and path independence of $E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]$ do not imply one another is illustrated in Müller et al. 2005.

finite, negative length cycles. Let $C = \left(r_1^i, \ldots, r_m^i, r_{m+1}^i = r_1^i\right)$ denote a finite cycle in T_f^i. Absence of finite, negative length cycles implies that

$$\sum_{j=1}^{m} l^i\left(r_j^i, r_{j+1}^i\right) \geq 0$$

which can be rewritten using (6) and (8) as

$$\sum_{j=1}^{m} E_{-i}\left[\beta^i\left(f\left(r_j^i, t^{-i}\right) \mid t^{-i}\right) - \beta^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid t^{-i}\right)\right] r_j^i \geq 0.$$

This implies that

$$\sum_{j=1}^{m} E_{-i}\left[\beta^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid t^{-i}\right)\right]\left(r_{j+1}^i - r_j^i\right) \geq 0.$$

Thus, $E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right]$ is cyclically monotone.[6] From Rockafellar (1970), Theorem 24.8, it follows that there exists a convex function φ: $T^i \mapsto \mathbb{R}$ such that $E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right]$ is a selection from its subdifferential mapping, that is,

$$E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right] \in \partial\varphi\left(r^i\right), \forall r^i \in T^i.$$

This implies (see Krishna and Maenner, 2001, Theorem 1) that for any smooth path S in T^i joining $\underline{r}^i$ and $\bar{r}^i$ the following holds:

$$\int_{\underline{r}^i, S}^{\bar{r}^i} E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right] = \varphi\left(\bar{r}^i\right) - \varphi\left(\underline{r}^i\right),$$

so $E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right]$ is path independent.

(2)⇒(1): Let us assume that f satisfies weak monotonicity and that for every agent i, $E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right]$ is path independent. Take any edge from T_f^i and denote its starting node $\underline{r}^i$ and its ending node $\bar{r}^i$. Let L denote the line segment between $\underline{r}^i$ and $\bar{r}^i$, i.e. $L = \left\{r^i \in T^i \mid r^i = (1-\alpha)\underline{r}^i + \alpha\bar{r}^i, \alpha \in [0,1]\right\}$. Now we pick any $r^i \in L$ and substitute the original edge with the path $A = \left(\underline{r}^i, r^i, \bar{r}^i\right)$ which has length $l^i\left(\underline{r}^i, r^i\right) + l^i\left(r^i, \bar{r}^i\right)$. By Lemma 1 we have

$$l^i\left(\underline{r}^i, \bar{r}^i\right) \geq l^i\left(\underline{r}^i, r^i\right) + l^i\left(r^i, \bar{r}^i\right), \tag{10}$$

that is, the original edge is at least as long as the path A. By repeated substitution we can generate a new path $\tilde{A} = \left(r_1^i = \underline{r}^i, \ldots, r_m^i, r_{m+1}^i = \bar{r}^i\right)$ where $r_j^i \in L, \forall j \in \{1, \ldots, m+1\}$. Then (10) implies that the original edge is at least as long as $\tilde{A}$, that is,

$$l^i\left(\underline{r}^i, \bar{r}^i\right) \geq \sum_{j=1}^{m} l^i\left(r_j^i, r_{j+1}^i\right).$$

[6]The notion of cyclical monotonicity was introduced by Rockafellar (1966).

Note that

$$
\begin{aligned}
& \sum_{j=1}^{m} l^i\left(r_j^i, r_{j+1}^i\right) \\
= & \sum_{j=1}^{m} E_{-i}\left[v^i\left(f\left(r_j^i, t^{-i}\right) \mid r_j^i, t^{-i}\right)-v^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid r_j^i, t^{-i}\right)\right] \\
= & E_{-i}\left[v^i\left(f\left(r_1^i, t^{-i}\right) \mid r_1^i, t^{-i}\right)-v^i\left(f\left(r_{m+1}^i, t^{-i}\right) \mid r_m^i, t^{-i}\right)\right] \\
& +\sum_{j=1}^{m-1} E_{-i}\left[v^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid r_{j+1}^i, t^{-i}\right)-v^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid r_j^i, t^{-i}\right)\right] \\
= & E_{-i}\left[v^i\left(f\left(r_1^i, t^{-i}\right) \mid r_1^i, t^{-i}\right)-v^i\left(f\left(r_{m+1}^i, t^{-i}\right) \mid r_{m+1}^i, t^{-i}\right)\right] \\
& +\sum_{j=1}^{m} E_{-i}\left[v^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid r_{j+1}^i, t^{-i}\right)-v^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid r_j^i, t^{-i}\right)\right] \\
= & E_{-i}\left[v^i\left(f\left(\underline{r}^i, t^{-i}\right) \mid \underline{r}^i, t^{-i}\right)-v^i\left(f\left(\bar{r}^i, t^{-i}\right) \mid \bar{r}^i, t^{-i}\right)\right] \\
& +\sum_{j=1}^{m} E_{-i}\left[\beta^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid t^{-i}\right)\right]\left(r_{j+1}^i-r_j^i\right).
\end{aligned}
$$

The first equality follows from the definition of the edge length given in (6). The second equality follows from rearranging the terms of the summation. The third equality is derived by adding and subtracting $E_{-i}\left[v^i\left(f\left(r_{m+1}^i, t^{-i}\right) \mid r_{m+1}^i, t^{-i}\right)\right]$. To derive the last equality we use (8) and that $r_1^i=\underline{r}^i$, $r_{m+1}^i=\bar{r}^i$. By repeated substitution we can generate paths with more and more edges. In the limit the distance between neighboring nodes goes to zero and

$$
\sum_{j=1}^{m} E_{-i}\left[\beta^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid t^{-i}\right)\right]\left(r_{j+1}^i-r_j^i\right) \rightarrow \int_{\underline{r}^i, L}^{\bar{r}^i} E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right].
$$

Thus, the length of $\tilde{A}$ goes to

$$
\begin{aligned}
& E_{-i}\left[v^i\left(f\left(\underline{r}^i, t^{-i}\right) \mid \underline{r}^i, t^{-i}\right)-v^i\left(f\left(\bar{r}^i, t^{-i}\right) \mid \bar{r}^i, t^{-i}\right)\right] \\
& +\int_{\underline{r}^i, L}^{\bar{r}^i} E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right], \qquad (11)
\end{aligned}
$$

as $m \rightarrow \infty$. Now, let $C=\left(r_1^i, \ldots, r_m^i, r_{m+1}^i=r_1^i\right)$ denote a finite cycle in T_f^i. Furthermore, let L_j denote the line segment between r_j^i and r_{j+1}^i. The result in (11) and the path independence of $E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right]$ imply for the

length of C that

$$\begin{aligned}
& \sum_{j=1}^{m} l^i\left(r_j^i, r_{j+1}^i\right) \\
\geq & \sum_{j=1}^{m} E_{-i}\left[v^i\left(f\left(r_j^i, t^{-i}\right) \mid r_j^i, t^{-i}\right) - v^i\left(f\left(r_{j+1}^i, t^{-i}\right) \mid r_{j+1}^i, t^{-i}\right)\right] \\
& + \sum_{j=1}^{m} \int_{r_j^i, L_j}^{r_{j+1}^i} E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right] \\
= & \ 0,
\end{aligned}$$

that is, C has non-negative length. In order to see the equality relation, note the following: the terms of the first summation cancel each other out. Furthermore, the second summation describes an integral over a closed path in T^i which, due to path independence, equals zero.

□

If f is Bayes-Nash incentive compatible, the corresponding payments can be constructed by using shortest path lengths (as described in the proof of Theorem 1). For each $i \in N$, let us pick some a^i as the source node in T_f^i. Thus, if agent i reports t^i, he has to make a payment

$$P_i\left(t^i\right) = \inf \sum_{j=1}^{m} l^i\left(r_j^i, r_{j+1}^i\right), \tag{12}$$

where the infimum is taken over all finite paths from t^i to a^i. Take any finite path $A = \left(r_1^i = t^i, \ldots, r_{m+1}^i = a^i\right)$ in T_f^i. Let L_j denote the line segment between r_j^i and r_{j+1}^i, whereas L_t denotes the line segment between the source and t^i. Following the repeated substitution approach presented in the second part of the proof of Theorem 2, we can construct paths that are shorter (or as long) by letting them visit the same nodes as A and also additional nodes along the line segments in between. In the limit, as the number of nodes goes to infinity, the distance between neighboring nodes goes to zero and the length of the paths goes to

$$\begin{aligned}
\sum_{j=1}^{m} \Big(& E_{-i}\left[v^i\left(f\left(r_j^i, t^{-i}\right) \mid r_j^i, t^{-i}\right) - v^i\left(f^i\left(r_{j+1}^i, t^{-i}\right) \mid r_{j+1}^i, t^{-i}\right)\right] \\
& + \int_{r_j^i, Lj}^{r_{j+1}^i} E_{-i}\left[\beta^i\left(f\left(r^i, t^{-i}\right) \mid t^{-i}\right)\right] \Big).
\end{aligned} \tag{13}$$

Using path independence in (13) we have that[7]

$$\sum_{j=1}^{m}\int_{r_j^i,Lj}^{r_{j+1}^i} E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]=\int_{t^i,L_t}^{a^i} E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right].$$

Applying the above to (12) yields

$$\begin{aligned}P_i\left(t^i\right) &= E_{-i}\left[v^i\left(f\left(t^i,t^{-i}\right)\mid t^i,t^{-i}\right)-v^i\left(f^i\left(a^i,t^{-i}\right)\mid a^i,t^{-i}\right)\right]\\ &\quad -\int_{a^i,L_t}^{t^i} E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right], \qquad (14)\end{aligned}$$

implying that the expected utility (see (1) for definition) for truthfully reporting t^i is[8]

$$U^i\left(t^i\mid t^i\right)=U^i\left(a^i\mid a^i\right)+\int_{a^i,L_t}^{t^i} E_{-i}\left[\beta^i\left(f\left(r^i,t^{-i}\right)\mid t^{-i}\right)\right]. \qquad (15)$$

5 Acknowledgements

The authors are grateful to the participants of the Second World Congress of the Game Theory (2004) and the First Spain Italy Netherlands Meeting on Game Theory (2005) for helpful discussions. We especially thank Philip J. Reny, Rakesh V. Vohra and an anonymous associate editor of Games and Economic Behavior for their useful comments.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, New Jersey, 1993.

[2] S. Bikhchandani, S. Chatterji, and A. Sen. Incentive compatibility in multi-unit auctions. *Levine's Bibliography*, 122247000000000750, UCLA Department of Economics, 2004.

[3] H. Gui, R. Müller, and R. V. Vohra. Dominant strategy mechanisms with multidimensional types. *METEOR Research Memorandum*, 04/046, 2004.

[4] P. Jehiel and B. Moldovanu. Efficient design with interdependent valuations. *Econometrica*, 69(5):1237–1259, 2001.

[5] P. Jehiel, B. Moldovanu, and E. Stacchetti. Multidimensional mechanism design for auctions with externalities. *Journal of Economic Theory*, 85(2):258–293, 1999.

[7] The line segment L_t for the path of integration is picked for convenience. Due to path independence, it can be replaced with any other path connecting the source and t^i.

[8] In order to derive (15) one can use that by construction $P_i\left(a^i\right)=0$ and thus add this term to the right-hand side of (14).

[6] P. Klemperer. Auction theory: A Guide to the literature. *Journal of Economic Surveys*, 13(3):227–286, 1999.

[7] V. Krishna and E. Maenner. Convex potentials with an application to mechanisms design. *Econometrica*, 69(4):1113–1119, 2001.

[8] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *Proc. 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS-2003)*. IEEE Computer Society, 2003.

[9] R. Müller, A. Perea, and S. Wolf. Weak monotonicity and Bayes-Nash incentive compatibility. *METEOR Research Memorandum*, 05/040, 2005.

[10] K. Roberts. The characterization of implementable choice rules. In *J. J. Laffont (Ed.), Aggregation and Revelation of Preferences*. North-Holland, Amsterdam, 1979.

[11] J.-C. Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16(2):191–200, 1987.

[12] R. T. Rockafellar. Characterization of the subdifferentials of convex functions. *Pacific Journal of Mathematics*, 17(3):487–510, 1966.

[13] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

[14] M. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proc. 6th ACM Conference on Electronic Commerce (EC-2005)*. ACM Press, 2005.

[15] P. Shostak. Deciding linear inequalities by computing loop residues. *Journal of the ACM*, 28(4):769–779, 1981.

Rudolf Müller
Department of Quantitative Economics, University Maastricht
P.O. Box 616, 6200 MD Maastricht, The Netherlands
Email: `r.muller@ke.unimaas.nl`

Andrés Perea
Department of Quantitative Economics, University Maastricht
P.O. Box 616, 6200 MD Maastricht, The Netherlands
Email: `a.perea@ke.unimaas.nl`

Sascha Wolf
Department of Quantitative Economics, University Maastricht
P.O. Box 616, 6200 MD Maastricht, The Netherlands
Email: `s.wolf@ke.unimaas.nl`

Voting Systems and Automated Reasoning: the QBFEVAL Case Study

Massimo Narizzano and Luca Pulina and Armando Tacchella [1]

Abstract

Systems competitions play a fundamental role in the advancement of the state of the art in several automated reasoning fields. The goal of such events is to answer the question: "Which system should I buy?". Usually the answer comes as the byproduct of a ranking obtained by considering a pool of problem instances and then aggregating the performances of the systems on each member of the pool. In this paper, we consider voting systems as an alternative to other procedures which are well established in automated reasoning contests. Our research is aimed to compare methods that are customary in the context of social choice, with methods that are targeted to artificial settings, including a new hybrid method that we introduce. Our analysis is empirical, in that we compare the aggregation procedures by computing measures which should account for their effectiveness using the data from the 2005 evaluation of quantified Boolean formulas solvers that we organized. The results of our experiments give useful indications about the relative strengths and weaknesses of the procedures under test, and allow us to infer also some conclusions that are independent of the specific procedure adopted.

1 Introduction

Systems competitions play a fundamental role in the advancement of the state of the art in several automated reasoning fields. A non-exhaustive list of such events includes the CADE ATP System Competition (CASC) [1] for theorem provers in first order logic, the SAT Competition [2] for propositional satisfiability solvers, the International Planning Competition (see, e.g., [3]) for symbolic planners, the CP Competition (see, e.g., [4]) for constraint programming systems, the Satisfiability Modulo Theories (SMT) Competition (see, e.g., [5]) for SMT solvers, and the evaluation of quantified Boolean formulas solvers (QBFEVAL, see [6, 7, 8] for previous reports). The main purpose of the above events is to designate a winner, i.e., to answer the question: "Which system should I buy?". Even if such perspective can be limiting, and the results of automated reasoning systems competitions may provide less insight than controlled experiments in the spirit of [9], there is a general agreement that competitions raise interest in the community and they help to set research challenges for developers and assess the current technological frontier for users. The usual way to designate a winner in competitions is to compute a ranking obtained by considering a pool of problem instances and then aggregating the performances of the systems on each member of the pool. While the definition of performances can encompass many

[1] The authors wish to thank the Italian Ministry of University and Research (MIUR) for its financial support, the anonymous reviewers who helped us to improve the quality of the paper, and Elena Seghezza for making some relevant references available to us.

aspects of a system, usually it is the capability of giving a sound solution to a high number of problems in a relatively short time that matters most. Therefore, one of the issues that occurred to us as organizers of QBFEVAL, relates to the procedures used to compute the final ranking of the solvers, i.e., we had to answer the question "Which aggregation procedure is best?". Indeed, even if the final rankings cannot be interpreted as absolute measures of merit, they should at least represent the relative strength of a system with respect to the other competitors based on the difficulty of the problem instances used in the contest.

Our analysis of aggregation procedures considers three voting systems, namely Borda's method [10], range voting [11] and Schulze's method [12], as an alternative to methods which are well estabished in automated reasoning contests, namely CASC [1], the SAT competitions [2], and QBFEVAL [13] (before 2006). We adapted voting systems to the artificial setting of systems competition by considering the systems as candidates and the problem instances as voters. Each instance casts its vote on the systems in such a way that systems with the best performances on the instance will be preferred over other candidates. The individual preferences are aggregated to obtain a collective choice that determines the winner of the contest. Our motivation to investigate methods which are customary in the context of social choice by applying them to the artificial setting of systems competitions is twofold. First, although voting systems do not enjoy a great popularity in automated reasoning systems contests (one exception is Robocup [14] using Borda's method), there is a substantial amount of literature in social choice (see, e.g., [15]) that deals with the problem of identifying and formalizing appropriate methods of aggregation in specific domains. Second, voting can be seen as a way to "infer the candidates' absolute goodness based on the voters' noisy signals, i.e., their votes." [16]. Therefore, the use of voting systems as aggregation procedures could pave the way to extracting hints about the absolute value of a system from the results of a contest.

In the paper, we also propose a new procedure called YASM ("Yet Another Scoring Method")[2] that we selected as an aggregation procedure for QBFEVAL'06. YASM is an hybrid between a voting system and traditional aggregation procedures used in automated reasoning contests. Our results show that YASM provides a good compromise when considering some measures that should quantify desirable properties of the aggregation procedures. In particular, the measures we propose account for:

- the degree of fidelity of the procedures, i.e., given a synthesized set of raw data, evaluate whether a procedure distorts the results;
- the degree of stability of each procedure with respect to perturbations (i) in the size of the test set, (ii) in the amount of resources available (CPU time), and (iii) in the quality of the test-set;
- the representativeness of each procedure with respect to the state of the art expressed by the competitors.

[2]The terminology "scoring method" is somewhat inappropriate in the context of social choice, as it recalls a positional scoring procedure such as Borda's method and range voting; we decided to keep the original terminology for consistency across the previous works [17, 18, 21].

We compute the above measures using part of the results from QBFEVAL'05 [8]. In particular, the results of our experiments give useful indications about the relative strengths and weaknesses of the aggregation procedures, and allow us to infer also some conclusions that are independent of the specific method adopted.

This paper builds on and extends previous work by one of the authors [17]. First, the version of YASM that we present here improves on the one presented in [17]. In particular, the new YASM is simpler and more effective when compared to the old one. Moreover, the comparison of aggregation procedures is broadened by the addition of new effectiveness measures (fidelity, see Section 4), and an improved definition of State-Of-The-Art (SOTA) relevance (see Section 4).

The paper is structured as follows. In Section 2 we introduce the case study of QBFEVAL'05 [8], and we introduce the state of the art aggregation procedures. In Section 3 we introduce our new aggregation procedure, and then we compare it with other methods in Section 4 using several effectiveness measures. We conclude the paper in Section 5 with a discussion about the presented results.

2 Preliminaries

2.1 QBFEVAL'05

QBFEVAL'05 [8] is the third in a series of non-competitive events that preceded QBFEVAL'06. QBFEVAL'05 accounted for 13 competitors, 553 quantified Boolean formulas (QBFs) and three QBF generators submitted. The test set was assembled using a selection of 3191 QBFs obtained considering the submissions and the instances archived in QBFLIB [19]. The results of QBFEVAL'05 can be listed in a table RUNS comprised of four attributes (column names): SOLVER, INSTANCE, RESULT, and CPUTIME. The attributes SOLVER and INSTANCE report which solver is run on which instance. RESULT is a four-valued attribute: SAT, i.e., the instance was found satisfiable by the solver, UNSAT, i.e., the instance was found unsatisfiable by the solver, TIME, i.e., the solver exceeded a given time limit without solving the instance (900 seconds in QBFEVAL'05), and FAIL, i.e., the solver aborted for some reason (e.g., a run-time error, an inherent limitation of the solver, or any other reason beyond our control). Finally, CPUTIME reports the CPU time spent by the solver on the given instance, in seconds. In the analysis herewith presented we used a subset of QBFEVAL'05 RUNS table, including only the solvers that, as far as we know, work correctly (the solvers of the second stage of the evaluation) and the QBFs coming from classes of instances having fixed structure (see [8] for more details). Under these assumptions, RUNS table reduces to 4408 entries, one order of magnitude less than the original one. This choice allows us to disregard correctness issues, to reduce considerably the overhead of the computations required for our analysis, and, at the same time, maintain a significant number of runs. The aggregation procedures that we evaluate, the measures that we compute and the results that we obtain, are based on the assumption that a table identical to RUNS as described above is the only input required by a procedure. As a consequence, the aggregation procedures (and thus our analysis) do not take into

account (i) memory consumption, (ii) correctness of the solution, and (iii) "quality" of the solution.

2.2 State of the art aggregation procedures

In the following we describe in some details the state of the art aggregation procedures used in our analysis. For each method we describe only those features that are relevant for our purposes. Further details can be found in the references provided.

CASC [1] Using CASC methodology, the solvers are ranked according to the number of problems solved, i.e., the number of times RESULT is either SAT or UNSAT. Under this procedure, solver A is better than solver B, if and only if A is able to solve at least one problem more than B within the time limit. In case of a tie, the solver faring the lowest average on CPUTIME fields over the problems solved is the one which ranks first.

QBF evaluation [13] QBFEVAL methodology is the same as CASC, except for the tie-breaking rule, which is based on the sum of CPUTIME fields over the problems solved.

SAT competition [2] The last SAT competition uses a *purse-based method*, i.e., the measure of effectiveness of a solver on a given instance is obtained by adding up three purses:

- the solution purse, which is divided equally among all solvers that solve the problem;
- the speed purse, which is divided unequally among all the competitors that solve the problem, first by computing the speed factor $F_{s,i}$ of a solver s on a problem instance i:
$$F_{s,i} = \frac{k}{1 + T_{s,i}} \tag{1}$$
where k is an arbitrary scaling factor (we set $k = 10^4$ according to [20]), and $T_{s,i}$ is the time spent by s to solve i; then by computing the speed award $A_{s,i}$, i.e., the portion of speed purse awarded to the solver s on the instance i:
$$A_{s,i} = \frac{P_i \cdot F_{s,i}}{\sum_r F_{r,i}} \tag{2}$$
where r ranges over the solvers, and P_i is the total amount of the speed purse for the instance i.
- the series purse, which is divided equally among all solvers that solve at least one problem in a given series (a series is a family of instances that are somehow related, e.g., different QBF encodings for some problem in a given domain).

The overall ranking of the solvers under this method is obtained by considering the sum of the purses obtained on each instance, and the winner of the contest is the solver with the highest sum.

Borda's method [10] Suppose that n solvers (candidates) and m instances (voters) are involved in the contest. Consider the sorted list of solvers obtained for each instance by considering the value of the CPUTIME field in ascending order. Let $p_{s,i}$ be the position of a solver s ($1 \leq s \leq n$) in the list associated with instance i ($1 \leq i \leq m$). According to Borda's method, each voter's ballot consists of a vector of individual scores given to candidates, where the score $S_{s,i}$ of solver s on instance i is simply $S_{s,i} = n - p_{s,i}$. In cases of time limit attainment or failure, we default $S_{s,i}$ to 0. The score of a candidate, given the individual preferences, is just $S_s = \sum_i S_{s,i}$, and the winner is the solver with the highest score.

Range voting [11] Again, suppose that n solvers and m instance are involved in the contest and $p_{s,i}$ is obtained as described above for Borda's method. We let the score $S_{s,i}$ of solver s on instance i be the quantity $ar^{n-p_{s,i}}$, i.e., we use a positional scoring rule following a geometric progression with a common ratio $r = 2$ and a scale factor $a = 1$. We consider failures and time limit attainments in the same way (we call this the failure-as-time-limit model in [21]), and thus we assume that all the voters express an opinion about all the solvers. The overall score of a candidate is again $S_s = \sum_i S_{s,i}$ and the candidate with the highest score wins the election.

Schulze's method We denote as such an extension of the method described in Appendix 3 of [12]. Since Schulze's method is meant to compute a single overall winner, we extended the method according to Schulze's suggestions [22] in order to make it capable of generating an overall ranking.

3 YASM: Yet Another Scoring Method (Revisited)

While the aggregation procedures used in CASC and QBF evaluations are straightforward, they do not take into account some aspects that are indeed considered by the purse-based method used in the last SAT competition. On the other hand, the purse-based method used in SAT requires some oracle to assign purses to the problem instances, so the results can be influenced heavily by the oracle. In [17] a first version of YASM was introduced as an attempt to combine the two approaches: a rich method like the purse-based one, but using the data obtained from the runs only. As reported in [17], YASM featured a somewhat complex calculation, yielding unsatisfactory results, particularly in the comparison with the final ranking produced by voting systems. Here we revise the original version of YASM to make its computation simpler, and to improve its performance using ideas borrowed from voting systems. From here on, we call YASMv2 the revised version, and YASM the original one presented in [17]. YASMv2 requires a preliminary classification whereby a hardness degree H_i is assigned to each problem instance i using the same equation as in CASC [1] (and YASM):

$$H_i = 1 - \frac{S_i}{S_t} \tag{3}$$

	CASC	QBF	SAT	YASM	YASMv2	Borda	r.v.	Schulze
CASC	–	1	0.71	0.86	0.79	0.86	0.71	0.86
QBF		–	0.71	0.86	0.79	0.86	0.71	0.86
SAT			–	0.86	0.86	0.71	0.71	0.71
YASM				–	0.86	0.71	0.71	0.71
YASMv2					–	0.86	0.86	0.86
Borda						–	0.86	1
r. v.							–	0.86
Schulze								–

Table 1: Homogeneity of aggregation procedures.

where S_i is the number of solvers that solved i, and S_t is the total number of participants to the contest. Considering equation (3), we notice that $0 \leq H_i \leq 1$, where $H_i = 0$ means that i is relatively easy, while $H_i = 1$ means that i is relatively hard. We can then compute the measure of effectiveness $S_{s,i}$ of a solver s on a given instance i (this definition changes with respect to YASM):

$$S_{s,i} = k_{s,i} \cdot (1 + H_i) \cdot \frac{L - T_{s,i}}{L - M_i} \tag{4}$$

where L is the time limit, $T_{s,i}$ is the CPU time used up by s to solve i ($T_{s,i} \leq L$), and $M_i = min_s\{T_{s,i}\}$, i.e., M_i is the time spent on the instance i by the *SOTA solver* defined in [8] to be the ideal solver that always fares the best time among all the participants. The hybridization with voting systems comes into play with the coefficient $k_{s,i}$ which is computed as follows. Suppose that n solvers are participating to the contest. Each instance ranks the solvers in ascending order considering the value of the CPUTIME field. Let $p_{s,i}$ be the position of a solver s in the ranking associated with instance i ($1 \leq p_{s,i} \leq n$), then $k_{s,i} = n - p_{s,i}$. In case of time limit attainment and failure, we default $k_{s,i}$ to 0, and thus also $S_{s,i}$ is 0. The overall ranking of the solvers is computed by considering the values $S_s = \sum_i S_{s,i}$ for all $1 \leq s \leq n$, and the solver with the highest sum wins.

We can see from equation (4) that in YASMv2 the effectiveness of a solver on a given instance is influenced by three factors, namely (i) a Borda-like positional weight ($k_{s,i}$), (ii) the relative hardness of the instance ($1 + H_i$), and (iii) the relative speed of the solver with respect to the fastest solver on the instance ($\frac{L-T_{s,i}}{L-M_i}$). Intuitively, coefficient (ii) rewards the solvers that are able to solve hard instances, while (iii) rewards the solvers that are faster than other competitors. The coefficient $k_{s,i}$ has been added to stabilize the final ranking and make it less sensitive to an initial bias in the test set. As we show in the next Section, this combination allows YASMv2 to reach the best compromise among different effectiveness measures.

4 Experimental Evaluation

4.1 Homogeneity

The rationale behind this measure (introduced in [17]) is to verify that, on a given test set, the aggregation procedures considered (i) do not produce exactly the same solver

Method	Mean	Std	Median	Min	Max	IQ Range	F
QBF	182.25	7.53	183	170	192	13	88.54
CASC	182.25	7.53	183	170	192	13	88.54
SAT	87250	12520.2	83262.33	78532.74	119780.48	4263.94	65.56
YASM	46.64	2.22	46.33	43.56	51.02	2.82	85.38
YASMv2	1257.29	45.39	1268.73	1198.43	1312.72	95.11	91.29
Borda	984.5	127.39	982.5	752	1176	194.5	63.95
r. v.	12010.25	5183.86	12104	5186	21504	8096	24.12
SCHULZE	–	–	–	–	–	–	–

Table 2: Fidelity of aggregation procedures. As far as **SAT** is concerned, the series purse is not assigned.

rankings, but, at the same time, (ii) do not yield antithetic solver rankings. Thus, homogeneity is not an effectiveness measure per se, but it is a preliminary assessment that we are performing an apple-to-apple comparison and that the apples are not exactly the same.

Homogeneity is computed as in [17] considering the Kendall rank correlation coefficient τ which is a nonparametric coefficient best suited to compare rankings. τ is computed between any two rankings and it is such that $-1 \leq \tau \leq 1$, where $\tau = -1$ means perfect disagreement, $\tau = 0$ means independence, and $\tau = 1$ means perfect agreement. Table 1 shows the values of τ computed for the aggregation procedures considered, arranged in a symmetric matrix where we omit the elements below the diagonal (r.v. is a shorthand for range voting). Values of τ close to, but not exactly equal to 1 are desirable. Table 1 shows that this is indeed the case for the aggregation procedures considered using QBFEVAL'05 data. Only two couples of methods (QBF-CASC and Schulze-Borda) show perfect agreement, while all the other couples agree to some extent, but still produce different rankings.

4.2 Fidelity

We introduce this measure to check whether the aggregation procedures under test introduce any distortion with respect to the true merits of the solvers. Our motivation is that we would like to extract some scientific insight from the final ranking of QBFEVAL'06 and not just winners and losers. Of course, we have no way to know the true merits of the QBF solvers: this would be like knowing the true statistic of some population. Therefore, we measure fidelity by feeding each aggregation procedure with "white noise", i.e., several samples of table RUNS having the same structure outlined in Subsection 2.1 and filled with random results. In particular, we assign to RESULT one of SAT/UNSAT, TIME and FAIL values with equal probability, and a value of CPUTIME chosen uniformly at random in the interval [0;1]. Given this artificial setting, we know in advance that the true merit of the competitors is approximately the same. A high-fidelity aggregation procedure is thus one that computes approximately the same scores for each solver, and thus produces a final ranking where scores have a small variance-to-mean ratio.

The results of the fidelity test are presented in Table 2 where each line contains the statistics of a aggregation procedure. The columns show, from left to right, the mean,

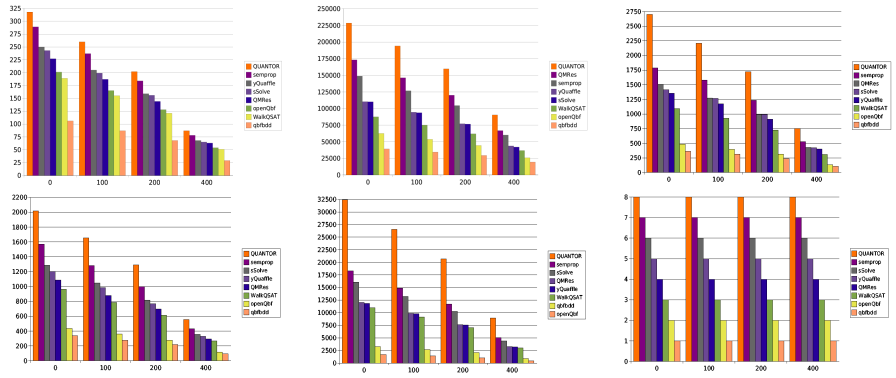


Figure 1: RDT-stability plots.

the standard deviation, the median, the minimum, the maximum and the interquartile range of the scores produced by each aggregation procedure when fed by white noise. The last column is our fidelity coefficient F, i.e., the percent ratio between the lowest score (solver ranked last) and the highest one (solver ranked first): the higher the value of F, the more the fidelity of the aggregation procedure. As we can see from Table 2, the fidelity of YASMv2 is better than that of all the other methods under test, including QBF and CASC which are second best, and have higher fidelity than YASM. Notice that range voting, and to a lesser extent also SAT and Borda's methods, introduce a substantial distortion. In the case of range voting, this can be explained by the exponential spread that separates the scores, and thus amplifies even small differences. Measuring fidelity does not make sense in the case of Schulze's method. Indeed, given the characteristics of the "white noise" data set, Schulze's method yields a tie among all the solvers. Thus, checking for fidelity would essentially mean checking the tie-breaking heuristic, and not the main method.

4.3 RDT-stability and DTL-stability

Stability on a randomized decreasing test set (RDT-stability), and stability on a decreasing time limit (DTL-stability) have been introduced in [17] to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set, and how much an aggregation procedure is sensitive to perturbations that diminish the maximum amount of CPU time granted to the solvers, respectively. The results of RDT- and DTL-stability tests are presented in the plots of Figures 1 and 2. We obtained such plots considering the CPU time noise model in [18], and considering YASMv2 instead of YASM and the Schulze's method instead of the sum of victories method.

On Figure 1, the first row shows, from left to right, the plots regarding QBF/CASC, SAT and YASMv2 procedures, while the second row shows, again from left to right, the plots regarding Borda's method, range voting and Schulze's method. Each histogram reports, on the x-axis the number of problems m discarded from the origi-

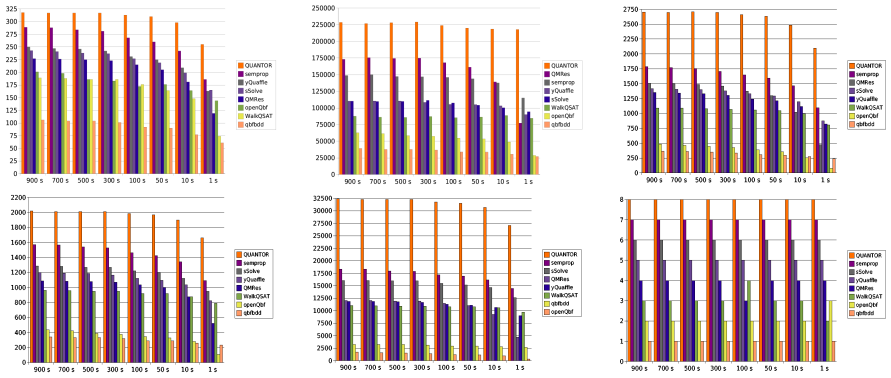


Figure 2: DTL-stability plots.

nal test set (0, 100, 200 and 400 out of 551) and on the y-axis the score. Schulze's scores are the straightforward translation of the ordinal ranking derived by applying the method which is not based on cardinal ranking. For each value of the x-axis, eight bars are displayed, corresponding to the scores of the solvers. The legend is sorted according to the ranking computed by the specific procedure, and the bars are also displayed accordingly. This makes easier to identify perturbations of the original ranking, i.e., the leftmost group of bars in each plot corresponding to $m = 0$. On Figure 2, the histograms are arranged in the same way as Figure 1, except that the x-axis now reports the amount of CPU time seconds used as a time limit when evaluating the scores of the solvers. The leftmost value is $L = 900$, i.e., the original time limit that produces the ranking according to which the legend and the bars are sorted, and then we consider the values $L' = \{700, 500, 300, 100, 50, 10, 1\}$.
The conclusion that we reach are the same of [17], and precisely:

- All the aggregation procedures considered are RDT-stable up to 400, i.e., a random sample of 151 instances is sufficient for all the procedures to reach the same conclusions that each one reaches on the heftier set of 551 instances used in QBFEVAL'05.

- Decreasing the time limit substantially, even up to one order of magnitude, is not influencing the stability of the aggregation procedures considered, except for some minor perturbations for QBF/CASC, SAT and Schulze's methods. Moreover, independently from the procedure used and the amount of CPU time granted, the best solver is always the same.

Indeed, while the above measures can help us extract general guidelines about running a competition, in our setting they do not provide useful insights to discriminate the relative merits of the procedures.

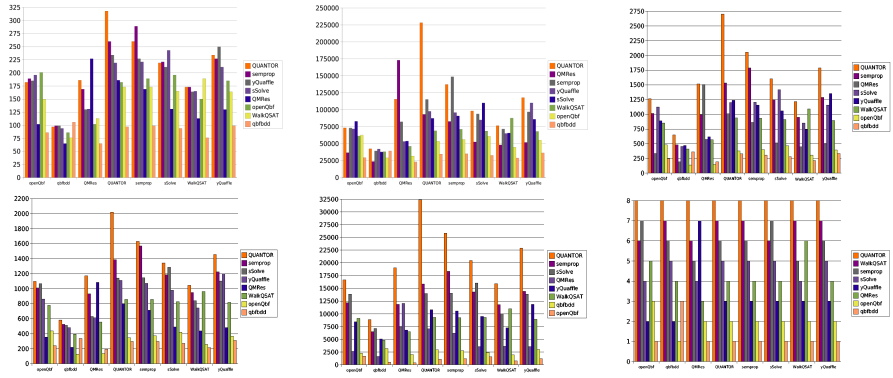


Figure 3: SBT-stability plots.

4.4 SBT-stability

Stability on a solver biased test set (SBT-stability) is introduced in [17] to measure how much an aggregation procedure is sensitive to a test set that is biased in favor of a given solver. Let Γ be the original test set, and Γ_s be the subset of Γ such that the solver s is able to solve exactly the instances in Γ_s. Let $R_{q,s}$ be the ranking obtained by applying the aggregation procedure q on Γ_s. If $R_{q,s}$ is the same as the original ranking R_q, then the aggregation procedure q is SBT-stable with respect to the solver s. Notice that, contrarily to what stated in [17], SBT-stability alone is not a sufficient indicator of the capacity of an aggregation procedure to detect the absolute merit of the participants. Indeed, it turns out that a very low-fidelity method such as range voting is remarkably SBT-stable. This because we can raise the SBT-stability of a ranking by decreasing its fidelity: in the limit, a aggregation procedure that assigns fixed scores to each solver, has the best SBT-stability and the worst fidelity. Therefore, an aggregation procedure showing a high SBT-stability is relatively immune to bias in the test set, but it must also feature a high fidelity if we are to conclude that the method provides a good hint at detecting the absolute merit of the solvers.

Figure 3 shows the plots with the results of the SBT-stability measure for each aggregation procedure considering the noise model and YASMv2 (the layout is the same as Figures 1 and 2). The x-axis reports the name of the solver s used to compute the solver-biased test set Γ_s and the y-axis reports the score value. For each of the Γ_s's, we report eight bars showing the scores obtained by the solvers using only the instances in Γ_s. The order of the bars (and of the legend) corresponds to the ranking obtained with the given aggregation procedure on the original test set Γ. As we can see from Figure 3 (top-left), CASC/QBF aggregation procedures are not SBT-stable: for each of the Γ_s, the original ranking is perturbed and the winner becomes s. Notice that on Γ_{QUANTOR}, CASC/QBF yield the same ranking that they output on the complete test set Γ. The SAT competition procedure (Figure 3, top-center) is not SBT-stable, not even on the test set biased on its alleged winner QUANTOR. YASMv2 is better than both CASC/QBF and SAT, since its alleged winner QUANTOR is the winner on

	CASC/QBF	SAT	YASM	YASMv2	Borda	r. v.	Schulze
OPENQBF	0.43	0.57	0.36	0.64	0.79	0.79	0.79
QBFBDD	0.43	0.43	0.36	0.64	0.79	0.86	0.79
QMRES	0.64	0.86	0.76	0.79	0.71	0.86	0.79
QUANTOR	1	0.86	0.86	0.86	0.93	0.86	0.93
SEMPROP	0.93	0.71	0.71	0.79	0.93	0.86	0.93
SSOLVE	0.71	0.57	0.57	0.79	0.86	0.79	0.86
WALKQSAT	0.57	0.57	0.43	0.71	0.64	0.79	0.79
YQUAFFLE	0.71	0.64	0.57	0.71	0.86	0.86	0.93
Mean	0.68	0.65	0.58	0.74	0.81	0.83	0.85

Table 3: Kendall coefficient between the ranking obtained on the original test set and each of the rankings obtained on the solver-biased test sets.

biased test sets as well. Borda's method (Figure 3, bottom-left) is not SBT-stable with respect to any solver, but the alleged winner (QUANTOR) is always the winner on the biased test sets. Moreover, the rankings obtained on the test sets biased on QUANTOR and SEMPROP are not far from the ranking obtained on the original test set. Also range voting (Figure 3, bottom-center), is not SBT-stable with respect to any solver, but the solvers ranking first and last do not change over the biased test sets and it is true for the Schulze's method (Figure 3, bottom-right) too.

Looking at the results presented above, we can see that YASMv2 performance in terms of SBT stability lies in between classical automated reasoning contests methods and methods based on voting systems. This fact is highlighted in Table 3, where for each procedure we compute the Kendall coefficient between the ranking obtained on the original test set Γ and each of the rankings obtained on the Γ_s test sets, including the mean coefficient observed. Overall, YASMv2 turns out to be, on average, better than CASC/QBF, SAT, and YASM, while it is worse, on average, than the methods based on voting systems. However, if we consider also the results of Table 2 about fidelity, we can see that YASMv2 offers the best compromise between SBT-stability and fidelity. Indeed, while CASC/QBF methods have a relatively high fidelity, they perform poorly in terms of SBT-stability, and SAT method is worse than YASMv2 both in terms of fidelity and in terms of SBT-stability. Methods based on voting systems are all more SBT-stable that YASMv2, but they have poor fidelity coefficients. We consider this good performance of YASMv2 a result of our choice to hybridize classical methods used in automated reasoning contests and methods based on voting systems. This helped us to obtain an aggregation procedure which is less sensitive to bias, and, at the same time, a good indicator of the absolute merit of the competitors.

4.5 SOTA-relevance

This measure was introduced in [17] to understand the relationship between the ranking obtained with an aggregation procedure and the strength of a solver, as witnessed by its contribution to the SOTA solver. As mentioned in Section 3, the SOTA solver is the ideal solver that always fares the best time among all the participants. Indeed, a participant contributes to the SOTA solver whenever it is the fastest solver on some instance. In [17] SOTA-relevance was obtained by counting the number of such events for any given solver, and then computing the Kendall coefficient between the ranking

	SOTA-distance
CASC	1
QBF	1
SAT	0.71
YASM	0.86
YASM v2	0.79
Borda	0.86
range voting	0.71
Schulze	0.86

Table 4: SOTA-relevance.

thereby induced and the ranking obtained with any given procedure. However, it turns out that evaluating the SOTA-contribution of each solver by simply counting the number of times that it is faster than other solvers can be misleading. To understand this, consider the following example. Suppose that a solver A solves 50% of the test set using time *at most* t_A and times out on the rest, and that solver B, on the contrary, solves all the problems where A times out using time *at most* t_B but it does time out on the problems that A solves. Finally, suppose that a solver C is able to solve all the problems in the test set using time *at least* t_C where $t_C > t_A$ and $t_C > t_B$. Given our definition of SOTA solver, it turns out that C is never contributing to it. Evaluating the SOTA contribution using a simple count as described in [17] would induce a ranking where C is last. However, C is, on average, better than both A and B and this will probably be correctly spotted by high-fidelity methods, which would turn out to have a very low SOTA-relevance.

In order to overcome the above problem we redefine here SOTA-relevance in terms of SOTA-distance. SOTA-distance is the distance metric obtained by computing the Euclidean norm between the CPU times of any given solver and the SOTA solver. The resulting values of the metrics induce a ranking that can be used to compute the Kendall coefficient yielding the SOTA-relevance. Table 4 shows the values of the coefficients thereby obtained for each procedure. Notice that according to our new definition of SOTA-relevance, CASC/QBF methods turn out to have the highest such relevance possible, i.e., $\tau = 1$. Therefore the other coefficients correspond to the first row of Table 1 about homogeneity results. Notice that YASMv2 has a better SOTA relevance than SAT and range voting, but worse than all the other methods, including YASM. Given the positive results of YASMv2 insofar fidelity and SBT-stability are concerned, we consider this result as a matter for further investigation either in the quality of YASMv2, or in the explanatory power of the SOTA-distance metric.

5 Conclusions

Summing up, the analysis presented in this paper allowed us to make some progress in the research agenda associated to QBFEVAL. Indeed, in [17] improving YASM was cited as one of the future directions, and in this paper we have presented YASMv2, which features a simpler calculation, yet it is more powerful than YASM in terms of SBT-stability and fidelity. Our empirical evaluation tools of aggregation procedures have also improved with the addition of the fidelity measure and the improved def-

inition of SOTA-relevance. We confirmed some of the conclusions reached in [17], namely that independently of the specific procedure used, a larger test set is not necessarily a better test set, and that a higher time limit does not necessarily result in a more informative contest. On the other hand, while aggregation procedures based on voting systems emerged from [17] as "moral" winners over other procedures, the analysis presented in this paper shows that better results could be achieved using hybrid techniques such as YASMv2.

References

[1] G. Sutcliffe and C. Suttner. The CADE ATP System Competition. `http://www.cs.miami.edu/~tptp/CASC` [2006-6-2].

[2] D. Le Berre and L. Simon. The SAT Competition. `http://www.satcompetition.org` [2006-6-2].

[3] D. Long and M. Fox. The 3rd International Planning Competition: Results and Analysis. *Artificial Intelligence Research*, 20:1–59, 2003.

[4] M.R.C. van Dongen. Introduction to the Solver Competition. In *CPAI 2005 proceedings*, 2005.

[5] C. W. Barrett, L. de Moura, and A. Stump. SMT-COMP: Satisfiability Modulo Theories Competition. In *CAV*, volume 3576 of *Lecture Notes in Computer Science*, pages 20–23, 2005.

[6] D. Le Berre, L. Simon, and A. Tacchella. Challenges in the QBF arena: the SAT'03 evaluation of QBF solvers. In *Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, volume 2919 of *Lecture Notes in Computer Science*. Springer Verlag, 2003.

[7] D. Le Berre, M. Narizzano, L. Simon, and A. Tacchella. The second QBF solvers evaluation. In *Seventh International Conference on Theory and Applications of Satisfiability Testing (SAT 2004)*, Lecture Notes in Computer Science. Springer Verlag, 2004.

[8] M. Narizzano, L. Pulina, and A. Tacchella. The third QBF solvers comparative evaluation. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:145–164, 2006. Available on-line at `http://jsat.ewi.tudelft.nl/`.

[9] J. N. Hooker. Testing Heuristics: We Have It All Wrong. *Journal of Heuristics*, 1:33–42, 1996.

[10] D. G. Saari. *Chaotic Elections! A Mathematician Looks at Voting*. American Mathematical Society, 2001.

[11] W. D. Smith. Range voting. Available on-line at`http://www.math.temple.edu/~wds/homepage/rangevote.pdf` [2006-9-29].

[12] M. Schulze. A New Monotonic and Clone-Independent Single-Winner Election Method. *Voting Matters*, pages 9–19, 2003.

[13] M. Narizzano, L. Pulina, and A. Taccchella. QBF solvers competitive evaluation (QBFEVAL). `http://www.qbflib.org/qbfeval`.

[14] RoboCup. `http://www.robocup.org`.

[15] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1. Elsevier, 2002.

[16] V. Conitzer and T. Sandholm. Common Voting Rules as Maximum Likelihood Estimators. In *6th ACM Conference on Electronic Commerce (EC-05)*, Lecture Notes in Computer Science, pages 78–87, 2005.

[17] L. Pulina. Empirical Evaluation of Scoring Methods. In *Proc. STAIRS 2006*, volume 142 of *Frontiers in Artificial Intelligence and Applications*, pages 108–119, 2006.

[18] M. Narizzano, L. Pulina, and A. Tacchella. Competitive Evaluation of QBF Solvers: noisy data and scoring methods. Technical report, STAR-Lab - University of Genoa, May 2006.

[19] E. Giunchiglia, M. Narizzano, and A. Tacchella. Quantified Boolean Formulas satisfiability library (QBFLIB), 2001. `www.qbflib.org`.

[20] A. Van Gelder, D. Le Berre, A. Biere, O. Kullmann, and L. Simon. Purse-Based Scoring for Comparison of Exponential-Time Programs, 2006. Unpublished draft.

[21] M. Narizzano, L. Pulina, and A. Tacchella. Competitive Evaluation of Automated Reasoning Tools: Statistical Testing and Empirical Scoring. In *First Workshop on Empirical Methods for the Analysis of Algorithms (EMAA 2006)*, 2006.

[22] M. Schulze. Extending schulze's method to obtain an overall ranking. Personal communications.

Massimo Narizzano
DIST, Università di Genova
Viale Causa, 13 – 16145 Genova, Italy
Email: `mox@dist.unige.it`

Luca Pulina
DIST, Università di Genova
Viale Causa, 13 – 16145 Genova, Italy
Email: `pulina@dist.unige.it`

Armando Tacchella
DIST, Università di Genova
Viale Causa, 13 – 16145 Genova, Italy
Email: `tac@dist.unige.it`

Bicriteria Models for Fair Resource Allocation[1]

Włodzimierz Ogryczak

Abstract

Resource allocation problems are concerned with the allocation of limited resources among competing activities so as to achieve the best performances. In systems which serve many users, like in networking, there is a need to respect some fairness rules while looking for the overall efficiency. The so-called Max-Min Fairness is widely used to meet these goals. However, allocating the resource to optimize the worst performance may cause a dramatic worsening of the overall system efficiency. Therefore, several other fair allocation schemes are searched and analyzed. In this paper we show how the scalar inequality measures can be consistently used in bicriteria models to search for fair and efficient allocations.

1 Introduction

Resource allocation problems are concerned with the allocation of limited resources among competing activities [4]. In this paper, we focus on approaches that, while allocating resources to maximize the system efficiency, they also attempt to provide a fair treatment of all the competing activities [8]. The problems of efficient and fair resource allocation arise in various systems which serve many users, like in telecommunication systems among others. In networking a central issue is how to allocate bandwidth to flows efficiently and fairly [3, 18]. In location analysis of public services, the decisions often concern the placement of a service center or another facility in a position so that the users are treated fairly in an equitable way, relative to certain criteria citeogr00. Recently, several research publications relating the fairness and equity concepts to the multiple criteria optimization methodology have appeared [7, 8, 14].

The generic resource allocation problem may be stated as follows. Each activity is measured by an individual performance function that depends on the corresponding resource level assigned to that activity. A larger function value is considered better, like the performance measured in terms of quality level, capacity, service amount available, etc. Models with an (aggregated) objective function that maximizes the mean (or simply the sum) of individual performances are widely used to formulate resource allocation problems, thus defining the so-called mean solution concept. This solution concept is primarily concerned with the overall system efficiency. As based on averaging, it often

[1]This work was partially supported by the Ministry of Science and Information Society Technologies under grant 3T11C 005 27 "Models and Algorithms for Efficient and Fair Resource Allocation in Complex Systems".

provides solution where some smaller services are discriminated in terms of allocated resources. An alternative approach depends on the so-called Max-Min solution concept, where the worst performance is maximized. The Max-Min approach is consistent with Rawlsian [20] theory of justice, especially when additionally regularized with the lexicographic order. The latter is called the Max-Min Fairness (MMF) and commonly used in networking [18]. Allocating the resources to optimize the worst performances may cause, however, a large worsening of the overall (mean) performances. Therefore, there is a need to seek a compromise between the two extreme approaches discussed above.

Fairness is, essentially, an abstract socio-political concept that implies impartiality, justice and equity [19, 24], Nevertheless, fairness was frequently quantified with the so-called inequality measures to be minimized [1, 21, 22]. Unfortunately, direct minimization of typical inequality measures contradicts the maximization of individual outcomes and it may lead to inferior decisions. The concept of fairness has been studied in various areas beginning from political economics problems of fair allocation of consumption bundles [2, 17, 19] to abstract mathematical formulation [23]. In order to ensure fairness in a system, all system entities have to be equally well provided with the system's services. This leads to concepts of fairness expressed by the equitable efficiency [6, 8, 16]. The concept of equitably efficient solution is a specific refinement of the Pareto-optimality taking into account the inequality minimization according to the Pigou-Dalton approach. In this paper the use of scalar inequality measures in bicriteria models to search for fair and efficient allocations is analyzed. There is shown that properties of convexity and positive homogeneity together with some boundedness condition are sufficient for a typical inequality measure to guarantee that it can be used consistently with the equitable optimization rules.

2 Equity and fairness

The generic resource allocation problem may be stated as follows. There is a system dealing with a set I of m services. There is given a measure of services realization within a system. In applications we consider, the measure usually expresses the service quality. In general, outcomes can be measured (modeled) as service time, service costs, service delays as well as in a more subjective way. There is also given a set Q of allocation patterns (allocation decisions). For each service $i \in I$ a function $f_i(\mathbf{x})$ of the allocation pattern $\mathbf{x} \in Q$ has been defined. This function, called the individual objective function, measures the outcome (effect) $y_i = f_i(\mathbf{x})$ of allocation $\mathbf{x}$ pattern for service i. In typical formulations a larger value of the outcome means a better effect (higher service quality or client satisfaction). Otherwise, the outcomes can be replaced with their complements to some large number. Therefore, without loss of generality, we can assume that each individual outcome y_i is to be maximized which allows us to view the generic resource allocation problem as a vector maximization model:

$$\max \{\mathbf{f}(\mathbf{x}) \ : \ \mathbf{x} \in Q\} \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is a vector-function that maps the decision space $X = R^n$ into the criterion space $Y = R^m$, and $Q \subset X$ denotes the feasible set.

Model (1) only specifies that we are interested in maximization of all objective functions f_i for $i \in I = \{1, 2, \ldots, m\}$. In order to make it operational, one needs to assume some solution concept specifying what it means to maximize multiple objective functions. The solution concepts may be defined by properties of the corresponding preference model. The preference model is completely characterized by the relation of weak preference, denoted hereafter with $\succeq$. Namely, the corresponding relations of strict preference $\succ$ and indifference $\cong$ are defined by the following formulas:

$$\begin{aligned} \mathbf{y}' \succ \mathbf{y}'' \quad &\Leftrightarrow \quad (\mathbf{y}' \succeq \mathbf{y}'' \quad \text{and} \quad \mathbf{y}'' \not\succeq \mathbf{y}'), \\ \mathbf{y}' \cong \mathbf{y}'' \quad &\Leftrightarrow \quad (\mathbf{y}' \succeq \mathbf{y}'' \quad \text{and} \quad \mathbf{y}'' \succeq \mathbf{y}'). \end{aligned}$$

The standard preference model related to the Pareto-optimal (efficient) solution concept assumes that the preference relation $\succeq$ is *reflexive*:

$$\mathbf{y} \succeq \mathbf{y}, \tag{2}$$

transitive:

$$(\mathbf{y}' \succeq \mathbf{y}'' \quad \text{and} \quad \mathbf{y}'' \succeq \mathbf{y}''') \quad \Rightarrow \quad \mathbf{y}' \succeq \mathbf{y}''', \tag{3}$$

and *strictly monotonic*:

$$\mathbf{y} + \varepsilon \mathbf{e}_i \succ \mathbf{y} \quad \text{for } \varepsilon > 0;\ i = 1, \ldots, m, \tag{4}$$

where $\mathbf{e}_i$ denotes the i–th unit vector in the criterion space. The last assumption expresses that for each individual objective function more is better (maximization). The preference relations satisfying axioms (2)–(4) are called hereafter *rational preference relations*. The rational preference relations allow us to formalize the Pareto-optimality (efficiency) concept with the following definitions. We say that outcome vector $\mathbf{y}'$ rationally dominates $\mathbf{y}''$ ($\mathbf{y}' \succ_r \mathbf{y}''$), iff $\mathbf{y}' \succ \mathbf{y}''$ for all rational preference relations $\succeq$. We say that feasible solution $\mathbf{x} \in Q$ is a *Pareto-optimal (efficient)* solution of the multiple criteria problem (1), iff $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is rationally nondominated.

Simple solution concepts for multiple criteria problems are defined by aggregation (or utility) functions $g : Y \to R$ to be maximized. Thus the multiple criteria problem (1) is replaced with the maximization problem

$$\max \{g(\mathbf{f}(\mathbf{x})) \ : \ \mathbf{x} \in Q\} \tag{5}$$

In order to guarantee the consistency of the aggregated problem (5) with the maximization of all individual objective functions in the original multiple criteria problem (or Pareto-optimality of the solution), the aggregation function must be strictly increasing with respect to every coordinate.

The simplest aggregation functions commonly used for the multiple criteria problem (1) are defined as the mean (average) outcome

$$\mu(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} y_i \tag{6}$$

or the worst outcome

$$M(\mathbf{y}) = \min_{i=1,\ldots,m} y_i. \tag{7}$$

The mean (6) is a strictly increasing function while the minimum (7) is only nondecreasing. Therefore, the aggregation (5) using the sum of outcomes always generates a Pareto-optimal solution while the maximization of the worst outcome may need some additional refinement. The mean outcome maximization is primarily concerned with the overall system efficiency. As based on averaging, it often provides a solution where some services are discriminated in terms of performances. On the other hand, the worst outcome maximization, ie, the so-called Max-Min solution concept is regarded as maintaining equity. Indeed, in the case of a simplified resource allocation problem with the knapsack constraints, the Max-Min solution meets the perfect equity requirement. In the general case, with possibly more complex feasible set structure, this property is not fulfilled. Nevertheless, if the perfectly equilibrated outcome vector $\bar{y}_1 = \bar{y}_2 = \ldots = \bar{y}_m$ is nondominated, then it is the unique optimal solution of the corresponding Max-Min optimization problem [13]. In other words, the perfectly equilibrated outcome vector is a unique optimal solution of the Max-Min problem if one cannot find any (possibly not equilibrated) vector with improved at least one individual outcome without worsening any others. Unfortunately, it is not a common case and, in general, the optimal set to the Max-Min aggregation may contain numerous alternative solutions including dominated ones. The Max-Min solution may be then regularized according to the Rawlsian principle of justice [20] which leads us to the lexicographic Max-Min concepts or the so-called Max-Min Fairness [9, 8].

In order to ensure fairness in a system, all system entities have to be equally well provided with the system's services. This leads to concepts of fairness expressed by the equitable rational preferences [6, 11]. First of all, the fairness requires impartiality of evaluation, thus focusing on the distribution of outcome values while ignoring their ordering. That means, in the multiple criteria problem (1) we are interested in a set of outcome values without taking into account which outcome is taking a specific value. Hence, we assume that the preference model is impartial (anonymous, symmetric). In terms of the preference relation it may be written as the following axiom

$$(y_{\pi(1)}, y_{\pi(2)}, \ldots, y_{\pi(m)}) \cong (y_1, y_2, \ldots, y_m) \quad \text{for any permutation } \pi \text{ of } I \tag{8}$$

which means that any permuted outcome vector is indifferent in terms of the preference relation. Further, fairness requires equitability of outcomes which causes that the preference model should satisfy the (Pigou–Dalton) principle of transfers. The principle of transfers states that a transfer of any small amount from an outcome to any other relatively worse–off outcome results in a more preferred outcome vector. As a property of the preference relation, the principle of transfers takes the form of the following axiom

$$y_{i'} > y_{i''} \quad \Rightarrow \quad \mathbf{y} - \varepsilon\mathbf{e}_{i'} + \varepsilon\mathbf{e}_{i''} \succ \mathbf{y} \quad \text{for } 0 < \varepsilon < y_{i'} - y_{i''} \tag{9}$$

The rational preference relations satisfying additionally axioms (8) and (9) are called hereafter *fair (equitable) rational preference relations*. We say that outcome vector $\mathbf{y}'$ *fairly (equitably) dominates* $\mathbf{y}''$ ($\mathbf{y}' \succ_e \mathbf{y}''$), iff $\mathbf{y}' \succ \mathbf{y}''$ for all fair rational preference relations $\succeq$. In other words, $\mathbf{y}'$ fairly dominates $\mathbf{y}''$, if there exists a finite sequence of vectors $\mathbf{y}^j$ $(j = 1, 2, \ldots, s)$ such that $\mathbf{y}^1 = \mathbf{y}''$, $\mathbf{y}^s = \mathbf{y}'$ and $\mathbf{y}^j$ is constructed from $\mathbf{y}^{j-1}$ by application of either permutation of coordinates, equitable transfer, or increase of a coordinate. An allocation pattern $\mathbf{x} \in Q$ is called *fairly (equitably) efficient* or simply *fair* if $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is fairly nondominated. Note that each fairly efficient solution is also Pareto-optimal, but not vice verse.

In order to guarantee fairness of the solution concept (5), additional requirements on the class of aggregation (utility) functions must be introduced. In particular, the aggregation function must be additionally symmetric (impartial), i.e. for any permutation π of I,

$$g(y_{\pi(1)}, y_{\pi(2)}, \ldots, y_{\pi(m)}) = g(y_1, y_2, \ldots, y_m) \tag{10}$$

as well as be equitable (to satisfy the principle of transfers)

$$g(y_1, \ldots, y_{i'} - \varepsilon, \ldots, y_{i''} + \varepsilon, \ldots, y_m) > g(y_1, y_2, \ldots, y_m) \tag{11}$$

for any $0 < \varepsilon < y_{i'} - y_{i''}$. In the case of a strictly increasing function satisfying both the requirements (10) and (11), we call the corresponding problem (5) a *fair (equitable) aggregation* of problem (1). Every optimal solution to the fair aggregation (5) of a multiple criteria problem (1) defines some fair (equitable) solution.

Note that both the simplest aggregation functions, the sum (6) and the minimum (7), are symmetric although they do not satisfy the equitability requirement (11). To guarantee the fairness of solutions, some enforcement of concave properties is required. For any strictly concave, increasing utility function $u : R \to R$, the function $g(\mathbf{y}) = \sum_{i=1}^{m} u(y_i)$ is a strictly monotonic and equitable thus defining a family of the fair aggregations. Various concave utility functions u can be used to define such fair solution concepts. In the case of the outcomes restricted to positive values, one may use logarithmic function thus resulting in the *Proportional Fairness* (PF) solution concept [5]. Actually, it corresponds to the so-called Nash criterion which maximizes the product of additional utilities compared to the status quo. For a common case of upper bounded outcomes $y_i \leq y^*$ one may maximize power functions $-\sum_{i=1}^{m} (y^* - y_i)^p$ for $1 < p < \infty$ which corresponds to the minimization of the corresponding p-norm distances from the common upper bound y^* [7].

Fig. 1 presents the structure of fair dominance for two-dimensional outcome vectors. For any outcome vector $\bar{\mathbf{y}}$, the fair dominance relation distinguishes set $D(\bar{\mathbf{y}})$ of dominated outcomes (obviously worse for all fair rational preferences) and set $S(\bar{\mathbf{y}})$ of dominating outcomes (obviously better for all fair rational preferences). However, some outcome vectors are left (in white areas) and they can be differently classified by various specific fair rational preferences.

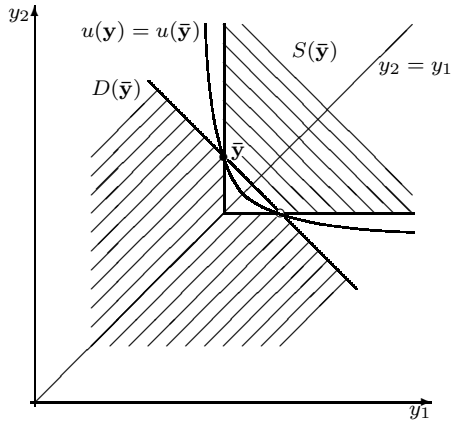


Figure 1: Structure of the fair dominance: $D(\bar{\mathbf{y}})$ – the set fairly dominated by $\bar{\mathbf{y}}$, $S(\bar{\mathbf{y}})$ – the set of outcomes fairly dominating $\bar{\mathbf{y}}$.

The MMF fairness assigns the entire interior of the inner white triangle to the set of preferred outcomes while classifying the interior of the external open triangles as worse outcomes. Isolines of various utility functions split the white areas in different ways. One may notice that the set $D(\bar{\mathbf{y}})$ of directions leading to outcome vectors being dominated by a given $\bar{\mathbf{y}}$ is, in general, not a cone and it is not convex. Although, when we consider the set $S(\bar{\mathbf{y}})$ of directions leading to outcome vectors dominating given $\bar{\mathbf{y}}$ we get a convex set.

3 Inequality measures and fair consistency

Inequality measures were primarily studied in economics [22] while recently they become very popular tools in Operations Research. Typical inequality measures are some deviation type dispersion characteristics. They are *translation invariant* in the sense that $\varrho(\mathbf{y} + a\mathbf{e}) = \varrho(\mathbf{y})$ for any outcome vector $\mathbf{y}$ and real number a (where $\mathbf{e}$ vector of units $(1, \ldots, 1)$), thus being not affected by any shift of the outcome scale. Moreover, the inequality measures are also *inequality relevant* which means that they are equal to 0 in the case of perfectly equal outcomes while taking positive values for unequal ones.

The simplest inequality measures are based on the absolute measurement of the spread of outcomes, like the *mean absolute difference*

$$\Gamma(\mathbf{y}) = \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |y_i - y_j| \tag{12}$$

or the *maximum absolute difference*

$$d(\mathbf{y}) = \max_{i,j=1,\ldots,m} |y_i - y_j|. \tag{13}$$

In most application frameworks better intuitive appeal may have inequality measures related to deviations from the mean outcome like the mean absolute deviation

$$\delta(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} |y_i - \mu(\mathbf{y})|. \tag{14}$$

or the *maximum absolute deviation*

$$R(\mathbf{y}) = \max_{i \in I} |y_i - \mu(\mathbf{y})|. \tag{15}$$

Note that the *standard deviation* σ (or the *variance* σ^2) represents both the deviations and the spread measurement as

$$\sigma(\mathbf{y}) = \sqrt{\frac{1}{m} \sum_{i \in I} (y_i - \mu(\mathbf{y}))^2} = \sqrt{\frac{1}{2m^2} \sum_{i \in I} \sum_{j \in I} (y_i - y_j)^2}. \tag{16}$$

Deviational measures may be focused on the downside semideviations as related to worsening of outcome while ignoring upper semideviations related to improvement of outcome. One may define the *maximum (downside) semideviation*

$$\Delta(\mathbf{y}) = \max_{i \in I} (\mu(\mathbf{y}) - y_i) \tag{17}$$

and the *mean (downside) semideviation*

$$\bar{\delta}(\mathbf{y}) = \frac{1}{m} \sum_{i \in I} (\mu(\mathbf{y}) - y_i)_+ \tag{18}$$

where $(.)_+$ denotes the nonnegative part of a number. Similarly, the *standard (downside) semideviation* is given as

$$\bar{\sigma}(\mathbf{y}) = \sqrt{\frac{1}{m} \sum_{i \in I} (\mu(\mathbf{y}) - y_i)_+^2}. \tag{19}$$

In economics one usually considers relative inequality measures normalized by mean outcome. Among many inequality measures perhaps the most commonly accepted by economists is the Gini coefficient, which is the relative mean difference. One can easily notice that direct minimization of typical inequality measures (especially the relative ones) may contradict the optimization of individual outcomes resulting in equal but very low outcomes. As some resolution one may consider a bicriteria mean-equity model:

$$\max \{(\mu(\mathbf{f}(\mathbf{x})), -\varrho(\mathbf{f}(\mathbf{x}))) \ : \ \mathbf{x} \in Q\} \tag{20}$$

which takes into account both the efficiency with optimization of the mean outcome $\mu(\mathbf{y})$ and the equity with minimization of an inequality measure $\varrho(\mathbf{y})$. For typical inequality measures bicriteria model (20) is computationally very attractive since both the criteria are concave and LP implementable for many measures. Unfortunately, for any dispersion type inequality measures the bicriteria mean-equity model is not consistent with the outcomes maximization, and therefore is not consistent with the fair dominance. When considering a simple discrete problem with two allocation patterns P1 and P2 generating outcome vectors $\mathbf{y}' = (0, 0)$ and $\mathbf{y}'' = (2, 8)$, respectively, for any dispersion type inequality measure one gets $\varrho(\mathbf{y}'') > 0 = \varrho(\mathbf{y}')$ while $\mu(\mathbf{y}'') = 5 > 0 = \mu(\mathbf{y}')$. Hence, $\mathbf{y}''$ is not bicriteria dominated by $\mathbf{y}'$ and vice versa. Thus for any dispersion type inequality measure ϱ, allocation P1 with obviously worse outcome vector than that for allocation P2 is a Pareto-optimal solution in the corresponding bicriteria mean-equity model (20).

Note that the lack of consistency of the mean-equity model (20) with the outcomes maximization applies also to the case of the maximum semideviation $\Delta(\mathbf{y})$ (17) used as an inequality measure whereas subtracting this measure from the mean $\mu(\mathbf{y}) - \Delta(\mathbf{y}) = M(\mathbf{y})$ results in the worst outcome and thereby the first criterion of the MMF model. In other words, although a direct use of the maximum semideviation in the mean-equity model may contradict the outcome maximization, the measure can be used complementary to the mean leading us to the worst outcome criterion which does not contradict the outcome maximization. This construction can be generalized for various (dispersion type) inequality measures. Moreover, we allow the measures to be scaled with any positive factor $\alpha > 0$, in order to avoid creation of new inequality measures as one could consider $\varrho_\alpha(X) = \alpha\varrho(X)$ as a different inequality measure. For any inequality measure ϱ we introduce the corresponding underachievement function defined as the difference of the mean outcome and the (scaled) inequality measure itself, i.e.

$$M_{\alpha\varrho}(\mathbf{y}) = \mu(\mathbf{y}) - \alpha\varrho(\mathbf{y}). \tag{21}$$

This allows us to replace the original mean-equity bicriteria optimization (20) with the following bicriteria problem:

$$\max\{(\mu(\mathbf{f}(\mathbf{x})), \mu(\mathbf{f}(\mathbf{x})) - \alpha\varrho(\mathbf{f}(\mathbf{x}))) \ : \ \mathbf{x} \in Q\} \tag{22}$$

where the second objective represents the corresponding underachievement measure $M_{\alpha\varrho}(\mathbf{y})$ (21). Note that for any inequality measure $\varrho(\mathbf{y}) \geq 0$ one gets $M_{\alpha\varrho}(\mathbf{y}) \leq \mu(\mathbf{y})$ thus really expressing underachievements (comparing to mean) from the perspective of outcomes being maximized.

We will say that an inequality measure ϱ is *fairly α-consistent* if

$$\mathbf{y}' \succeq_e \mathbf{y}'' \quad \Rightarrow \quad \mu(\mathbf{y}') - \alpha\varrho(\mathbf{y}') \geq \mu(\mathbf{y}'') - \alpha\varrho(\mathbf{y}'') \tag{23}$$

The relation of fair α-consistency will be called *strong* if, in addition to (23), the following holds

$$\mathbf{y}' \succ_e \mathbf{y}'' \quad \Rightarrow \quad \mu(\mathbf{y}') - \alpha\varrho(\mathbf{y}') > \mu(\mathbf{y}'') - \alpha\varrho(\mathbf{y}''). \tag{24}$$

THEOREM 1 *If the inequality measure $\varrho(\mathbf{y})$ is fairly α-consistent (23), then except for outcomes with identical values of $\mu(\mathbf{y})$ and $\varrho(\mathbf{y})$, every efficient solution of the bicriteria problem (22) is a fairly efficient allocation pattern. In the case of strong consistency (24), every allocation pattern $\mathbf{x} \in Q$ efficient to (22) is, unconditionally, fairly efficient.*

Proof. Let $\mathbf{x}^0 \in Q$ be an efficient solution of (22). Suppose that $\mathbf{x}^0$ is not fairly efficient. This means, there exists $\mathbf{x} \in Q$ such that $\mathbf{y} = \mathbf{f}(\mathbf{x}) \succ_e \mathbf{y}^0 = \mathbf{f}(\mathbf{x}^0)$. Then, it follows $\mu(\mathbf{y}) \geq \mu(\mathbf{y}^0)$, and simultaneously $\mu(\mathbf{y}) - \alpha\varrho(\mathbf{y}) \geq \mu(\mathbf{y}^0) - \alpha\varrho(\mathbf{y}^0)$, by virtue of the fair α-consistency (23). Since $\mathbf{x}^0$ is efficient to (22) no inequality can be strict, which implies $\mu(\mathbf{y}) = \mu(\mathbf{y}^0)$ and and $\varrho(\mathbf{y}) = \varrho(\mathbf{y}^0)$.

In the case of the strong fair α-consistency (24), the supposition $\mathbf{y} = \mathbf{f}(\mathbf{x}) \succ_e \mathbf{y}^0 = \mathbf{f}(\mathbf{x}^0)$ implies $\mu(\mathbf{y}) \geq \mu(\mathbf{y}^0)$ and $\mu(\mathbf{y}) - \alpha\varrho(\mathbf{y}) > \mu(\mathbf{y}^0) - \alpha\varrho(\mathbf{y}^0)$ which contradicts the efficiency of $\mathbf{x}^0$ with respect to (22). Hence, the allocation pattern $\mathbf{x}^0$ is fairly efficient. □

4 Fair consistency conditions

Typical dispersion type inequality measures are convex, i.e. $\varrho(\lambda\mathbf{y}' + (1-\lambda)\mathbf{y}'') \leq \lambda\varrho(\mathbf{y}') + (1-\lambda)\varrho(\mathbf{y}'')$ for any $\mathbf{y}', \mathbf{y}''$ and $0 \leq \lambda \leq 1$. Certainly, the underachievement function $M_{\alpha\varrho}(\mathbf{y})$ must be also monotonic for the fair consistency which enforces more restrictions on the inequality measures. We will show further that convexity together with positive homogeneity and some boundedness of an inequality measure is sufficient to guarantee monotonicity of the corresponding underachievement measure and thereby to guarantee the fair α-consistency of inequality measure itself.

We say that (dispersion type) inequality measure $\varrho(\mathbf{y}) \geq 0$ is Δ-*bounded* if it is upper bounded by the maximum downside deviation, i.e.,

$$\varrho(\mathbf{y}) \leq \Delta(\mathbf{y}) \quad \forall \mathbf{y}. \tag{25}$$

Moreover, we say that $\varrho(\mathbf{y}) \geq 0$ is strictly Δ-bounded if inequality (25) is a strict bound, except from the case of perfectly equal outcomes, i.e., $\varrho(\mathbf{y}) < \Delta(\mathbf{y})$ for any $\mathbf{y}$ such that $\Delta(\mathbf{y}) > 0$.

THEOREM 2 *Let $\varrho(\mathbf{y}) \geq 0$ be a convex, positively homogeneous and translation invariant (dispersion type) inequality measure. If $\alpha\varrho(\mathbf{y})$ is Δ-bounded, then $\varrho(\mathbf{y})$ is fairly α-consistent in the sense of (23).*

Proof. The relation of fair dominance $\mathbf{y}' \succeq_e \mathbf{y}''$ denotes that there exists a finite sequence of vectors $\mathbf{y}^0 = \mathbf{y}'', \mathbf{y}^1, \ldots, \mathbf{y}^t$ such that $\mathbf{y}^k = \mathbf{y}^{k-1} - \varepsilon_k \mathbf{e}_{i'} + \varepsilon_k \mathbf{e}_{i''}$, $0 \leq \varepsilon_k \leq y_{i'}^{k-1} - y_{i''}^{k-1}$ for $k = 1, 2, \ldots, t$ and there exists a permutation π such that $y'_{\pi(i)} \geq y_i^t$ for all $i \in I$. Note that the underachievement function $M_{\alpha\varrho}(\mathbf{y})$, similar as $\varrho(\mathbf{y})$ depends only on the distribution of outcomes. Further, if $\mathbf{y}' \geq \mathbf{y}''$, then $\mathbf{y}' = \mathbf{y}'' + (\mathbf{y}' - \mathbf{y}'')$ and $\mathbf{y}' - \mathbf{y}'' \geq 0$. Hence, due to concavity

and positive homogeneity, $M_{\alpha\varrho}(\mathbf{y}') \geq M_{\alpha\varrho}(\mathbf{y}'') + M_{\alpha\varrho}(\mathbf{y}' - \mathbf{y}'')$. Moreover, due to the bound (25), $M_{\alpha\varrho}(\mathbf{y}' - \mathbf{y}'') \geq \mu(\mathbf{y}' - \mathbf{y}'') - \Delta(\mathbf{y}' - \mathbf{y}'') \geq \mu(\mathbf{y}' - \mathbf{y}'') - \mu(\mathbf{y}' - \mathbf{y}'') = 0$. Thus, $M_{\alpha\varrho}(\mathbf{y})$ satisfies also the requirement of monotonicity. Hence, $M_{\alpha\varrho}(\mathbf{y}') \geq M_{\alpha\varrho}(\mathbf{y}^t)$. Further, let us notice that $\mathbf{y}^k = \lambda\bar{\mathbf{y}}^{k-1} + (1-\lambda)\mathbf{y}^{k-1}$ where $\bar{\mathbf{y}}^{k-1} = \mathbf{y}^{k-1} - (y_{i'} - y_{i''})\mathbf{e}_{i'} + (y_{i'} - y_{i''})\mathbf{e}_{i''}$ and $\lambda = \varepsilon/(y_{i'} - y_{i''})$. Vector $\bar{\mathbf{y}}^{k-1}$ has the same distribution of coefficients as $\mathbf{y}^{k-1}$ (actually it represents results of swapping $y_{i'}$ and $y_{i''}$). Hence, due to concavity of $M_{\alpha\varrho}(\mathbf{y})$, one gets $M_{\alpha\varrho}(\mathbf{y}^k) \geq \lambda M_{\alpha\varrho}(\bar{\mathbf{y}}^{k-1}) + (1-\lambda)M_{\alpha\varrho}(\mathbf{y}^{k-1}) = M_{\alpha\varrho}(\mathbf{y}^{k-1})$. Thus, $M_{\alpha\varrho}(\mathbf{y}') \geq M_{\alpha\varrho}(\mathbf{y}'')$ which justifies the fair α-consistency of $\varrho(\mathbf{y})$. □

For strong fair α-consistency some strict monotonicity and concavity properties of the underachievement function are needed. Obviously, there does not exist any inequality measure which is positively homogeneous and simultaneously strictly convex. However, one may notice from the proof of Theorem 2 that only convexity properties on equally distributed outcome vectors are important for monotonous underachievement functions.

We say that inequality measure $\varrho(\mathbf{y}) \geq 0$ is *strictly convex on equally distributed outcome vectors*, if

$$\varrho(\lambda\mathbf{y}' + (1-\lambda)\mathbf{y}'') < \lambda\varrho(\mathbf{y}') + (1-\lambda)\varrho(\mathbf{y}'')$$

for $0 < \lambda < 1$ and any two vectors $\mathbf{y}' \neq \mathbf{y}''$ representing the same outcomes distribution as some $\mathbf{y}$, i.e., $\mathbf{y}' = (y_{\pi'(1)}, \ldots, y_{\pi'(m)})$ π' and $\mathbf{y}'' = (y_{\pi''(1)}, \ldots, y_{\pi''(m)})$ for some permutations π' and π'', respectively.

THEOREM 3 *Let $\varrho(\mathbf{y}) \geq 0$ be a convex, positively homogeneous and translation invariant (dispersion type) inequality measure. If $\varrho(\mathbf{y})$ is also strictly convex on equally distributed outcomes and $\alpha\varrho(\mathbf{y})$ is strictly Δ-bounded, then the measure $\varrho(\mathbf{y})$ is fairly strongly α-consistent in the sense of (24).*

Proof. The relation of weak fair dominance $\mathbf{y}' \succeq_e \mathbf{y}''$ denotes that there exists a finite sequence of vectors $\mathbf{y}^0 = \mathbf{y}'', \mathbf{y}^1, \ldots, \mathbf{y}^t$ such that $\mathbf{y}^k = \mathbf{y}^{k-1} - \varepsilon_k\mathbf{e}_{i'} + \varepsilon_k\mathbf{e}_{i''}$, $0 \leq \varepsilon_k \leq y_{i'}^{k-1} - y_{i''}^{k-1}$ for $k = 1, 2, \ldots, t$ and there exists a permutation π such that $y'_{\pi(i)} \geq y_i^t$ for all $i \in I$. The strict fair dominance $\mathbf{y}' \succ_e \mathbf{y}''$ means that $y'_{\pi(i)} > y_i^t$ for some $i \in I$ or at least one ε_k is strictly positive. Note that the underachievement function $M_{\alpha\varrho}(\mathbf{y})$ is strictly monotonous and strictly convex on equally distributed outcome vectors. Hence, $M_{\alpha\varrho}(\mathbf{y}') > M_{\alpha\varrho}(\mathbf{y}'')$ which justifies the fair strong α-consistency of the measure $\varrho(\mathbf{y})$. □

The specific case of fair 1-consistency is also called *the mean-complementary fair consistency*. Note that the fair $\bar{\alpha}$-consistency of measure $\varrho(\mathbf{y})$ actually guarantees the mean-complementary fair consistency of measure $\alpha\varrho(\mathbf{y})$ for all $0 < \alpha \leq \bar{\alpha}$, and the same remain valid for the strong consistency properties. It follows from a possible expression of $\mu(\mathbf{y}) - \alpha\varrho(\mathbf{y})$ as the convex combination of $\mu(\mathbf{y}) - \bar{\alpha}\varrho(\mathbf{y})$ and $\mu(\mathbf{y})$. Hence, for any $\mathbf{y}' \succeq_e \mathbf{y}''$, due to $\mu(\mathbf{y}') \geq \mu(\mathbf{y}'')$ one gets $\mu(\mathbf{y}') - \alpha\varrho(\mathbf{y}') \geq \mu(\mathbf{y}'') - \alpha\varrho(\mathbf{y}'')$ in the case of the fair $\bar{\alpha}$-consistency of measure $\varrho(\mathbf{y})$ (or respective strict inequality in the case of strong consistency). Therefore, while analyzing specific inequality measures we seek the largest values α guaranteeing the corresponding fair consistency.

As mentioned, typical inequality measures are convex and many of them are positively homogeneous. Moreover, the measures such as the mean absolute (downside) semideviation $\bar{\delta}(\mathbf{y})$ (18), the standard downside semideviation $\bar{\sigma}(\mathbf{y})$ (19), and the mean absolute difference $\Gamma(\mathbf{y})$ (12) are Δ-bounded. Indeed, one may easily notice that $y_i - \mu(\mathbf{y}) \leq \Delta(\mathbf{y})$ and therefore $\bar{\delta}(\mathbf{y}) \leq \frac{1}{m}\sum_{i\in I}\Delta(\mathbf{y}) = \Delta(\mathbf{y})$, $\bar{\sigma}(\mathbf{y}) \leq \sqrt{\Delta(\mathbf{y})^2} = \Delta(\mathbf{y})$ and $\Gamma(\mathbf{y}) = \frac{1}{m^2}\sum_{i\in I}\sum_{j\in I}(\max\{y_i, y_j\} - \mu(\mathbf{y})) \leq \Delta(\mathbf{y})$. Actually, all these inequality measures are strictly Δ-bounded since for any unequal outcome vector at least one outcome must be below the mean thus leading to strict inequalities in the above bounds. Obviously, Δ-bounded (but not strictly) is also the maximum absolute downside deviation $\Delta(\mathbf{y})$ itself. This allows us to justify the maximum downside deviation $\Delta(\mathbf{y})$ (17), the mean absolute (downside) semideviation $\bar{\delta}(\mathbf{y})$ (18), the standard downside semideviation $\bar{\sigma}(\mathbf{y})$ (19) and the mean absolute difference $\Gamma(\mathbf{y})$ (12) as fairly 1-consistent (mean-complementary fairly consistent) in the sense of (23).

We emphasize that, despite the standard semideviation is a fairly 1-consistent inequality measure, the consistency is not valid for variance, semivariance and even for the standard deviation. These measures, in general, do not satisfy the all assumptions of Theorem 2. Certainly, we have enumerated only the simplest inequality measures studied in the resource allocation context which satisfy the assumptions of Theorem 2 and thereby they are fairly 1-consistent. Theorem 2 allows one to show this property for many other measures. In particular, one may easily find out that any convex combination of fairly α-consistent inequality measures remains also fairly α-consistent. On the other hand, among typical inequality measures the mean absolute difference seems to be the only one meeting the stronger assumptions of Theorem 3 and thereby maintaining the strong consistency.

As mentioned, the mean absolute semideviation is twice the mean absolute upper semideviation which means that $\alpha\delta(\mathbf{y})$ is Δ-bounded for any $0 < \alpha \leq 0.5$. The symmetry of mean absolute semideviations $\bar{\delta}(\mathbf{y}) = \sum_{i\in I}(y_i - \mu(\mathbf{y}))_+ = \sum_{i\in I}(\mu(\mathbf{y}) - y_i)_+$ can be also used to derive some Δ-boundedness relations for other inequality measures. In particular, one may find out that for m-dimensional outcome vectors of unweighted problem, any downside semideviation from the mean cannot be larger than $m-1$ upper semideviations. Hence, the maximum absolute deviation satisfies the inequality $\frac{1}{m-1}R(\mathbf{y}) \leq \Delta(\mathbf{y})$, while the maximum absolute difference fulfills $\frac{1}{m}d(\mathbf{y}) \leq \Delta(\mathbf{y})$. Similarly, for the standard deviation one gets $\frac{1}{\sqrt{m-1}}\delta(\mathbf{y}) \leq \Delta(\mathbf{y})$. Actually, $\alpha\sigma(\mathbf{y})$ is strictly Δ-bounded for any $0 < \alpha \leq 1/\sqrt{m-1}$ since for any unequal outcome vector at least one outcome must be below the mean thus leading to strict inequalities in the above bounds. These allow us to justify the mean absolute semideviation with $0 < \alpha \leq 0.5$, the maximum absolute deviation with $0 < \alpha \leq \frac{1}{m-1}$, the maximum absolute difference with $0 < \alpha \leq \frac{1}{m}$ and the standard deviation with $0 < \alpha \leq \frac{1}{\sqrt{m-1}}$ as fairly α-consistent within the specified intervals of α. Moreover, the α-consistency of the standard deviation is strong.

The fair consistency results for basic dispersion type inequality measures

Table 1: Fair consistency results for the basic inequality measures

Measure			α–consistency	
Mean absolute semideviation	$\bar{\delta}(\mathbf{y})$	(18)	1	
Mean absolute deviation	$\delta(\mathbf{y})$	(14)	0.5	
Maximum semideviation	$\Delta(\mathbf{y})$	(17)	1	
Maximum absolute deviation	$R(\mathbf{y})$	(15)	$1/(m-1)$	
Mean absolute difference	$\Gamma(\mathbf{y})$	(12)	1	strong
Maximum absolute difference	$d(\mathbf{y})$	(13)	$1/m$	
Standard semideviation	$\bar{\sigma}(\mathbf{y})$	(19)	1	
Standard deviation	$\sigma(\mathbf{y})$	(16)	$1/\sqrt{m-1}$	strong

considered in resource allocation problems are summarized in Table 1 where α values for unweighted as well as weighted problems are given and the strong consistency is indicated. Table 1 points out how the inequality measures can be used in resource allocation models to guarantee their harmony both with outcome maximization (Pareto-optimality) and with inequalities minimization (Pigou-Dalton equity theory). Exactly, for each inequality measure applied with the corresponding value α from Table 1 (or smaller positive value), every efficient solution of the bicriteria problem (22), ie. $\max\{(\mu(\mathbf{f}(\mathbf{x})), \mu(\mathbf{f}(\mathbf{x})) - \alpha\varrho(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\}$, is a fairly efficient allocation pattern, except for outcomes with identical values of $\mu(\mathbf{y})$ and $\varrho(\mathbf{y})$. In the case of strong consistency (as for mean absolute difference or standard deviation), every solution $\mathbf{x} \in Q$ efficient to (22) is, unconditionally, fairly efficient.

5 Conclusions

The problems of efficient and fair resource allocation arise in various systems which serve many users. Fairness is, essentially, an abstract socio-political concept that implies impartiality, justice and equity. Nevertheless, in operations research it was quantified with various solution concepts. The equitable optimization with the preference structure that complies with both the efficiency (Pareto-optimality) and with the Pigou-Dalton principle of transfers may be used to formalize the fair solution concepts. Multiple criteria models equivalent to equitable optimization allows to generate a variety of fair and efficient resource allocation patterns [16].

In this paper we have analyzed how scalar inequality measures can be used to guarantee the fair consistency. It turns out that several inequality measures can be combined with the mean itself into the optimization criteria generalizing the concept of the worst outcome and generating fairly consistent underachievement measures. We have shown that properties of convexity and positive homogeneity together with being bounded by the maximum downside semideviation are sufficient for a typical inequality measure to guarantee the corresponding fair consistency. It allows us to identify various inequality measures which can be effectively used to incorporate fairness factors into various resource alloca-

tion problems while preserving the consistency with outcomes maximization. Among others the standard semideviation turns out to be such a consistent inequality measure while the mean absolute difference is strongly consistent.

Our analysis is related to the properties of solutions to resource allocation models. It has been shown how inequality measures can be included into the models avoiding contradiction to the maximization of outcomes. We do not analyze algorithmic issues for the specific resource allocation problems. Generally, the requirement of convexity necessary for the consistency, guarantees that the corresponding optimization criteria belong to the class of convex optimization, not complicating the original resource allocation model with any additional discrete structure. Many of the inequality measures, we analyzed, can be implemented with auxiliary linear programming constraints. Nevertheless, further research on efficient computational algorithms for solving the specific models is necessary.

References

[1] A. B. Atkinson. On the measurement of inequality. *J. of Economic Theory*, 2:244–263, 1970.

[2] H. Dalton. The measurement of the inequality of income. *Economic Journal*, 30:348–361, 1920.

[3] R. Denda, A. Banchs, and W. Effelsberg. The fairness challenge in computer networks. *Lect. Notes Comp. Sci.*, 1922:208–220, 2000.

[4] T. Ibaraki, and N. Katoh. *Resource Allocation Problems, Algorithmic Approaches*, MIT Press, Cambridge, 1988.

[5] F. Kelly, A. Mauloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.*, 49:206–217, 1997.

[6] M. M. Kostreva, and W. Ogryczak. Linear optimization with multiple equitable criteria. *RAIRO Rech. Opér.*, 33:275–297, 1999.

[7] M. M. Kostreva, W. Ogryczak, and A. Wierzbicki. Equitable aggregations and multiple criteria analysis, *European J. of Operational Research*, 158:362–367, 2004.

[8] H. Luss. On equitable resource allocation problems: A lexicographic minimax approach. *Operations Research*, 47:361–378, 1999.

[9] E. Marchi, and J. A. Oviedo. Lexicographic optimality in the multiple objective linear programming: The nucleolar solution. *European J. of Operational Research*, 57:355–359, 1992.

[10] A. W. Marshall, and I. Olkin. *Inequalities: Theory of Majorization and Its Applications*. Academic Press, New York, 1979.

[11] W. Ogryczak. *Linear and Discrete Optimization with Multiple Criteria: Preference Models and Applications to Decision Support* (in Polish). Warsaw Univ. Press, 1997.

[12] W. Ogryczak. Inequality measures and equitable approaches to location problems, *European J. of Operational Research*, 122:374–391, 2000.

[13] W. Ogryczak. Comments on properties of the minimax solutions in goal programming. *European J. of Operational Research*, 132:17–21, 2001.

[14] W. Ogryczak, and T. Śliwiński. On equitable approaches to resource allocation problems: the conditional minimax solution, *J. of Telecommunications and Information Technology*, 3:40–48, 2002.

[15] W. Ogryczak, and T. Śliwiński. On direct methods for lexicographic minmax optimization. *Lect. Notes Comp. Sci.*, 3982:774–783, 2006.

[16] W. Ogryczak, and A. Wierzbicki. On multi-criteria approaches to bandwidth allocation. *Control and Cybernetics*, 33:427–448, 2004.

[17] A. C. Pigou. *Wealth and Welfare*. Macmillan, London, 1912.

[18] M. Pióro, and D. Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publ., 2004.

[19] J. Rawls. Justice as fairness. *Philosophical Review*, LXVII:164–194, 1958.

[20] J. Rawls. *The Theory of Justice*. Harvard Univ. Press, Cambridge, 1971.

[21] M. Rothschild, and J. E. Stiglitz. Some further results nn the measurement of inequality, *J. of Economic Theory*, 6:188–204, 1973.

[22] A. Sen. *On Economic Inequality*. Clarendon Press, Oxford, 1973.

[23] H. Steinhaus. Sur la division pragmatique. *Econometrica*, 17:315–319, 1949.

[24] H. P. Young. *Equity in Theory and Practice*. Princeton Univ. Press, 1994.

Włodzimierz Ogryczak
Warsaw University of Technology
Institute of Control and Computation Engineering
Nowowiejska 15/19
00–665 Warsaw, Poland
Email: W.Ogryczak@ia.pw.edu.pl

Some Results on *Adjusted Winner*[1]

Eric Pacuit Rohit Parikh Samer Salame

Abstract

We study the *Adjusted Winner* procedure of Brams and Taylor for dividing goods fairly between two individuals, and prove several results. In particular we show rigorously that as the differences between the two individuals become more acute they both benefit. We introduce a geometric approach which allows us to give alternate proofs of some of the Brams-Taylor results and which gives some hope for understanding the many-agent case also. We also point out that while honesty may not always be the best policy, it *is* as Parikh and Pacuit [4] point out in the context of voting, the only *safe* one. Finally, we show that provided that the assignments of valuation points are allowed to be real numbers, the final result is a continuous function of the valuations given by the two agents and suggest a generalization of the adjusted winner function to take into account nonlinear utility functions.

1 Introduction

In this paper we study one particular algorithm, or procedure, for settling a dispute between two players over a finite set of goods. The algorithm we are interested in is called *Adjusted Winner* (*AW*) and due to Steven Brams and Alan Taylor [2]. See also [1] for a relevant discussion. Suppose there are two players, called Ann (A) and Bob (B), and n (divisible[2]) goods ($G_1, \ldots, G_n$) which must be distributed to Ann and Bob. The goal of the Adjusted Winner algorithm is to *fairly* distribute the n goods between Ann and Bob. We begin by discussing an example which illustrates the Adjusted Winner algorithm.

Suppose Ann and Bob are dividing three goods: G_1, G_2, and G_3. *Adjusted Winner* begins by giving both Ann and Bob 100 points to divide among the three goods. Suppose that Ann and Bob assign these points according to the following table.

Item	Ann	Bob
G_1	$\underline{10}$	7
G_2	$\underline{65}$	43
G_3	25	$\underline{50}$
Total	100	100

[1] Working paper which has been presented at the Stony Brook International Game Theory Conference, June 2005 and Multiagent Resource Allocation Workshop (MARA), September 2005.

[2] Actually all we need to assume is that *one* good is divisible. However, since we do not know before the algorithm begins *which* good will be divided, we assume all goods are divisible. See [2, 3] for a discussion of this fact.

The first step of the procedure is to give G_1 and G_2 to Ann since she assigned more points to those items, and item G_3 to Bob. However this is not an equitable outcome since Ann has received 75 points while Bob only received 50 points (each according to their personal valuation). We must now transfer some of Ann's goods to Bob. In order to determine which goods should be transfered from Ann to Bob, we look at the ratios of Ann's valuations to Bob's valuations. For G_1 the ratio is $10/7 \approx 1.43$ and for G_2 the ratio is $65/43 \approx 1.51$. Since 1.43 is less than 1.51, we transfer as much of G_1 as needed from Ann to Bob[3] to achieve equitability.

However, even giving all of item G_1 to Bob will not create an equitable division since Ann still has 65 points, while Bob has only 57 points. In order to create equitability, we must transfer part of item G_2 from Ann to Bob. Let p be the proportion of item G_2 that Ann will keep. p should then satisfy

$$65p = 100 - 43p$$

yielding $p = 100/108 = 0.9259$, so Ann will keep 92.59% of item G_2 and Bob will get 7.41% of item G_2. Thus both Ann and Bob receive 60.185 points. It turns out that this allocation (Ann receives 92.59% of item G_2 and Bob receives all of item G_1 and item G_3 plus 7.41% of item G_2) is *envy-free*, *equitable* and *efficient*, or *Pareto optimal*. In fact, Brams and Taylor show that Adjusted Winner *always* produces such an allocation [2]. We will discuss these properties in more detail below.

2 The Adjusted Winner Procedure

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods, or items. A **valuation** of these goods is a vector of natural numbers $\langle a_1, \ldots, a_n \rangle$ whose sum is 100. Let $\alpha, \alpha', \alpha'', \ldots$ denote possible valuations for Ann and $\beta, \beta', \beta'', \ldots$ denote possible valuations for Bob. An **allocation** is a vector of n real numbers where each component is between 0 and 1 (inclusive). An allocation $\sigma = \langle s_1, \ldots, s_n \rangle$ is interpreted as follows. For each $i = 1, \ldots, n$, s_i is the proportion of G_i given to Ann. Thus if there are three goods, then $\langle 1, 0.5, 0 \rangle$ means, "Give all of item 1 and half of item 2 to Ann and all of item 3 and half of item 2 to Bob." Thus AW can be viewed as a function that accepts Ann's valuation α and Bob's valuation β and returns an allocation σ. It is not hard to see that every allocation produced by AW will have a special form: all components except one will be either 1 or 0.

We now give the details of the procedure. Suppose that Ann and Bob are each given 100 points to distribute among n goods as he/she sees fit. In other words, Ann and Bob each select a valuation, $\alpha = \langle a_1, \ldots, a_n \rangle$ and $\beta = \langle b_1, \ldots, b_n \rangle$ respectively. For convenience rename the goods so that

$$a_1/b_1 \geq a_2/b_2 \geq \cdots a_r/b_r \geq 1 > a_{r+1}/b_{r+1} \geq \cdots a_n/b_n$$

[3]When the ratio is closer to 1, a unit gain for Bob costs a smaller loss for Ann.

Let α/β be the above vector of real numbers (after renaming of the goods). Notice that this renaming of the goods ensures that Ann, based on her valuation α, values the goods $G_1, \ldots, G_r$ at least as much as Bob; and Bob, based on his valuation β, values the goods $G_{r+1}, \ldots, G_n$ more than Ann does. Then the *AW* algorithm proceeds as follows:

1. Give all the goods $G_1, \ldots, G_r$ to Ann and $G_{r+1}, \ldots, G_n$ to Bob. Let X, Y be the number of points received by Ann and Bob respectively. Assume for simplicity that $X \geq Y$.

2. If $X = Y$, then stop. Otherwise, transfer a portion of G_r from Ann to Bob which makes $X = Y$. If equitability is not achieved even with all of G_r going to Bob, transfer $G_{r-1}, G_{r-2}, \ldots, G_1$ in that order to Bob until equitability is achieved.

Thus the *AW* procedure is a function from pairs of valuations to allocations. Let $\mathsf{AW}(\alpha, \beta) = \sigma$ mean that σ is the allocation given by the procedure *AW* when Ann announces valuation α and Bob announces valuation $\underline{e}ta$. In [2, 3], it is argued that *AW* is a "fair" procedure, where fairness is judged according to the following properties.

Let $\alpha = \langle a_1, \ldots, a_n \rangle$ and $\beta = \langle b_1, \ldots b_n \rangle$ be valuations for Ann and Bob respectively. An allocation $\sigma = \langle s_1, \ldots, s_n \rangle$ is

- **Proportional** if both Ann and Bob receive at least 50% of their valuation. That is, $\sum_{i=1}^{n} s_i a_i \geq 50$ and $\sum_{i=1}^{n} (1 - s_i) b_i \geq 50$

- **Envy-Free** if no party is willing to give up its allocation in exchange for the other player's allocation. That is, $\sum_{i=1}^{n} s_1 a_i \geq \sum_{i=1}^{n} (1 - s_i) a_i$ and $\sum_{i=1}^{n} (1 - s_i) b_i \geq \sum_{i=1}^{n} s_i b_i$.

- **Equitable** if both players receive the same total number of points. That is $\sum_{i=1}^{n} s_i a_i = \sum_{i=1}^{n} (1 - s_i) b_i$

- **Efficient** if there is no other allocation that is strictly better for one party without being worse for another party. That is for each allocation $\sigma' = \langle s'_1, \ldots, s'_n \rangle$ if $\sum_{i=1}^{n} a_i s'_i > \sum_{i=1}^{n} a_i s_i$, then $\sum_{i=1}^{n} (1 - s'_i) b_i < \sum_{i=1}^{n} (1 - s_i) b_i$. (Similarly for Bob).

In order to simplify notation, let $V_A(\alpha, \sigma)$ be the total number of points Ann receives according to valuation α and allocation σ and $V_B(\beta, \sigma)$ the total number of points Bob receives according to valuation β and allocation σ.

It is not hard to see that for two-party disputes, proportionality and envy-freeness are equivalent. For a proof, notice that

$$\sum_{i=1}^{n} a_i s_i + \sum_{i=1}^{n} a_i (1 - s_i) = \sum_{i=1}^{n} a_i s_i + \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} a_i s_i = 100$$

Then if σ is envy free for Ann, then $\sum_{i=1}^{n} a_i s_i \geq \sum_{i=1}^{n} a_i(1-s_i)$. Hence, $2\sum_{i=1}^{n} a_i s_i \geq \sum_{i=1}^{n} a_i = 100$. And so, $\sum_{i=1}^{n} a_i s_i \geq 50$. The argument is similar for Bob. Conversely, suppose that σ is proportional. Then since $\sum_{i=1}^{n} a_i s_i \geq 50$, $\sum_{i=1}^{n} a_i s_i + \sum_{i=1}^{n} a_i s_i \geq 100 = \sum_{i=1}^{n} a_i$. Then $\sum_{i=1}^{n} a_i s_i + \sum_{i=1}^{n} a_i s_i - \sum_{i=1}^{n} a_i \geq 0$. Hence, $\sum_{i=1}^{n} a_i s_i - \sum_{i=1}^{n} a_i(1-s_i) \geq 0$. And so, $\sum_{i=1}^{n} a_i s_i \geq \sum_{i=1}^{n} a_i(1-s_i)$. The proof is similar for Bob.

Returning to AW, it is easy to see the AW only produces equitable allocations (equitability is essentially built in to the procedure). Brams and Taylor go on to show that AW, in fact, satisfies all of the above properties.

Theorem 1 (Brams and Taylor [2]) *AW produces an allocation of the goods based on the announced valuations that is efficient, equitable and envy-free.*

A formal proof of this Theorem is provided in [2]. For completeness, we include here a proof that AW is proportional (and hence envy-free). Effiency is discussed in the next section.

Lemma 2 *For all α, β, $V_{AW}(\alpha, \beta) \geq 50$.*

Proof Suppose not. That is suppose that $V_{AW}(\alpha, \beta) < 50$. Then the goods can be reordered so that

$$a_1 + \cdots + pa_r = (1-p)b_r + \cdots + b_n < 50$$

Hence $a_1 + \cdots + pa_r + (1-p)b_r + \cdots + b_n < 100$. Now since for each $j = 1, \ldots, r$, $a_j \geq b_j$, we have

$$\begin{array}{rcl} 100 & > & a_1 + \cdots + pa_r + (1-p)b_r + \cdots + b_n + pa_r + (1-p)b_r + \cdots + b_n \\ & \geq & b_1 + \cdots + pb_r + (1-p)b_r + \cdots + b_n \end{array}$$

This is a contradiction since $b_1 + \cdots + pb_r + (1-p)b_r + \cdots + b_n = 100$. □

In fact, we can show something more — AW gives each agent 50 points precisely when the agents input the same valuations.

Lemma 3 *For all α, β, $\alpha = \beta$ iff $V_{AW}(\alpha, \beta) = 50$*

Proof ($\Rightarrow$) Suppose that $\alpha = \beta$. Let $G_1, G_2, \ldots$ be the order of goods induced by the AW procedure. Now the AW procedure will distribute the goods so that

$$a_1 + a_2 + \cdots + pa_r = (1-p)b_r + b_{r+1} + \cdots b_n$$

Since $\alpha = \beta$, for each $j = r, \ldots, n$, $b_j = a_j$. Hence, we have

$$a_1 + a_2 + \cdots + pa_r = (1-p)a_r + a_{r+1} + \cdots a_n$$

Now, since $\sum_{i=1}^{n} a_i = 100$,

$$a_1 + a_2 + \cdots + pa_r = (1-p)a_r + 100 - (a_1 + \cdots + a_r)$$

Thus $2(a_1+a_2+\cdots+pa_r) = 100$ and so $a_1+\cdots+pa_r = 50$. Hence, $V_{AW}(\alpha,\beta) = 50$.

($\Leftarrow$) Suppose that $V_{AW}(\alpha,\beta) = 50$. Suppose that $\alpha \neq \beta$. Then there exist i and j such that $a_i > b_i$ and $a_j < b_j$. The AW procedure produces an allocation where (after renaming the goods)

$$a_1 + \cdots + pa_r = (1-p)b_r + \cdots + b_n = 50$$

Furthermore, the procedure ensures that $i \leq r$. WLOG we can assume $i = 1$ by simply choosing the i that maximizes the ratio a_i/b_i. Using basic algebra, we have

$$a_1 + a_2 + \cdots + a_{r-1} + b_{r+1} + b_{r+2} + \cdots b_n = 100 - pa_r - (1-p)b_r$$

Since $a_1 > b_1$ and for each $k = 2, \ldots, r-1$, $a_k \geq b_k$, we have

$$\begin{aligned} 100 - pa_r - (1-p)b_r &= a_1 + a_2 + \cdots + a_{r-1} + b_{r+1} + b_{r+2} + \cdots b_n \\ &> b_1 + b_2 + \cdots + b_{r-1} + b_{r+1} + \cdots + b_n \end{aligned}$$

Hence,

$$100 - pa_r - pb_r > b_1 + b_2 + \cdots b_n = 100$$

This is a contradiction since $p, a_r, b_r > 0$. □

3 A Geometrical Interpretation of AW

In this section and the one on continuity, it will be useful to think of both valuations and allocations as vectors in n-space, and to use vector notation where such notation will assist our geometric intuition.

Notice that the AW procedure only produces allocations in which all components, except possibly one, are either 1 or 0. In this section, we show that this is not an accident. We will be working in $\mathbb{R}^k$ for $k \geq 1$. An allocation is a vector $\vec{x} \in \mathbb{R}^k$ where each component is a non-negative real less than or equal to 1. Thus the set of all possible allocations is a hypercube in $\mathbb{R}^k$. Let $\mathcal{C}_k = \{\vec{x} \mid \forall i \;\; 0 \leq x_i \leq 1\}$ be this hypercube of dimension k (we will leave out the k when possible).

A **valuation** is a vector $\vec{P} \in \mathbb{R}^k$ where $\sum_{i=1}^{k} P_i = 100$. Let $\cdot$ denote the dot product, that is $\vec{x} \cdot \vec{P} = \sum_{i=1}^{k} x_i P_i$. Now, let $\vec{P}$ and $\vec{Q}$ be two fixed vectors (Ann's valuation and Bob's valuation). As we want to ensure that Ann and Bob both receive the same valuation, we are interested in the hyperplane $\mathcal{H}_{\vec{P},\vec{Q}}$ generated by the following equation

$$\vec{x} \cdot \vec{P} = (\vec{1} - \vec{x}) \cdot \vec{Q}$$

Since $\vec{1} \cdot \vec{Q} = 100$, we have

$$\vec{x} \cdot (\vec{P} + \vec{Q}) = \vec{x} \cdot (\vec{Q} + \vec{P}) = \vec{x} \cdot \vec{Q} + (\vec{1} - \vec{x}) \cdot \vec{Q} = \vec{1} \cdot \vec{Q} = 100$$

Thus $\mathcal{H}_{\vec{P},\vec{Q}} = \{\vec{x} \mid \vec{x} \cdot (\vec{P} + \vec{Q}) = 100\}$. Again we will leave out the subscripts when possible.

For a fixed $\vec{P}$ and $\vec{Q}$, wanting efficency, we can ask for the allocations $\vec{x}$ that maximize $\vec{x} \cdot \vec{P}$ (subject to the above constraints): Let $\mathcal{I} = \mathcal{C}_k \cap \mathcal{H}_{\vec{P},\vec{Q}}$. Define the function $f : \mathcal{I} \to \mathbb{R}$ by $f(\vec{x}) = \vec{x} \cdot \vec{P}$. Then, since $\mathcal{I}$ is a closed and bounded subset of $\mathbb{R}^k$ (hence compact by the Heine-Borel Theorem), f has a maximum value on $\mathcal{I} = \mathcal{C}_k \cap \mathcal{H}_{\vec{P},\vec{Q}}$. Let m be this maximum value, so that for each $\vec{x} \in \mathcal{I}$, $f(\vec{x}) \leq m$ and the set $\mathcal{M} = \{\vec{x} \mid f(\vec{x}) = m\} \neq \emptyset$.

We claim that there is a point of $\mathcal{M}$ which lies on an edge of the hypercube $\mathcal{C}_k$. More formally,

Theorem 4 *There is a point $\vec{x} \in \mathcal{M}$ with all components either 1 or 0 except possibly one. I.e., $\exists j$ such that $\forall i$, if $i \neq j$ then $x_i = 1$ or $x_i = 0$.*

Proof We will show that
$(*)$ if $\vec{x} \in \mathcal{M}$ with $0 < x_i < 1$ and $0 < x_j < 1$ for $i \neq j$, then there is a point $\vec{x}' \in \mathcal{M}$ with $x_l = x'_l$ for all $l \neq i, j$ and either $x'_i = 1$ or $x'_j = 1$.

To see that this statement implies the theorem, take an arbitrary element $\vec{x} \in \mathcal{M}$ (such an element exists since $\mathcal{M}$ is nonempty). Now, each time that $(*)$ is used, the number of strictly fractional components (not 0 or 1) decreases by one. Thus when we are finished there will be at most one fractional component left.

To prove $(*)$ WLOG we may assume that $i = 1$ and $j = 2$. Thus we have

$$x_1 P_1 + x_2 P_2 + \sum_{i=3}^{k} x_i P_i = m$$

where m is the maximum of the function f. Now we must show that either there is $0 \leq x'_1 \leq 1$

$$x'_1 P_1 + P_2 + \sum_{i=3}^{k} x_i P_i = m$$

or there is $0 \leq x'_2 \leq 1$ such that

$$P_1 + x'_2 P_2 + \sum_{i=3}^{k} x_i P_i = m$$

Now if we set $x'_1 = \frac{x_1 P_1 + x_2 P_2 - P_2}{P_1}$, and $x'_2 = 1$ then it is not hard to see that $x'_1 P_1 + P_2 + \sum_{i=3}^{k} x_i P_i = m$. Similarly, if we set $x''_2 = \frac{x_1 P_1 + x_2 P_2 - P_1}{P_2}$ and $x''_1 = 1$. But to show that one of the other of these assignments work, we still need to show that either $0 \leq x'_1 \leq 1$ or $0 \leq x''_2 \leq 1$.

Since x_1 and x_2 are both between 0 and 1, $x_1P_1 + x_2P_2 < P_1 + P_2$. Thus using basic algebra, $x_1' < 1$ and $x_2'' < 1$.

Suppose that $x_1' < 0$ and $x_2'' < 0$. Then since P_1 and P_2 are both positive real numbers, $x_1P_1 + x_2P_2 - P_2 < 0$ and $x_1P_1 + x_2P_2 - P_1 < 0$. Therefore, $x_1P_1 + x_2P_2 < P_2$ and $x_1P_1 + x_2P_x < P_1$ and so $x_1P_1 + x_2P_2 < \frac{1}{2}P_1 + \frac{1}{2}P_2$. Thus

$$\frac{1}{2}P_1 + \frac{1}{2}P_2 + \sum_{i=3}^{k} x_iP_i > x_1P_1 + x_2P_2 + \sum_{i=3}^{k} x_iP_i = m$$

which is a contradiction since we could clearly have used $\frac{1}{2}, \frac{1}{2}$ as our values, and m is the maximum. □

This proof shows that *there exists* an efficient and equitable allocation that only splits one item. Of course, this is not the same as proving that the algorithm *adjusted winner* actually produces such an outcome. This is what Brams and Taylor show in [2].

4 Continuity

Intuitively, as a function from pairs of vectors to real numbers, AW is continuous. That is, "minor changes" in the valuations produces small changes in the points assigned to the agents by AW. In this section we make this statement precise.

For this section assume that there are k goods. We will think of AW as a function that takes two vectors of *real* numbers and returns a real number, i.e., $\mathsf{AW} : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ where $\mathsf{AW}(\alpha, \beta) = V_A(\alpha, AW(\alpha, \beta))$. Of course, stated this way AW is only a partial function on $\mathbb{R}^k \times \mathbb{R}^k$ (only defined on pairs of vectors whose components add up to 100).

Two notions of continuity relevant for our study. The first is the standard notion of continuity and it amounts to AW being continuout in the number of *points* received. Given $v \in \mathbb{R}^k$, the Euclidean norm of v is $||v|| = \sqrt{\sum_{i=1}^{k} v_i^2}$. We say that $F : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ is **continuous** in its frist argument provided for a fixed $v \in \mathbb{R}^k$, for all $\epsilon > 0$ there exists a $\delta > 0$ such that $||x - y|| < \delta$ implies $|F(x, v) - F(y, v)| < \epsilon$. Similarly for "continuity in its second argument". As we will see below, AW is continuous in both of its arguments. The second notion of continuity involves the set of items received by each agent. Thus we think of AW as a funciton from pairs of vectors of real number to *allocations*.

Definition 1 *A function F from $\mathbb{R}^k \times \mathbb{R}^k$ to allocations is said to be* **item continuous in the first argument** *if for a fixed $v \in \mathbb{R}^k$, for all $\epsilon > 0$ there exists $v_1, v_2 \in \mathbb{R}^k \times \mathbb{R}^k$ with $F(v_1, v) = \sigma$, $F(v_2, v) = \sigma'$ and $||v_1 - v_2|| < \epsilon$, then for all $i = 1, \ldots, k$, $\sigma_i = 1$ iff $\sigma_i' = 1$ and $\sigma_i = 0$ iff $\sigma_i' = 0$.*

In other words, small changes in valuations allocates the same set of items to the agents. As we see below, AW is *not* item continuous. We now show that AW is continuous in both arguments. The result follows from the next Lemma.

Suppose that α is Ann's valuation, β is Bob's valuation and σ is the allocation produced by AW (that is $\mathsf{AW}(\alpha, \beta) = \sigma$). Let r be the ratio a_i/b_i where G_i is the item that is divided by the procedure. Define $I = \{l \mid a_l/b_l = r\}$, i.e., I is the set of indices of the goods that have the same ratio as the item which is divided by the procedure.

Lemma 5 *Suppose that α, β, σ and I are defined as above. Suppose that y_1, y_2, y_3 where y_2 is Ann's value of the item being split and y_1, y_3 is Ann's value of all other items in I. Suppose that we choose another item from I to split, call this allocaiton σ'. Say z_1, z_2, z_3 are integers where z_2 is Ann's value of the (new) item being split and z_1, z_3 are Ann's values for all other items in I. Then $V_A(\alpha, \sigma) = V_A(\alpha, \sigma')$, i.e., Ann (and hence Bob) receives the same number of points.*

Proof Let X be the value of allocation out side I that will be allocated to Bob by his valuation. Let Y be the value of allocation out side I that will be allocated to Ann by her valuation. Then

$$V_A(\alpha, \sigma) = X + ry_1 + pry_2 = Y + y_3 + (1-p)y_2$$

where p is the percentage that Bob will get from the item that correspond to y_2. On the other hand

$$V_A(\alpha, \sigma') = X + rz_1 + qrz_2 = Y + z_3 + (1-q)z_2$$

where q is the percentage that Bob will get from the item that correspond to z_2. Also note that $y_1 + y_2 + y_3 = z_1 + z_2 + z_3$. Let $S = y_1 + y_2 + y_3$.

Let $A = ry_1 + pry_2$ and let $B = y_3 + (1-p)y_2$ then $A/r + B = S$ and that gives us $A = r(S - B)$. Substitute in the above equation we get $V_A(\alpha, \sigma) = X + r(S - B) = Y + B$ then $(Y + B)(1 + r) = X + rS + rY$ and that give us $V_A(\alpha, \sigma) = Y + B = (X + rS + rY)/(1 + r)$.

In a similar argument, Let $A' = ry_1 + pry_2$ and let $B' = y_3 + (1-p)y_2$ then $A'/r + B' = S$ and that gives us $A' = r(S - B')$. Substitute in the above equation we get $V_A(\alpha, \sigma') = X + r(S - B') = Y + B'$ then $(Y + B')(1 + r) = X + rS + rY$ and that give us $V_A(\alpha, \sigma) = Y + B' = (X + rS + rY)/(1 + r)$. Thus we $V_A(\alpha, \sigma) = V_A(\alpha, \sigma')$.

□

5 Discontinuity on the Set of Items

For the rest of this section, assume we have k goods. Let α, β be Ann's and Bob's valuations respectively. Define $V_\Sigma(\alpha, \beta, \sigma) = V_A(\alpha, \sigma) + V_B(\beta, \sigma) = \Sigma s_i a_i + \Sigma(1 - s_i)b_i$. For simplicity we will write $V_\Sigma(\sigma)$ instead of $V_\Sigma(\alpha, \beta, \sigma)$ when α, β are clear in the context. Consider the following example.

Assume we have four items and given this valuation v_1 by both player to be:

	Ann	Bob
G_1	25+$\varepsilon/2$	25-$\varepsilon/2$
G_2	25+$\varepsilon/2$	25-$\varepsilon/2$
G_3	25-$\varepsilon/2$	25+$\varepsilon/2$
G_4	25-$\varepsilon/2$	25+$\varepsilon/2$

Clearly, Ann will get the first two items and Bob will get the last two items. Let us consider this valuation v_2 by both player to be:

	Ann	Bob
G_1	25-$\varepsilon/2$	25+$\varepsilon/2$
G_2	25-$\varepsilon/2$	25+$\varepsilon/2$
G_3	25+$\varepsilon/2$	25-$\varepsilon/2$
G_4	25+$\varepsilon/2$	25-$\varepsilon/2$

According to AW, Ann will get the last two items and Bob will get the first two instead. Note that $||v_1 - v_2|| = \varepsilon$. In fact, we have the following straightforward proposition.

Proposition 6 *Assume we have k goods. For any $\varepsilon > 0$ there are valuations v_1 and v_2 such that:*

- $||v_1 - v_2|| = \varepsilon$
- $\forall i$ *we have* $\sigma_1(i) = 1$ *iff* $\sigma_2(i) = 0$
- $\forall i$ *we have* $\sigma_1(i) = 0$ *iff* $\sigma_2(i) = 1$

Proof We have two cases. First, assume that k is even. Then define v_1 as following: Ann's and Bob's valuation are $a_i = 100/k + \varepsilon/2, b_i = 100/k - \varepsilon/2$ for $i \leq k/2$, i.e. for the first half of the goods, and $a_i = 100/k - \varepsilon/2, b_i = 100/k + \varepsilon/2$ for $i > k/2$. Define v_2 as following: Ann's and Bob's valuation are $a_i = 100/k - \varepsilon/2, b_i = 100/k + \varepsilon/2$ for $i \leq k/2$, i.e. for the first half of the goods, and $a_i = 100/k + \varepsilon/2, b_i = 100/k - \varepsilon/2$ for $i > k/2$. Then these v_1, v_2 satisfies all the three properties. The case when k is odd is similar. □

6 The Distance Between Announced Allocations

In this section we formalize the intuition that the more the valuations differ, the more points each agent will receive. Since AW only produces equitable allocations, we can think of the function AW as a function from pairs of valuations to real numbers. Let $V_{AW}(\alpha, \beta)$ denote the total points that AW allocates to each agent – say Ann, (according to the announced valuations α and β). Formally, $V_{AW}(\alpha, \beta)$ is defined to be $V_A(\alpha, \mathsf{AW}(\alpha, \beta))$. Of course, we coild define it in terms of Bob's valuation, but they are equal so it does not matter which definition is used.

Given an allocation α for Ann, if Ann increases any component then she must decrease another component as the sum of the components must be 100. Now if Ann wants to accentuate the difference between her allocation and Bob's allocation, then she will only increase points on goods that she values more than Bob. Let α, α' and β, β' be two valuations for Ann and Bob, respectively. We say that $(\alpha, \beta) \prec^A_{ij} (\alpha', \beta')$ if

1. $\beta = \beta'$
2. $\alpha_i > \beta_i$, $\alpha_j < \beta_j$, $\alpha'_i = \alpha_i + 1$ and $\alpha'_j = \alpha_j - 1$.
3. for all $k \neq i, j$, $\alpha'_k = \alpha_k$

Similarly, we define $\prec^B_{ij}$ with respect to Bob's valuation. The intuition is that if $(\alpha, \beta) \prec^A_{ij} (\alpha', \beta')$, then the pair (α', β') represents a situation in which Ann has "increased" by 1 unit the difference between α and β. We say $(\alpha, \beta) \prec (\alpha', \beta')$ if there is a sequence of pairs of valuations linearly ordered by the $\prec^A_{ij}, \prec^B_{ij}$ relations (with varying i, j) that begins with (α, β) and ends with (α', β'). Thus $\prec$ is the transitive closure of the **union** of the relations $\prec^A_{ij}$ and $\prec^B_{ij}$. It is not hard to see that $\prec$ is a (non-reflexive) partial order. The main theorem of this section is

Theorem 7 *If $(\alpha, \beta) \prec (\alpha', \beta')$, then $V_{AW}(\alpha, \beta) < V_{AW}(\alpha', \beta')$.*

We return to the proof of the main theorem of this section (Theorem 7). The proof of the theorem is an easy consequence of the following fact.

Lemma 8 *Suppose that $(\alpha, \beta) \prec^A_{ij} (\alpha', \beta')$, then $V_A(\alpha, \mathsf{AW}(\alpha, \beta)) < V_A(\alpha', \mathsf{AW}(\alpha', \beta'))$.*

Proof To see this, note that when Ann increased some valuations by 1, where it already exceeded Bob's valuation for that item, then she gets that item in the initial allocation both before this change and after the change. Hence Ann receives more points in her first allocation, and Bob must be compensated for this fact in the final allocation. Thus Bob's final score will increase. But since both Ann and Bob receive the same final score, they will both benefit. We postpone the details and the arithmetic to the final version of the paper. □

7 NonLinear Utility Functions

There are two assumptions about the agent's utility functions that are needed for the previous discussions. First of all, the agents utilities are assumed to be **additive**. That is the utility of a set of goods is the sum of the utilities assigned to each individual good. Second, the utility function for each individual good is assumed to be linear. In this section we consider situations in which this second assumption is dropped.

The intuition for dropping the linearity assumption is that there are many situations in which agents mare share a good but each may get *more* utility than can be described by a linear utility function. For example, suppose that Ann and Bob both assign 100 points to a car. The *AW* procedure would force Ann and Bob to split the car in half. Thus both receive 50 points. Suppose that both Ann and Bob only want to use the car on weekends. Some weekends Bob uses the car and some weekends Ann uses the car. If It is not always the case that they both need to use the car at the same time, then it is possible that each agent can actually receive *more* than 50 points. Suppose that on only half of the weekends is there a conflict between Ann and Bob over the use of the car. Thus both Ann and Bob get to use the car 75% of the time they need to. This can be interpreted as both Ann and Bob receiving 75 points.

Another good example is roommates. When two roommates share an apartment, they are not getting half of the value of that apartment. They are still both are getting to use the Kitchen, the bathroom and the living as if they are living by themselves. It is not the case the roommates always need to use the same resources at the same time.

The following example illustrates the type of situations we have in mind. Suppose that Ann and Bob have the following valuation:

Item	Ann	Bob
G_1	30	20
G_2	30	20
G_3	20	30
G_4	20	30
Total	100	100

In this case, *AW* will give the first two items to Ann and the last two items to Bob and they both receive 60 points. Now assume that both agents' partial utility function of each item is given by the equation $2x-x^2$ (x is the percentage of the good that the agent receives). Thus for good G_1, the total number of points that Ann receives from $(x \times 100)\%$ of G_1 is $60x - 30x^2$ for Ann and $40x - 20x^2$ for Bob. If Ann gets 60% of the first two items and 40% of the last two items, and Bob gets 40% of the first two items and 60% of the last two items, then they both end up with 76 points.

We propose a generalization of the adjusted winner procedure that takes into account the fact that agents' may have nonlinear utilities.

Formulation: Each player will supply two numbers for each items: his/her valuation of getting 100% of the item and his/her valuation of getting 50%. They we can compute the function that will represent each player. For example see the valuations below:

Item	Ann 100%	Ann 50%	Bob 100%	Ann 50%
G_1	30	20	20	15
G_2	30	20	50	30
G_3	40	30	30	20
Total:	100	70	100	65

Using these values, we can approximate the agents' (quadratic) valuation function. Then, using standard techniques, find the maximal total utility subject to the constraint that the agents' total valuations are the same. The details are left for the full version of the paper.

More formally, let $\Gamma = \{G_1, \ldots, G_k\}$ be a set of goods. It is assumed that goods are divisible, as such it is possible for an agent to receive a portion of a good. If $p \in [0,1]$, let (p, G) represent the situation where the agent receives $(p \times 100)\%$ of G. Since agents may receive portions of goods, we define utility functions as

$$u : [0,1] \times \Gamma \to [0,100]$$

where $u(p, G) = r$ means that the agent assigns utility r to receiving $(p \times 100)\%$ of G. Assuming linearity implies that $u(p, G) = pu(1, G)$. Of course any closed interval would work here since we can always normalize. Since Γ is finite, we can think of a utility function u as a tuple $\langle u_{G_1}, u_{G_2}, \ldots, u_{G_k} \rangle$ where $u_{G_i} : [0,1] \to [0, u(1, G_i)]$.

Assuming additivity, given a utility function u, we define the function $\overline{u}$ on the set of subsets of Γ as follows. Let $\Delta \subseteq \Gamma$, then

$$\overline{u}(\Delta) = \sum_{G \in \Delta} u(G)$$

In order to simplify notation, we will write $u(\Delta)$ instead of $\overline{u}(\Delta)$.

Let u be the utility function of Ann and v the utility function of Bob. The *AW* procedure asks Ann and Bob to represent their utility functions as vectors whose sum of the components is 100. Given Ann's valuation $\alpha = \langle a_1, \ldots, a_k \rangle$, *AW* approximates Ann's utility function as follows: each u_{G_i} is the straight line going through (0,0) and $(1, a_i)$. Given Bob's valuation $\beta = \langle b_1, \ldots, b_k \rangle$, *AW* approximates his utility function as follows: each v_{G_i} is the straight line from $(0, b_i)$ to $(1, 0)$. Viewed in this light, *AW* is a function that accepts two linear utility functions and returns an allocation which is equitable, envy-free and efficient with respect to its two arguments.

More generally, let F be any function from pairs of utility functions to the set of allocations. We will show that under suitable conditions, there is a function F such that $F(u, v)$ is envy-free, equitable and efficient. Furthermore, $F(u, v)$ will produce an allocation which is more efficient than the allocation produced by *AW*.

Definition 1 *Suppose that u is a utility functions, G a good and $x, y \in [0,1]$.*

- *u is* **strictly monotonic with respect to** *G_i if $x < y$ implies $u_G(x) < u_G(y)$*

- *u is* **strictly anti-monotonic with respect to** *G if $x < y$ implies $u_G(x) > u_G(y)$*
- *u is* **strictly concave with respect to** *G if for all $\lambda \in [0,1]$, $u_G\lambda x + (1-\lambda)y) > \lambda u_G(x) + (1-\lambda)u_G(y)$*
- *u is* **strictly convex with respect to** *G if for all $\lambda \in [0,1]$, $u_G(\lambda x + (1-\lambda)t) < \lambda u(x) + (1-\lambda)u_G(x)$*

We say that u is strictly monotonic if u is strictly monotonic with respect to G for each good G. Similarly for the other properties. The following fact is straightforward.

Fact: If u is strictly monotonic with respect to G, v is anti-monotonic with respect to G and u_G and v_G intersect, then they interesect at a unique point.

Definition 2 *Let u and v be two utility functions. We say that u and v are* **complementary with respect to** *G if*

1. *u_G is monotonic;*
2. *v_G is anti-monotonic; and*
3. *$u_G(0) = v_G(1)$ and $u_G(1) = v_G(0)$.*

We say u and v are complementary utility functions of u and v are complementary with respect to G for each good G. Finally, we say that a utility function is continuous if u_G is continuous for each good G. The following lemma shows that for one good, if we assume the agents' utility functions are complementary, continuous and concave then we can find an allocation which is better for both agents than the allocation produced by AW.

Lemma 9 *Suppose that u and v are continuous and complementary utility functions with respect to G. Then if u_G and v_G are concave, there exists a unique point x_0 such that $u_G(x_0) = v_G(x_0)$ and $u_G(x_0) \geq (u_G(0) + u_G(1))/2$ ($v_G(x_0) \geq (u_G(0) + u_G(1))/2$).*

Proof By assumption u_G is strictly monotonic, continuous and concave; v_G is continuous, strictly anti-monotonic and concave; and $u_G(0) = v_G(1)$ and $u_G(1) = v_G(0)$. It is easy to see that there must be a unique point x_0 such that $u_G(x_0) = v_G(x_0)$. We must show $u_G(x_0) \geq (u_G(0) + u_G(1))/2$. Suppose $u_G(x_0) < (u_G(0) + u_G(1))/2$. Then since u_G is concave,

$$(*) \quad v_G(x_0) = u_G(x_0) < (u_G(0) + u_G(1))/2 \leq u_G(1/2)$$

Furthermore since, $u_G(0) = v_G(1)$ and $u_G(1) = v_G(0)$ and v_G is concave.

$$(**) \quad v_G(x_0) = u_G(x_0) < (u_G(0) + u_G(1))/2 = (v_G(1) + v_G(0))/2 \leq v_G(1/2)$$

There are three cases to consider:

1. $x_0 < 1/2$. Then since v_G is anti-monotonic, $v_G(x_0) > v_G(1/2)$. But this contradicts $(**)$

2. $x_0 > 1/2$. Then since u_G is monotonic, $u_G(x_0) > u_G(1/2)$. But this contradicts $(*)$.

3. $x_0 = 1/2$. This contradicts both $(*)$ and $(**)$.

□

With one good, the *AW* procedure splits the good in half giving each agent 50 points. Thus the above theorem shows that under suitable assumptions about the utility function, there exists an envy-free, equitable and efficient allocation which is better for both parties than the one produced by *AW*. Can a similar argument be constructed for any number of goods?

References

[1] J. B. Barbanel (Author) and A. Taylor (Introduction) *The Geometry of Efficient Fair Division* Cambridge University Press, 2005.

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

[3] Steven Brams and Alan Taylor, *The Win-Win Solution: Guaranteeing Fair Shares to Everybody*, W. W. Norton & Company, New York, 1999.

[4] Rohit Parikh and Eric Pacuit, "Safe votes, sincere votes, and strategizing", presented at the *Workshop on Uncertainty in Economics*, Singapore 2005.

Eric Pacuit
ILLC, University of Amsterdam
1018 TV Amsterdam, The Netherlands
Email: epacuit@science.uva.nl

Rohit Parikh
Brooklyn College and CUNY Graduate Center
365 5th Avenue
New York City, NY 10016
Email: rparikh@gc.cuny.edu

Samer Salame
CUNY Graduate Center
365 5th Avenue
New York City, NY 10016
Email: ssalame@gmail.com

Merging judgments and the problem of truth-tracking

Gabriella Pigozzi and Stephan Hartmann

Abstract

The problem of the aggregation of consistent individual judgments on logically interconnected propositions into a collective judgment on the same propositions has recently drawn much attention. The difficulty lies in the fact that a seemingly reasonable aggregation procedure, such as propositionwise majority voting, cannot ensure an equally consistent collective outcome. The literature on judgment aggregation refers to such dilemmas as the *discursive paradox*. So far, three procedures have been proposed to overcome the paradox: the premise-based and conclusion-based procedures on the one hand, and the merging approach on the other hand. In this paper we assume that the decision which the group is trying to reach is factually right or wrong. Hence, the question is how good the merging approach is in tracking the truth, and how it compares with the premise-based and conclusion-based procedures.

1 Introduction

The problem of judgment aggregation was first identified by the Law professors Lewis Kornhauser and Larry Sager [10, 11]. In their example, a court has to make a decision on whether a person is liable of breaching a contract (proposition R, or conclusion). The judges have to reach a verdict following the legal doctrine. This states that a person is liable if and only if she did a certain action X (first premise P) and had contractual obligation not to do X (second premise Q). The legal doctrine can be formally expressed as the rule $(P \wedge Q) \leftrightarrow R$. Each member of the court expresses her judgment (in the form of yes/no) on the propositions P, Q and R such that the rule $(P \wedge Q) \leftrightarrow R$ is satisfied.

Suppose now that the seven members of the court make their judgments according to the following table:

	P	Q	R
Member 1	Yes	Yes	Yes
Member 2	Yes	Yes	Yes
Member 3	Yes	Yes	Yes
Member 4	Yes	No	No
Member 5	Yes	No	No
Member 6	No	Yes	No
Member 7	No	Yes	No
Majority	Yes	Yes	No

Each judge expresses a consistent opinion, i.e. she says yes to R if and only if she says yes to both P and Q. However, propositionwise majority voting (consisting in the separate aggregation of the votes for each proposition P, Q and R via majority rule) results in a majority for P and Q and yet a majority for $\neg R$. This is clearly an inconsistent collective result. The paradox lies in the fact that majority voting can lead a group of rational agents to endorse an irrational collective judgment. The literature on judgment aggregation refers to such dilemma as the *discursive paradox* (or *doctrinal paradox*).

The first two escape-routes that have been suggested are the *premise-based procedure* and the *conclusion-based procedure* [14, 4, 13]. The first procedure is to let each member vote on each premise and to declare the defendant liable only if a majority of the court believes that she did the action X and that she was under contract obligation not to do X. The second procedure requires the judges to privately decide about P and Q and to publicly express their opinions on R only. The defendant will be declared liable if and only if a majority of the judges actually believes that she is liable. Clearly, in the conclusion-based procedure nothing can be said about the reasons supporting the final decision.

In [15] it has been argued that the two above suggested escape-routes from the paradox are not satisfactory methods for group decision-making. The premise-based procedure is problematic because it does not univocally identify what a premise is. To see why, suppose that a group of individuals make their judgments on the propositions A, B and C according to the decision rule $((A \wedge B) \vee (\neg A \wedge \neg B)) \leftrightarrow C$. It is easy to construct examples where premise-based procedure gives two divergent results depending on what we take to be the premises (the atomic propositions A, B and C or the disjuncts $A \wedge B$ and $\neg A \wedge \neg B$). This problem was first noticed by Bovens and Rabinowicz [3] who referred to it as the *instability* of the premise-based procedure. On the other hand, the conclusion-based procedure avoids the paradox at the price of incomplete collective judgments. In all those situations in which a group has to reach a conclusion, but also needs to provide reasons for that decision (as in the original formulation of the doctrinal paradox), the conclusion-based cannot serve as proper aggregation method.

Therefore, a new aggregation procedure, providing a collective decision as well as the reasons for that decision, was introduced in [15]. This approach (that we will call merging — or fusion — procedure) was inspired by a family of operators defined in artificial intelligence [7, 6] in order to merge finite sets of propositions. Not only complex collective decisions are paradox-free when the inconsistent collective judgments are ruled out from the set of possible solutions. Also, an outcome in the merging approach is a complete collective judgment on the premises and on the conclusion

However, situations like the Kornhauser and Sager' court example do not only require that consistent individual opinions are aggregated into a rational group judgment, but also that the group makes the *right* decision. The defendant factually is (or is not) guilty: There exists an objective truth that the court is trying to reach. Therefore, a natural question is: In addition to guarantee

consistent group outcomes, does the merging procedure also select the correct decision? The present paper addresses this question.

An epistemic perspective on judgment aggregation and, in particular, on the premise-based and conclusion-based procedure, was discussed by Bovens and Rabinowicz in [3]. Following their work, and making various independence assumptions as in the Condorcet Jury Theorem, we will introduce our framework in order to test how good the fusion procedure is in tracking the truth. Finally, we will illustrate the results obtained by computer simulation and compare them with the results for premise-based and conclusion-based procedure.

Let us first start by briefly recalling the merging approach.

2 The merging procedure

The fusion procedure is inspired by an aggregation operator defined in artificial intelligence in order to combine several finite sets of propositions (*bases*) [7, 6]. In fact, one of the major problems that artificial intelligence needs to address is the combination of different and potentially conflicting sources of information. Examples are multi-sensor fusion, database integration and expert systems development.[1]

Clearly, belief fusion and judgment aggregation share a similar problem, viz. the definition of operators that produce collective opinion from individual bases. The discursive dilemma rests upon the fact that, when the individual judgments on atomic propositions conform to some logical constraints on those propositions, this does not ensure to obtain a consistent (i.e. obeying the same logical constraints) collective judgment set. On the other hand, one of the key points in the literature of belief fusion is precisely that the aggregation of consistent knowledge bases does not guarantee a consistent collective outcome. To overcome this problem, domain-specific restrictions (*integrity constraints*) are imposed on the final base. This ensures that unwanted solutions are ruled out from the set of possible group outcomes.

Let $N = \{1, 2, ..., n\}$ $(n \geq 2)$ be a set of individuals making their judgments on a given finite set X of propositions (*agenda*). Let $\mathcal{L}$ be a finitary propositional language, built up from a finite set $\mathcal{P}$ of propositional letters and the usual logical connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$). The belief base K_i of an agent i is a consistent and complete finite set of atomic propositions and compound propositions (this corresponds to the individual judgment set).

An *interpretation* is a function $\mathcal{P} \rightarrow \{0, 1\}$ and it is represented as the list of the binary evaluations. For example, given three propositional variables P, Q and R, the vector (0,1,0) stands for the interpretation in which P and R are false and Q is true. Let $\mathcal{W} = \{0, 1\}^{\mathcal{P}}$ be the set of all interpretations. An interpretation is a *model* of a propositional formula if and only if it makes the formula true in the usual truth functional way.

[1] See [5] for a survey on logic-based approaches to information fusion.

IC is the belief base whose elements are the integrity constraints. These are extra conditions imposed on the result of the merging operator. Given a multi-set $E = \{K_1, K_2, \ldots, K_n\}$ and IC, a merging operator $\mathcal{F}$ is a function that assigns a belief base to E and IC. By borrowing the term from judgment aggregation, we call E a *profile*. Let $\mathcal{F}_{IC}(E)$ denote the collective belief base resulting from the IC merging on E. In a model-based merging operator the only possible collective outcomes are the models of IC. A majority fusion operator will the select the (eventually more than one) model that minimizes the *distance* to the profiles.

The most widely used distance in the literature is the Hamming distance. This is defined as the number of propositional letters on which two interpretations differ. For example, the Hamming distance between $\omega = (1, 0, 0, 1)$ and $\omega' = (0, 1, 0, 1)$ is $d(\omega, \omega') = 2$.

The first step is to determine the Hamming distance between those interpretations that are models of IC and the models of each base K_i in the profile E. The next step is to assign a distance value to each model of IC and a profile E. This is defined by the sum of the Hamming distances defined before.

To illustrate how the majority belief fusion operator works, we apply it to our initial court example. In the new terminology, the agenda is $X = \{P, Q, R\}$ with $IC = \{(P \wedge Q) \leftrightarrow R\}$. The models for each belief base are the following:

$$\begin{aligned}
&\mathrm{Mod}(K_1) = \mathrm{Mod}(K_2) = \mathrm{Mod}(K_3) = \{(1,1,1)\} \\
&\mathrm{Mod}(K_4) = \mathrm{Mod}(K_5) = \{(1,0,0)\} \\
&\mathrm{Mod}(K_6) = \mathrm{Mod}(K_7) = \{(0,1,0)\}
\end{aligned}$$

The table below shows the result of the IC majority fusion operator on $E = \{K_1, \ldots, K_7\}$. The row with a shaded background correspond to the selected collective outcome.

	K_1	K_2	K_3	K_4	K_5	K_6	K_7	$\mathcal{F}_{IC}(E)$
(1,1,1)	0	0	0	2	2	2	2	8
(1,0,0)	2	2	2	0	0	2	2	10
(0,1,0)	2	2	2	2	2	0	0	10
(0,0,0)	3	3	3	1	1	1	1	13

Because $\mathcal{F}_{IC}(E)$ is an IC merging operator, the possible collective outcomes are chosen among the interpretations that are models of IC. Thus, no paradox arises by using this fusion operator. We should mention that the fusion operator does not necessarily select a unique group decision. In some cases, the operator selects a set of models, i.e. the result is a tie between some belief bases.

The question we want to address now is whether the fusion approach not only prevents the discursive dilemma, but also is a good truth-tracker. Hence, whether a group that applies the merging procedure can not only keep away from irrational decisions, but has also a good chance to make the right decision. Using the Condorcet Jury Theorem, Bovens and Rabinowicz have explored how good truth-trackers the premise-based and the conclusion-based procedures are.

Our framework is introduced in the next section following [3] and making various independence assumptions as in the Condorcet Jury Theorem. We will then present some results about the fusion procedure and, finally, we will compare the performance of the fusion operator with the performance of the premise-based and the conclusion-based approaches described in [3].

3 The framework

The Condorcet Jury Theorem provides a justification for the majority rule in epistemic terms. It states that if the chance that an individual correctly judges the truth or falsity of a proposition is greater than fifty percent (her *competence*), then the chance that the majority of the group will come to the right decision will increase with the size of the group. In other words, individual probabilities turn into a group probability that is greater. More precisely, the Condorcet Jury Theorem can be formulated as follows:

> Suppose there is a group of N individuals (with N odd and greater than 1). Assume also that each group member has a chance $0.5 < p < 1$ of correctly assessing the truth or falsity of a proposition, and this chance does not depend on the other group member's judgments. Then, the probability that the group's majority judgment on that proposition is correct is greater than p and converges to 1 as the number of voters increases to infinity.

The Condorcet Jury Theorem requires that the number of voters is odd, that the voters are equally competent and independent. In order to avoid computational complexity, we need to make additional assumptions. These are as in [3]:

(a) The prior probability that P and Q are true are equal (q).

(b) All voters have the same competence to assess the truth of P and Q.

(c) P and Q are (logically and probabilistically) independent.

As [3], we will model the merging procedure for $P \wedge Q \leftrightarrow R$. Both the literature on judgment aggregation and the fusion approach assume that each individual judgment set is logically consistent. Hence, for $P \wedge Q \leftrightarrow R$ only four situations are possible (their corresponding models are also annotated):

$S_1 = \{P, Q, R\} = (1, 1, 1)$
$S_2 = \{P, \neg Q, \neg R\} = (1, 0, 0)$
$S_3 = \{\neg P, Q, \neg R\} = (0, 1, 0)$
$S_4 = \{\neg P, \neg Q, \neg R\} = (0, 0, 0)$

From (a), we derive that the prior probabilities of the four possible situations are (with $\bar{x} := 1 - x$):

$$\mathcal{P}(S_1) = q^2; \;\; \mathcal{P}(S_2) = \mathcal{P}(S_3) = q\bar{q}; \;\; \mathcal{P}(S_4) = \bar{q}^2$$

We now want to calculate the probability that fusion ranks the right judgment set first (let us denote this proposition with $\mathcal{P}(F)$). Note that $\mathcal{P}(F) = \sum_{i=1}^{4} \mathcal{P}(F|S_i) \cdot \mathcal{P}(S_i)$. Thus, we have to calculate the conditional probabilities $\mathcal{P}(F|S_i)$ for $i = 1, \ldots, 4$. To see how it works, suppose that S_1 is the right judgment set. Then n_i (of N) voters will vote for profile S_i, with $n_1 + n_2 + n_3 + n_4 = N$.

We have seen that the majority merging operator selects the (eventually more than one) model that minimizes the distance to the profiles. This means that — if S_1 is the right judgment set — fusion is a good truth-tracker if $d_1 \leq \min(d_1, \ldots, d_4)$.

The distances d_i can be expressed in terms of the numbers n_i of voters for the situations S_i $(i = 1, \ldots, 4)$:

$$\begin{aligned} d_1 &= 2n_2 + 2n_3 + 3n_4 \quad ; \quad d_2 = 2n_1 + 2n_3 + n_4 \\ d_3 &= 2n_1 + 2n_2 + n_4 \quad ; \quad d_4 = 3n_1 + n_2 + n_3 \end{aligned}$$

For example, d_1 is obtained by summing the distances between S_1 and S_2, S_3 and S_4 times the number of voters for each S_i. The Hamming distance between S_1 and S_2 is 2. Hence, this value is multiplied by the number of voters for S_2 (which is n_2). The values $2n_3$ and $3n_4$ are obtained with the same procedure with respect to S_3 and S_4.

Finally, we can calculate the probability that fusion selects S_1 provided that S_1 is the right judgment set :

$$\mathcal{P}(F|S_1) = \sum_{n_1,\ldots,n_4=0}^{N} \binom{N}{n_1, \ldots, n_4} p^{2n_1} (p\bar{p})^{n_2+n_3} \bar{p}^{2n_4} \, \mathcal{C}(n_1, \ldots, n_4)$$

The sum is constrained: $\mathcal{C}(n_1, \ldots, n_4) = 1$ if (i) $\sum_{i=1}^{4} n_i = N$ and (ii) $d_1 \leq \min(d_1, \ldots, d_4)$. Otherwise $\mathcal{C}(n_1, \ldots, n_4) = 0$.

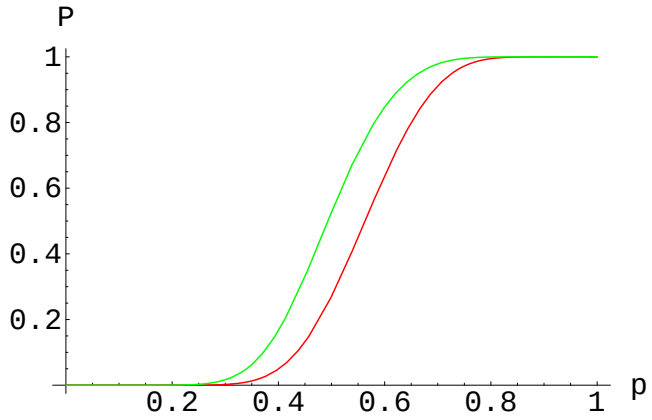
We can now present some results about how good in selecting the right judgment set the merging operator is. We will then turn to some figures showing the behavior of the fusion approach compared to the premise-based and the conclusion-based procedures.

4 Results

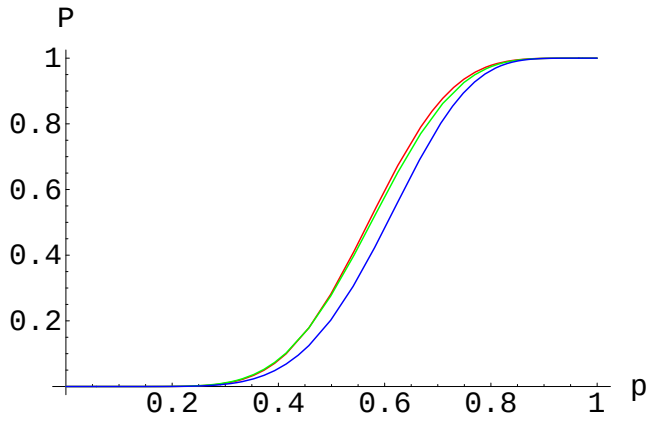
4.1 Testing the merging procedure

In Section 2, we have seen that the notion of distance used in the fusion approach defines a pre-order on the possible outcomes. Thus, our first question is how good is belief fusion in selecting the correct judgment set as the first element in the ranking. The figure below shows how fusion ranks the right profile first

(the red curve — *abbr.* R) or second (the green curve — *abbr.* G) for $N = 19$ and $q = .5$

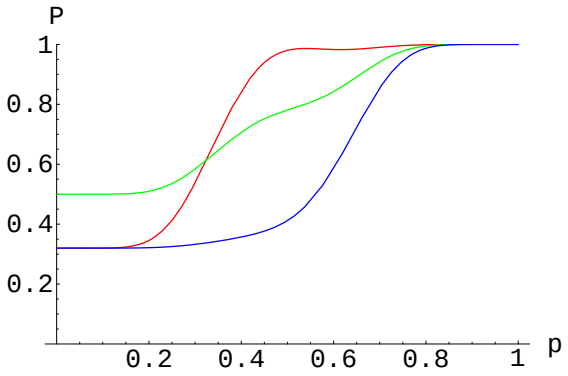


We now want to compare how the probability that fusion ranks the right profile first depends on different values of q. The plot is for $N = 11$ and three values of q: $q = .2$ (R), $q = .5$ (G), $q = .8$ (the blue line — *abbr.* B)



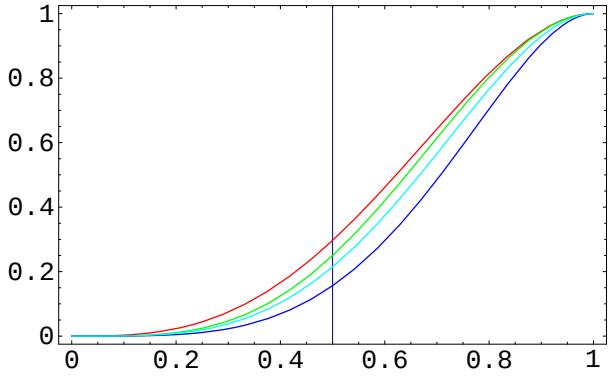
It turns out that the probability P is quite independent on the priors q.

However, different values of the priors q matter when we look at how good fusion is in ranking first a judgment set with the right decision (but not necessarily the correct reasons for that decision). The figure below shows the results for $N = 17$ and $q = .2$ (R), $q = .5$ (G), $q = .8$ (B)



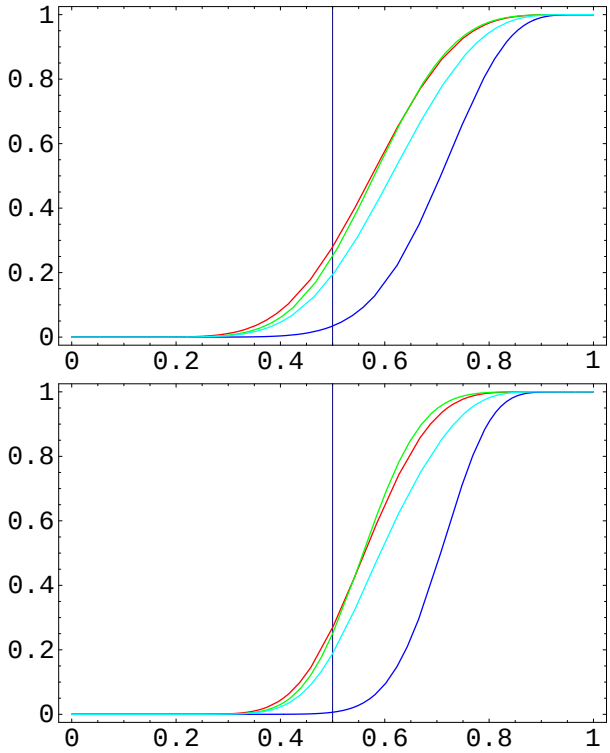
4.2 The merging approach compared to the premise-based and the conclusion-based procedures

The second set of our results compare the merging approach with the premise-based and the conclusion-based procedures. We start with a small number of voters ($N = 3$) and $q = .5$. The first figure below shows how fusion ranks the right profile first (R) compared with premise-based procedure (G), conclusion-based procedure (B) and the conclusion-based procedure with the right reasons (turquoise curve — *abbr.* T).



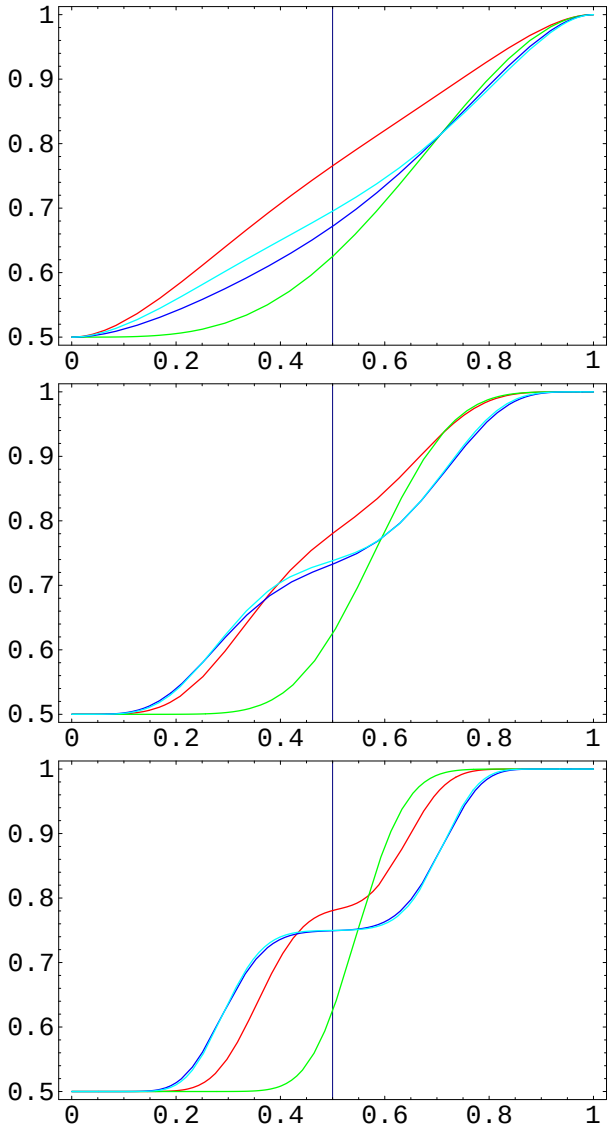
The merging operator outperform all the other procedures. However, it is no surprise that the second best procedure is the premise-based one. In fact, from the Bovens and Rabinowicz's findings, we know that if we aim at reaching the right decision for the right reasons, we should prefer the premise-based procedure to the conclusion-based.

The next two figures illustrate the behavior of the fusion operator compared to the contender procedures when the number of voters increases ($N = 11$ and $N = 21$ respectively):



Clearly, the fusion approach (R) does significantly better than the conclusion-based (for the right reasons or not — the turquoise and blue lines). However, for high values of competence p, the premise-based procedure (G) is slightly better as a truth-tracker.

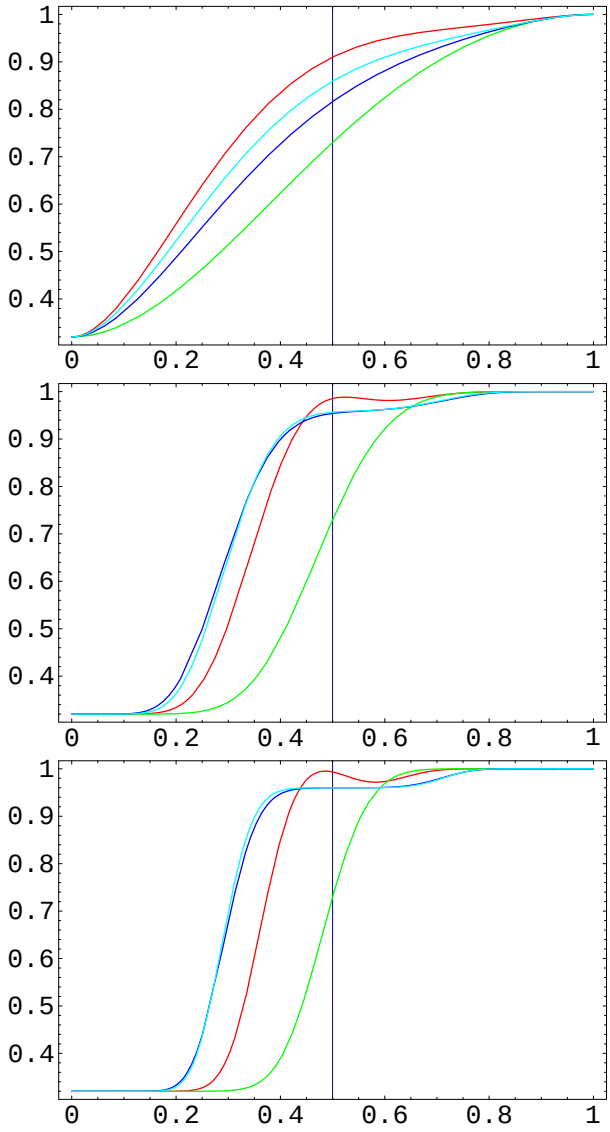
We now turn to evaluate how the fusion approach (R) ranks a judgment set with the right result (but not necessarily the right reasons) first, and we contrast this with the premise-based (G), the conclusion-based (B) and the conclusion-based for the right reasons (T) procedures. As before, we test the procedures for $q = .5$ and for increasing number of voters ($N = 3$, $N = 11$ and $N = 31$ respectively):



It turns out that fusion greatly outperforms all the other aggregation procedures under investigation for small size groups. Yet, as the size of the group increases, both the conclusion based procedures (B and T lines) do better than the fusion operator for low values in competence, and the premise-based procedure (G) does better than fusion for high values of p. But, for the middle values of p, merging is always superior. We can also observe that, whenever the fusion is not the best procedure, it lies in-between the premise-based and the conclusion-based procedures.

The next three pictures illustrate the same comparison, for a different value

of prior ($q = .2$). As before, the number of voters increases, from $N = 3$ (first plot) to $N = 21$ (second) and $N = 51$ (third plot).

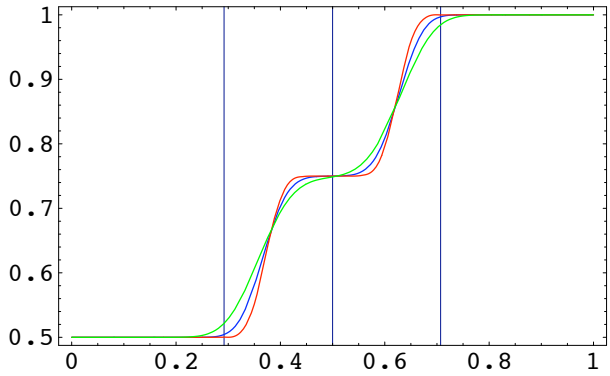


Again, for small-sized groups, fusion is the best procedure to reach the right decision. When the number of the voters increase, the conclusion-based procedures (B and T curves) do better than fusion, but only for low competence values. Different values of priors do not undermine the superiority of the fusion approach in the middle values of p ($.4 \leq p \leq .6$). More interesting, for p around .5, the probability that fusion selects the right decision is almost 1! Finally, for higher values of p, premise-based procedure is only slightly better than the

merging operator.

The last figure shows that the trend of fusion for $N =?$ voters (G) is very close to the curve obtained for increasingly higher number of voters: $N =?$ (B) and $N =?$ (R).

Say something about the two values (vertical lines) of BR06.



Summarizing, our computer simulations show that the fusion approach does especially well for middling values of p. Nevertheless, for other values of p, the fusion operator is often in between the premise-based and the conclusion-based procedures (whichever is better in the case at hand).

Hypothesis: Fusion works best for realistic cases ($p \approx .5$) and takes the best of both worlds, i.e. PBP and CBP.

5 Conclusion and future plans

- Belief merging as a valuable tool to aggregate individual judgment sets:
  - no paradox
  - ranking on all possible social outcomes
  - no instability problem
  - propositions can be give different interpretation $\Rightarrow$ different fusion operators?
- We examined how good a truth-tracker the fusion approach is.
- In future work, we will:
  - work with a larger number of voters,
  - a larger number of premises,
  - examine the disjunctive case, and
  - use other distance measures.
- We will also explore the political and philosophical significance of the fusion approach.

References

[1] K. J. Arrow (1963) *Social Choice and Individual Values*, Wiley, New York, second edition.

[2] K. J. Arrow, A. K. Sen, and K. Suzumura (eds.). *Handbook of Social Choice and Welfare*, Vol.1, Elsevier, 2002.

[3] L. Bovens and W. Rabinowicz. Democratic answers to complex questions. An epistemic perspective, *Synthese*, 150: 131–153, 2006.

[4] B. Chapman. Rational Aggregation. *Politics, Philosophy and Economics*, 1(3): 337-354, 2002.

[5] S. Konieczny and E. Grégoire. Logic-based approaches to information fusion. *Information Fusion*, 7: 4–18, 2006.

[6] S. Konieczny. *Sur la Logique du Changement: Révision et Fusion de Bases de Connaissance*, Ph.D. dissertation, University of Lille, France, 1999.

[7] S. Konieczny and R. Pino-Pérez. On the logic of merging. In *Proceedings of KR'98*, Morgan Kaufmann, pages 488–498, 1998.

[8] S. Konieczny and R. Pino-Pérez. Propositional belief base merging or how to merge beliefs/goals coming from several sources and some links with social choice theory. *European Journal of Operational Research*, 160(3): 785–802. 2005.

[9] L. A. Kornhauser. Modeling collegial courts II. Legal doctrine. *Journal of Law, Economics and Organization*, 8: 441-470, 1992.

[10] L. A. Kornhauser and L. G. Sager. Unpacking the court. *Yale Law Journal*, 96: 82-117, 1986.

[11] L. A. Kornhauser and L. G. Sager. The one and the many: Adjudication in collegial courts. *California Law Review*, 81: 1-51, 1993.

[12] C. List. Judgment Aggregation: a Bibliography on the Discursive Dilemma, the Doctrinal Paradox and Decisions on Multiple Propositions, 2006. *http://personal.lse.ac.uk/LIST/doctrinalparadox.htm*

[13] C. List. The discursive dilemma and public reason. *Ethics*, 116(2): 362–402, 2006.

[14] P. Pettit. Deliberative democracy and the discursive dilemma. *Philosophical Issues*, 11: 268–299, 2001.

[15] G. Pigozzi. Belief merging and the discursive dilemma: an argument-based account to paradoxes of judgment aggregation. Forthcoming in *Synthese*, 2007.

Gabriella Pigozzi
ILIAS, Computer Science and Communications
University of Luxembourg
6, rue Richard Coudenhove - Kalergi
L-1359 Luxembourg
Email: `gabriella@pigozzi.org`

Stephan Hartmann
Department of Philosophy, Logic and Scientific Method
London School of Economics and Political Science
Houghton Street
London WC2A 2AE - UK
Email: S.Hartmann@lse.ac.uk

On the Robustness of Preference Aggregation in Noisy Environments

Ariel D. Procaccia, Jeffrey S. Rosenschein, and Gal A. Kaminka

Abstract

In an election held in a noisy environment, agents may unintentionally perturb the outcome by communicating faulty preferences. We investigate this setting by introducing a theoretical model of noisy preference aggregation and formally defining the (worst-case) robustness of a voting rule. We use our model to analytically bound the robustness of various prominent rules. Our results essentially specify the voting rules that allow for reasonable preference aggregation in the face of noise.

1 Introduction

Preference aggregation mechanisms, and voting rules in particular, have been the object of scientific study for many years. Such mechanisms are used to aggregate the preferences of human or synthetic agents, over alternatives (or candidates). The alternatives in question may be entities such as joint plans for execution, schedules [7], movie choices [5], etc. A voting rule generates an outcome that reflects the individual preferences over candidates, while striving to satisfy different desiderata. Indeed, much of the research in social choice theory has focused on formally analyzing the properties of social choice mechanisms, with respect to these desiderata.

One important feature of study in preference aggregation mechanisms is their resistance to manipulation. Such manipulations are instances of adversarial worst-cases in the context of mechanisms: they consider self-interested voters that intentionally cast untruthful ballots in order to manipulate the outcome in their favor. An important theorem asserts that every voting rule (under certain minimal assumptions) is manipulable [6, 9]. More recent work in computer science suggests that computational complexity may help circumvent this impossibility result [1, 3].

However, little attention has been paid to a simpler—and arguably more common—form of voting manipulation, where the truthful votes are *unintentionally* changed, as a result of uncertainty in the actions or perception of the agents casting the votes. For instance, agents may misunderstand the choices laid out for them, and may thus inadvertently cast a vote that is inconsistent with their true choice. Or, in the case of robots operating where communication is unreliable, true choices may be miscommunicated, resulting again in unintentional manipulation.

This paper takes first steps towards a formal analysis of the impact of errors in the preferences of voters. We define the k-robustness of a voting rule to be the resistance of the rule to k faults. In more detail, it is the probability that

the outcome changes as a result of the faults, when each fault is chosen independently at random. We analyze the connection between 1-robustness (i.e., resistance to a single fault) and k-robustness, and conclude that it is sufficient to examine the 1-robustness of different rules. Most importantly, we use our definitions and tools to give tight upper and lower bounds on the robustness of several prominent voting rules. In fact, we show that the robustness of voting rules is extremely diverse, with some rules positioned at both ends of the spectrum.

Given that voters rank the candidates (voters express *ordinal preferences*), we analyze a theoretical model where a fault is a switch in the rankings of two adjacent candidates (e.g., the fifth-ranked candidate is accidentally ranked sixth, and the sixth is ranked fifth). Such faults may easily be caused by confusion on the part of voters, or even by a single bit-flip when communicating the votes (see Section 3). Our goal is to understand the robustness of different voting rules to such faults; this understanding would aid system designers in selecting voting rules that can faithfully aggregate the preferences of agents in the system.

Previous work by Kalai [8] has investigated the issue of *noise-sensitivity* of social welfare functions in simple games; such functions give an entire social ranking of the candidates, instead of simply designating the winner of the election. The author engages in an asymptotic average-case analysis, where the basic assumption is that the voters' votes are distributed uniformly at random. Kalai presents a family of "chaotic" social welfare functions: a change in the preferences of a small fraction of the voters leads to social preferences that are asymptotically uncorrelated with the original preferences. In contrast, our model in this paper is quite different; in addition, we are interested in examining the robustness of prominent voting rules, as opposed to investigating extreme asymptotic phenomena.

This paper is organized as follows. In Section 2 we give an introduction to voting, and describe the voting rules we examine thereafter. In Section 3 we outline our model of preference profile errors, and give some general results regarding robustness. In Section 4 we bound the 1-robustness of some prominent voting rules, and in Section 5 we discuss our results and directions for future work.

2 Preliminaries

In this section we give a brief introduction to classic social choice theory. The information here is sufficient to understand the paper, but readers who are interested in more details can consult [2].

Let $V = \{v^1, v^2, \ldots, v^n\}$ be the set of voters, and let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of candidates, $|C| = m$. We usually use the index i (in superscript) to refer to voters, and the index j (in subscript) to refer to candidates.

Let $\mathcal{L} = \mathcal{L}(C)$ be the set of all linear orders[1] on C. Each voter has ordinal preferences $\succ^i \in \mathcal{L}$, i.e., each voter v^i ranks the candidates: $c_{j_1} \succ^i c_{j_2} \succ^i \cdots \succ^i c_{j_m}$. We refer to $\succ^V = \langle \succ^1, \ldots, \succ^n \rangle \in \mathcal{L}^N$ as a *preference profile*.

Given $\succ^i$, let $j_1, \ldots, j_m$ be indices of candidates such that $c_{j_1} \succ^i c_{j_2} \succ^i \cdots \succ^i c_{j_m}$; we denote by $\pi_l(\succ^i)$ the candidate that voter i ranks in the l'th place, i.e., $\pi_l(\succ^i) = c_{j_l}$. We also denote by l^i_j the ranking of c_j in $\succ^i$; it holds that $\pi_{l^i_j}(\succ^i) = c_j$.

2.1 Voting rules

A *voting rule* is a function $F : \mathcal{L}^V \to C$, i.e., a mapping from preferences of voters to candidates, which designates the winning candidate. We shall consider the following voting rules:

- *Scoring rules* are defined by a vector $\vec{\alpha} = \langle \alpha_1, \ldots, \alpha_m \rangle$.[2] Given $\succ \in \mathcal{L}^N$, the score of candidate j is $s_j = \sum_i \alpha_{l^i_j}$. The candidate that wins the election is $F(\succ) = \text{argmax}_j s_j$. Some of the well-known scoring rules are:
  - *Borda*: $\vec{\alpha} = \langle m-1, m-2, \ldots, 0 \rangle$.
  - *Plurality*: $\vec{\alpha} = \langle 1, 0, \ldots, 0 \rangle$.
  - *Veto*: $\vec{\alpha} = \langle 1, \ldots, 1, 0 \rangle$.
- *Copeland*: we say that candidate j beats j' in a pairwise election if $|\{i : l^i_j < l^i_{j'}\}| > n/2$. The score s_j of candidate j is the number of candidates that j beats in pairwise elections, and $\text{Copeland}(\succ) = \text{argmax}_j s_j$.
- *Maximin*: the Maximin score of candidate j is the candidate's worst performance in a pairwise election: $s_j = \min_{j'} |\{i : l^i_j < l^i_{j'}\}|$, and $\text{Maximin}(\succ) = \text{argmax}_j s_j$.
- *Bucklin*: for any candidate c_j and $l \in \{1, \ldots, m\}$, let $B_{j,l} = \{i : l^i_j \leq l\}$. It holds that $\text{Bucklin}(\succ) = \text{argmin}_j(\min\{l : |B_{j,l}| > n/2\})$.
- *Plurality with Runoff*: The election proceeds in two rounds. After the first round, only the two candidates that maximize $|\{i \in N : l^i_j = 1\}|$ survive. In the second round, a pairwise election is held between these two candidates.

3 Our Model of Faults and Robustness

We consider situations where (for example) noisy communication leads to changes in voters' rankings of candidates. The exact manifestation of these

[1] Binary relations that satisfy antisymmetry, transitivity, and totality.

[2] More formally, a scoring rule is defined by a sequence of such vectors, one for each value of m, but we abandon this formulation for clarity's sake.

faults largely depends on the representation of preferences. In order to obtain results that are as general as possible, we here simply regard a fault as an alteration of one voter's ordering of candidates, which nevertheless maintains the integrity of the voter's preferences as a linear ordering (other types of faults remain for future work).

Definition 1. A preference profile $\succ_1^V$ is obtained from a preference profile $\succ^V$ by an *elementary transposition* (write: $\succ^V \rightsquigarrow \succ_1^V$) if there exists a voter v^i and $l \in \{2, \ldots, m\}$ such that:

1. for all $i' \neq i$, $\succ^{i'} = \succ_1^{i'}$.
2. $\pi_l(\succ^i) = c = \pi_{l-1}(\succ_1^i)$.
3. $\pi_{l-1}(\succ^i) = c' = \pi_l(\succ_1^i)$.
4. $\succ^i \downarrow_{C \setminus \{c,c'\}} = \succ_1^i \downarrow_{C \setminus \{c,c'\}}$.

We say that $\pi_{l-1}(\succ^i)$ was *demoted* and $\pi_l(\succ^i)$ was *promoted*.

Example 1. The preference profile

$\succ^1$	$\succ^2$
c_1	c_2
c_3	c_1
c_2	c_3

is obtained from the preference profile

$\succ^1$	$\succ^2$
c_1	c_1
c_3	c_2
c_2	c_3

by an elementary transposition that promotes c_2 and demotes c_1 (in the notations of the definition, $i = 2$ and $l = 2$).

In other words, we focus here on faults where a switch has occurred between two adjacent candidates in a voter's ranking of candidates. Such faults are interesting from a theoretical perspective, but may also occur in practice. For instance, if voters build their preferences incrementally, they may easily be confused by spatial proximity of alternative candidates. In other cases, depending on the representation and communication protocol, communication errors may cause a switch to occur. Below we describe a representation for preferences, in which a flip of a single bit either causes a switch between two adjacent candidates, or can easily be detected.

3.1 The Pairwise Representation

We here describe a representation of preferences that is compatible with our fault model, and argue that it has some nice advantages. One can represent preferences using a bit for each ordered pair of candidates (with $\binom{m}{2}$ ordered

pairs): the bit is 1 if the first candidate is preferred to the second, and 0 otherwise. We shall refer to this representation as the *pairwise representation*. In this representation, a flip of a single bit corresponding to a pair of adjacent candidates in the ordering entails an elementary transposition. However, flipping a bit that does not correspond to adjacent candidates would create an ordering that is not transitive, and therefore not linear. Indeed, if (w.l.o.g.) $c_1 \succ c_2 \succ c_3$, and the bit corresponding to c_1 and c_3 is flipped, then we obtain the preferences $c_1 \succ c_2$, $c_2 \succ c_3$, and $c_3 \succ c_1$ — transitivity is not satisfied. It follows that faults which switch the ranking of two non-adjacent candidates can always be detected. So, when considering bit flips that may change the outcome without being detected, we can restrict our attention to faults that manifest themselves as elementary transpositions.

The pairwise representation is not the most compact possible. Consider the following elementary representation: each voter specifies the location of each candidate in their ranking; this requires $m \log m$ bits. In the pairwise representation, each voter requires $\binom{m}{2} = \frac{m(m-1)}{2}$ bits to express its preferences.

On the other hand, the pairwise representation allows us to test properties in constant time using bitmasks. For instance, say we want to test if a voter has ranked candidate c_1 highest. We construct the ordered pairs in a way that candidate c_1 is always first; we then examine the conjunction (a bitwise AND) of all pairs in which c_1 participates — c_1 is ranked first iff this conjunction is 1. This can be done in constant time, while in the elementary representation, one would have to examine all $\log m$ bits that represent c_1's ranking in order to answer this question. Similarly, given that one knows (from polls, for example) which candidates are placed in the first k places, one can test in constant time whether a candidate is ranked in place $k+1$, using a bitmask on the pairwise representation.

3.2 The Definition of Robustness

So far, we have described our model of faults, and have argued that it has practical justification. The switches in preferences that we consider may seem harmless, but in fact, for essentially any voting rule, there exist instances where even one switch changes the outcome of the election.

Theorem 1. *Let $F : \mathcal{L}^V \to C$ be a voting rule such that $Ran(F) > 1$. Then there exists a preference profile $\succ^V$ and a profile $\succ_1^V$ which is obtained from $\succ^V$ by an elementary transposition, such that $F(\succ^V) \neq F(\succ_1^V)$.*

Proof. Assume that for every preference profile $\succ^V$ and any elementary transposition, the outcome does not change. Let $\succ^V$ and $\succ_1^V$ be any two preference profiles; we will derive a contradiction to the assumption on F's range by showing that they necessarily have the same value under F.

Indeed, a preference profile is essentially a series of permutations on C (one for each voter); a basic result regarding permutation groups implies that $\succ_1^V$ can be obtained from $\succ^V$ by iterative elementary transpositions [4]. In other

words, there are $\succ_{i_1}^V, \ldots, \succ_{i_t}^V$ such that $\succ_{i_1}^V = \succ^V$, $\succ_{i_t}^V = \succ_1^V$, and each $\succ_{i_{j+1}}^V$ can be obtained from $\succ_{i_j}^V$ by an elementary transposition, for $j = 1, \ldots, t-1$. By our assumption, all $\succ_{i_j}^V$ have the same value under F, and in particular $F(\succ^V) = F(\succ_1^V)$ — a contradiction. □

Given a voting rule, we wish to consider the implications of faults *in the worst-case*, i.e., in the worst instance. Theorem 1 motivates a probabilistic analysis: we will calculate the probability of the faults affecting the outcome in the worst-case.

Given a preference profile $\succ^V$, we define the probability distribution $D_k(\succ^V)$ over preference profiles as follows: the probability of the preference profile $\succ_1^V$ is the probability of obtaining $\succ_1^V$ from $\succ^V$ by k elementary transpositions chosen independently and randomly. In other words, in order to draw a profile $\succ_1^V$ according to this distribution, we independently choose k values $\{l_1, l_2, \ldots, l_k\}$ and k values $\{i_1, \ldots, i_k\}$, where each l_j is chosen according to the uniform distribution over $\{2, \ldots, m\}$, and each i_j is chosen according to the uniform distribution over $\{1, \ldots, n\}$. Now, starting with $\succ^V$, we perform k successive elementary transpositions — the j'th transposition promotes candidate $\pi_l(\succ^i)$ and demotes $\pi_{l-1}(\succ^i)$.

Definition 2. The *k-robustness* of a preference profile $\succ^V$ is:

$$\rho(F, \succ^V) = \Pr_{\succ_1^V \sim D_k(\succ^V)}[F(\succ^V) = F(\succ_1^V)].$$

The k-robustness of a profile reflects its immunity to k independent faults. As our analysis is worst-case, in order to define the robustness of a voting rule we take the minimum over all instances:

Definition 3.

The *k-robustness* of a voting rule F with n voters and m candidates is:

$$\rho_k^{n,m}(F) = \min_{\succ^V \in \mathcal{L}(C)^n} \rho(F, \succ^V).$$

Example 2. Consider the Plurality rule with 3 voters and 2 candidates, and consider the preference profile $\succ^V$ given by:

$\succ^1$	$\succ^2$	$\succ^3$
c_1	c_1	c_2
c_2	c_2	c_1

The outcome of this election is c_1. There are three possible profiles resulting from an elementary transposition:

$\succ_1^1$	$\succ_1^2$	$\succ_1^3$
c_2	c_1	c_2
c_1	c_2	c_1

$\succ_2^1$	$\succ_2^2$	$\succ_2^3$
c_1	c_2	c_2
c_2	c_1	c_1

$\succ_3^1$	$\succ_3^2$	$\succ_3^3$
c_1	c_1	c_1
c_2	c_2	c_2

In two of these profiles, the outcome is c_2. Therefore, $\rho(F, \succ^V) = 1/3$. Repeating the same calculation for all preference profiles $\succ^V \in \mathcal{L}(C)^n$ and taking the minimum, it is possible to conclude that $\rho_1^{3,2}(\text{Plurality}) = 1/3$.

3.3 Bounding k-robustness with 1-robustness

The definition of $D_k(\succ^V)$ as sampling k independent elementary transpositions allows a very strong link between 1-robustness and k-robustness: a lower bound on the former entails a lower bound on the latter.

Proposition 2. $\rho_k^{n,m}(F) \geq (\rho_1^{n,m}(F))^k$.

Proof. Consider the preference profile $\succ_1^V$, and the preference profile $\succ_2^V$ obtained by k independent and random elementary transpositions — we claim that the probability that $F(\succ_1^V) = F(\succ_2^V)$ is at least $(\rho_1^{n,m})^k$.

Indeed, let $\succ_{i_1}^V, \ldots, \succ_{i_{k+1}}^V$ be the intermediate preference profiles obtained by the elementary transpositions, i.e., $\succ_{i_1}^V = \succ_1^V$, $\succ_{i_{k+1}}^V = \succ_2^V$, and each $\succ_{i_{j+1}}^V$ is obtained from $\succ_{i_j}^V$ by an independently and randomly chosen elementary transposition, for $j = 1, \ldots, k$. By the definition of 1-robustness, we have that for every preference profile $\succ^V$, the probability that one randomly chosen elementary transposition does not change the outcome of the election under F is at least $\rho_1^{n,m}(F)$. Therefore, we have that for $j = 1, \ldots, k$,

$$\Pr[F(\succ_{i_j}^V) = F(\succ_{i_{j+1}}^V) \mid \succ_{i_j}^V] \geq \rho_1^{n,m}(F).$$

By analyzing the conditional probabilities we have that:

$$\begin{aligned}
\Pr\left[F(\succ_1^V) = F(\succ_2^V)\right] &= \Pr\left[\forall j = 1, \ldots, k,\ F(\succ_{i_j}^V) = F(\succ_{i_{j+1}}^V)\right] \\
&= \prod_{j=1}^{k} \Pr\left[F(\succ_{i_j}^V) = F(\succ_{i_{j+1}}^V) \mid \succ_{i_j}^V\right] \\
&\geq (\rho_1^{n,m})^k .
\end{aligned}$$

□

The above proposition is very useful when the number of errors is constant. Otherwise, the bound on k-robustness which the proposition yields may not be very good, even if the voting rule seems 1-robust. Nevertheless, we have the following immediate corollary regarding $k = m$:

Corollary 3. *Let F be a voting rule such that $\rho_1^{n,m}(F) \geq 1 - x/m$ for some constant x, and let $\epsilon > 0$. Then $\rho_m^{n,m}(F) \geq \frac{1}{e^x} - \epsilon$ for a large enough m.*

4 Results on 1-Robustness

Proposition 2 dictates the direction of the bulk of our results: we are satisfied with calculating the 1-robustness of voting rules. If we achieve a high lower bound, this also implies high k-robustness (at least for a constant k). However, in case 1-robustness is low, there is no point in considering the rule's k-robustness.

Remark 1. Given the number of voters and candidates, and a preference profile $\succ^V$, there are exactly $n(m-1)$ possible elementary transpositions. Therefore:

$$\rho_1^{n,m}(F) = \frac{|\{\succ_1^V \in \mathcal{L}(C)^n : \succ^V \rightsquigarrow \succ_1^V \ \wedge \ F(\succ^V) = F(\succ_1^V)\}|}{n(m-1)}.$$

Before we deal with specific voting rules, we note that we cannot expect a rule's 1-robustness to be exactly 1.

Proposition 4. *Let $F : \mathcal{L}(C)^n \to C$ be a voting rule such that $Ran(F) > 1$. Then $\mu_1^{n,m}(F) < 1$.*

Proof. Follows directly from Proposition 1 and the definition of 1-robustness. □

4.1 Scoring rules

In this subsection we fully characterize the robustness of scoring rules as a function of their parameters. Our results imply that some common scoring rules are very robust, while others are extremely susceptible to faults.

Given a scoring rule F with parameters $\vec{\alpha}$, let $A_F = |\{l \in \{2, \ldots, m\} : \alpha_{l-1} > \alpha_l\}|$; denote $|A_F| = a_F$.

Proposition 5. *Let n and m be the number of voters and candidates, let F be a scoring rule. Then $\rho_1^{n,m}(F) \geq \frac{m-1-a_F}{m-1}$.*

Proof. For any preference profile $\succ^V$, the outcome can only be affected by elementary transpositions that promote $\pi_l(\succ^i)$, for some $l \in A_F$ and i, and demote $\pi_{l-1}(\succ^i)$. For each voter v^i, there are exactly a_F such values of l, out of $m-1$ possible elementary switches. Therefore, the number of elementary transpositions that are guaranteed *not* to change the outcome is at least $n(m-1) - a_F n$, and the 1-robustness of F is at least $\frac{n(m-1)-a_F n}{n(m-1)} = \frac{m-1-a_F}{m-1}$. □

We match this lower bound with a pretty tight upper bound. In this example, we require that the number of candidates divide the number of voters. However, such a special case is sufficient, as it implies that the lower bound cannot be improved in general.

Proposition 6. *Let n and m be the number of voters and candidates such that m divides n, and let F a scoring rule. Then $\rho_1^{n,m}(F) \leq \frac{m-a_F}{m}$.*

Proof. By the assumption on n and m, it is possible to group the voters in m subsets of size d, $T_1, \ldots, T_m$. Consider the preference profile $\succ^V$ where the subsets of voters vote cyclically:

$\succ^{T_1}$	$\succ^{T_2}$	.	.	$\succ^{T_m}$
c_1	c_2	.	.	c_m
c_2	c_3	.	.	c_1
.	.	.	.	.
.	.	.	.	.
c_{m-1}	c_m	.	.	c_{m-2}
c_m	c_1	.	.	c_{m-1}

Notice that under any scoring rule, all candidates have the same score; without loss of generality candidate c_1 is the winner of this election. How many profiles obtained by a single transposition necessarily have a different outcome? An elementary transposition between places $l-1$ and l, where $l \in A_F$, strictly increases a candidate's score, and changes the outcome — given that the promoted candidate is not candidate 1. For every $l \in A_F$, exactly d voters rank candidate c_1 in place l. Hence, there are da_F voters with $a_F - 1$ possible elementary transpositions that change the outcome of the election (the voters that rank candidate c_1 in place $l \in A_F$), and $n - da_F$ voters with a_F such transpositions. It follows that the probability that the outcome changes, under the uniform distribution over instances such that $\succ^V \rightsquigarrow \succ_1^V$, is at least (substituting $n = md$):

$$\frac{da_F(a_F-1) + (dm - da_F)a_F}{dm(m-1)} = \frac{a_F}{m}.$$

In other words, the probability that the outcome *does not change* is at most $\frac{m-a_F}{m}$. As the robustness is defined to be the minimum over all instances, we obtain the desired result. □

We conclude that the Veto and Plurality rules, where $a_F = 1$, are extremely robust. On the other hand, the Borda rule, for which $a_F = m - 1$, is very susceptible to failures.

4.2 Copeland

We give an upper bound that relies on an example where the number of voters is even. However, since the number of candidates is not restricted, this example implies that it is not possible to establish a good general lower bound. In addition, as the upper bound is very small, an exact lower bound is of no consequence.

Proposition 7. *Let m be the number of candidates, and let the number of voters n be even. Then $\rho_1^{n,m}(Copeland) \leq 1/(m-1)$.*

Proof. Consider the preference profile where for $i = 1, 3, 5, \ldots, n-1$, voters v^i and v^{i+1} vote as follows:

$\succ^i$	$\succ^{i+1}$
c_1	c_m
c_2	c_{m-1}
.	.
.	.
c_m	c_1

Under the above profile, for every two candidates c and c', exactly $n/2$ voters prefer c over c'. Thus, the Copeland score of all candidates is 0, and the winner is some candidate $c \in C$. Any elementary transposition that promotes candidate $c' \neq c$ would raise the score of c' to 1, making c' the new winner. This implies that for every voter, there are at least $m-2$ elementary transpositions that change the outcome of the election, and thus the probability that the outcome does not change is at most $1 - \frac{n(m-2)}{n(m-1)} = \frac{1}{m-1}$. □

4.3 Maximin

Proposition 8. *Let n and m be the number of voters and candidates such that m divides n. Then $\rho_1^{n,m}(\mathit{Maximin}) \leq 1/(m-1)$.*

Proof. Our adversarial preference profile is identical to the one in the proof of Proposition 6. However, we are going to construct the profile algorithmically, as this is going to aid us in establishing some of the profile's properties. We iteratively expand the list of candidates; initially, it contains only c_1, so each voter's linear preferences are in fact the empty set. In the second stage, we add to the slate the candidate c_2; for $\frac{1}{m}n$ voters, candidate c_2 is ranked at the top (above c_1), but the other $\frac{m-1}{m}n$ voters rank c_2 below c_1. Now, c_3 is added as follows: $\frac{1}{m}n$ voters that ranked c_2 last (i.e., previously voted $c_1 \succ c_2$), now rank c_3 first (i.e., vote $c_3 \succ c_1 \succ c_2$); the other $\frac{m-1}{m}n$ voters rank c_3 immediately below c_2 (e.g., if the ranking was $c_2 \succ c_1$, it is now $c_2 \succ c_3 \succ c_1$). In general, when adding candidate c_j, $\frac{1}{m}n$ voters that ranked c_{j-1} last now rank c_j first,[3] and the rest rank c_j just below c_{j-1}.

For example, for 8 voters and 4 candidates, initially we have: (in each stage j, the $\frac{1}{m}n = 2$ grayed voters are the ones that rank candidate c_j first instead of just under c_{j-1})

$\succ^1$	$\succ^2$	$\succ^3$	$\succ^4$	$\succ^5$	$\succ^6$	$\succ^7$	$\succ^8$
c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1

In the second stage we have:

$\succ^1$	$\succ^2$	$\succ^3$	$\succ^4$	$\succ^5$	$\succ^6$	$\succ^7$	$\succ^8$
c_2	c_2	c_1	c_1	c_1	c_1	c_1	c_1
c_1	c_1	c_2	c_2	c_2	c_2	c_2	c_2

In the third stage we have:

[3] It is easy to verify that there always are $\frac{1}{m}n$ such voters.

$\succ^1$	$\succ^2$	$\succ^3$	$\succ^4$	$\succ^5$	$\succ^6$	$\succ^7$	$\succ^8$
c_2	c_2	c_3	c_3	c_1	c_1	c_1	c_1
c_3	c_3	c_1	c_1	c_2	c_2	c_2	c_2
c_1	c_1	c_2	c_2	c_3	c_3	c_3	c_3

Ultimately, the preference profile that the algorithm constructs is:

$\succ^1$	$\succ^2$	$\succ^3$	$\succ^4$	$\succ^5$	$\succ^6$	$\succ^7$	$\succ^8$
c_2	c_2	c_3	c_3	c_4	c_4	c_1	c_1
c_3	c_3	c_4	c_4	c_1	c_1	c_2	c_2
c_4	c_4	c_1	c_1	c_2	c_2	c_3	c_3
c_1	c_1	c_2	c_2	c_3	c_3	c_4	c_4

Lemma 9. *In stage j (after candidate c_j is added to the slate), it holds that for every $i < j$, the number of voters that prefer c_i to c_j is $\frac{m-(j-i)}{m}n$.*

Proof. By induction on j. The basis of the induction ($j = 1$) is trivial. Now, assume the claim holds for $j-1$; we shall prove it for j. Let $i < j$; if $i = j-1$, notice that c_j is ranked under c_i, except in $\frac{1}{m}n$ cases. In other words, the number of voters that prefer $c_i = c_{j-1}$ to c_j is $\frac{m-1}{m}n$, as desired.

It remains to deal with the case where $i < j-1$. Recall that c_j is always ranked directly under c_{j-1}, except for $\frac{1}{m}n$ voters that rank c_j first. As for the rest of the voters, c_i is ranked above c_j iff c_i was ranked above c_{j-1} in stage $j-1$. By the induction assumption, we had $\frac{m-((j-1)-i)}{m}n$ ranking c_i above c_{j-1} in stage $j-1$, and thus the number of voters ranking c_i above c_j is:

$$\frac{m-((j-1)-i)}{m}n - \frac{1}{m}n = \frac{m-(j-i)}{m}n,$$

as desired. □

Lemma 9 implies that candidate c_j's *unique* worst pairwise election is against c_{j-1} for $j > 1$: the number of voters that prefer c_j to c_{j-1} is exactly $\frac{1}{m}n$; notice that this is also true for c_1 versus c_m: only $\frac{1}{m}n$ rank c_1 above c_m. In addition, for $j > 1$, clearly c_j is ranked just under c_{j-1} by $\frac{m-1}{m}n$ voters — but this, too, is also true for c_1 versus c_m; indeed, c_1 is ranked just under c_m by all $\frac{m-1}{m}n$ voters that do not rank c_1 first.

So, the candidates are all tied with respect to their maximin scores, and each candidate c_j is ranked just below its "worst pairwise" candidate by all voters that do not rank c_j first. Therefore, any elementary transposition that promotes a candidate that is not the current winner of the election must change the outcome of the election. As before, we have that the probability of the outcome changing as a result of a single transposition, under our adversarial preference profile, is at least $\frac{m-2}{m-1}$, and thus robustness of this preference profile is at most $\frac{1}{m-1}$. □

4.4 Bucklin

Proposition 10. *$\rho_1^{n,m}(Bucklin) \geq \frac{m-2}{m-1}$ for any values of the number of voters n and the number of candidates m.*

Proof. Consider a preference profile $\succ^V$, and assume that the winner c_j of the election satisfies: $l_0 = \min_l B(j,l) > n/2$. We argue that any elementary transposition that switches the candidates in places l and $l-1$, for $l \neq l_0, l_0+1$, cannot change the outcome of the election. Indeed, we consider two cases:

Case 1: $l > l_0 + 1$. In this case, if some candidate $c_k \neq c_j$ is promoted, the switch increases $B(k, l-1)$ — but this is irrelevant to the outcome of the election, since $B(k, l_1)$ remains unchanged for $l_1 \leq l_0$.

Case 2: $l < l_0$. If candidate c_k is promoted, this might increase $B(k, l_0 - 2)$. However, the value of $B(k, l_0 - 2)$ after the switch took place is bounded from above by the value of $B(k, l_0 - 1)$ before the switch. We know that $B(k, l_0 - 1) \leq n/2$ before the switch — so this transposition is not going to affect the value of $\min_k(B(k,l) > n/2)$.

If so, it remains to consider the case where $l = l_0$ or $l = l_0 + 1$. When $l = l_0$, promoting $\pi_{l_0}(\succ^i)$ may affect the outcome only if $\pi_{l_0}(\succ^i) \neq c_j$, where c_j is the winner of the election. However, when $l = l_0 + 1$, promoting $\pi_{l_0+1}(\succ^i)$ and demoting $\pi_{l_0}(\succ^i)$ might affect the outcome only if $\pi_{l_0}(\succ^i) = c_j$. Otherwise, if $\pi_{l_0+1}(\succ^i) = c_k \neq c_j$, then $B(k, l_0)$ might be affected, but since c_j already has a majority of voters ranking it in the top l_0 places, the outcome of the election is indifferent to this perturbation.

As these two last subcases are mutually exclusive, it follows that for every voter there is at most one transposition that may affect the outcome of the election. Thus $\rho_1^{n,m}(\text{Bucklin}) \geq \frac{m-2}{m-1}$. □

4.5 Plurality with Runoff

The rules we have discussed in the previous subsections all have in common some concept of score. Since Plurality with Runoff is a bit different, we require an additional assumption regarding tie-breaking. Consider a situation where, say, c_{j_1} and c_{j_2} survive the first round, and exactly half the voters prefer c_{j_1} to c_{j_2}, but c_{j_1} is the winner of the election. We assume that if a fault makes c_{j_1} and c_{j_3} survive the first round, and again these two candidates are tied in the second round, then c_{j_1} *loses* the election. This assumption is consistent with our worst-case analysis throughout.

Proposition 11. *For all values of n and m, $\rho_1^{n,m}(Plurality\ with\ Runoff) \geq \frac{m-5/2}{m-1}$.*

Proof. Consider some preference profile $\succ^V$, and assume w.l.o.g. that candidates c_1 and c_2 survive the first round, and c_1 wins the election. Only two types of elementary transposition can potentially affect the outcome of the election. The first is promoting the candidate $\pi_2(\succ^i)$ for some i, i.e., making this candidate voter v^i's favorite — this might affect the list of candidates that

Function	Lower Bound	Upper Bound
Scoring	$\frac{m-1-a_F}{m-1}$	$\frac{m-a_F}{m}$
Copeland	0	$\frac{1}{m-1}$
Maximin	0	$\frac{1}{m-1}$
Bucklin	$\frac{m-2}{m-1}$	1
Plurality w. Runoff	$\frac{m-5/2}{m-1}$	$\frac{m-5/2}{m-1} + \frac{5/2}{m(m-1)}$

Table 1: Upper and lower bounds on the 1-robustness of several prominent voting rules.

are eliminated in the first round. A second transposition which might have an effect is one that promotes candidate c_2 and demotes c_1 — this might change the outcome of the second round, but only if exactly half the voters prefer c_1 to c_2 in $\succ^V$ (it cannot be the case that more voters prefer c_2, as then c_2 would have prevailed in the second round). To conclude, at most $n/2$ voters have two transpositions that may affect the outcome, and at least $n/2$ voters have only one. We have that

$$\rho(F, \succ^V) \geq \frac{n(m-1) - (n/2 \cdot 1 + n/2 \cdot 2)}{n(m-1)} = \frac{m - 5/2}{m-1}.$$

□

Proposition 12. $\rho_1^{2m,m}(\textit{Plurality with Runoff}) \leq \frac{m-5/2}{m-1} + \frac{5/2}{m(m-1)}$.

Proof. Omitted due to space constraints. □

5 Discussion

We have defined the k-robustness of a voting rule as the worst-case probability that k independent switches in the preferences of voters change the outcome of the election. We have shown that high 1-robustness implies high k-robustness, at least for a constant k. Inversely, low 1-robustness clearly suggests that the rule is not robust in general. Accordingly, we have presented bounds on the 1-robustness of different voting rules; these bounds are summarized in Table 5.

We intend our results to be used as a tool for designers of multiagent systems. When dealing with noisy environments, successful aggregation of preferences can only be expected when a robust voting rule is applied. In particular, among the prominent voting rules, our results imply that Plurality, Plurality with Runoff, Veto, and Bucklin are robust to faults, whereas Borda, Copeland, and Maximin are susceptible to faults.

The model of errors we have introduced is a theoretical one, but we have also shown it is grounded in a reasonable representation of preferences. Nevertheless, future work should include an investigation of different error models.

In addition, our analysis was worst-case — an approach which leads to the conclusion that when the number of errors is large, voting rules are bound to fail. It would be interesting to complement our results with an asymptotic *average-case* analysis.

References

[1] J. Bartholdi, 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 P. C. Fishburn. Voting procedures. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, chapter 4. North-Holland, 2002.

[3] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence*, pages 314–319, 2002.

[4] J. D. Dixon and B. Mortimer. *Permutation Groups*. Springer, 1996.

[5] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: the anatomy of a recommender system. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 434–435, 1999.

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

[7] T. Haynes, S. Sen, N. Arora, and R. Nadella. An automated meeting scheduling system that utilizes user preferences. In *Proceedings of the First International Conference on Autonomous Agents*, pages 308–315, 1997.

[8] G. Kalai. Noise sensitivity and chaos in social choice theory. Preprint, 2005.

[9] 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.

Ariel D. Procaccia and Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Givat Ram, Jerusalem 91904, Israel
Email: {arielpro,jeff}@cs.huji.ac.il

Gal A. Kaminka
Computer Science Department
Bar Ilan University
Ramat Gan 52900, Israel
Email: galk@cs.biu.ac.il

Automated Design of Voting Rules by Learning from Examples

Ariel D. Procaccia, Aviv Zohar, and Jeffrey S. Rosenschein

Abstract

While impossibility results have established that no perfect voting rules exist, efficiently designing a voting rule that satisfies at least a given subset of desiderata remains a difficult task. We argue that such custom-built voting rules can be constructed by learning from examples. Specifically, we consider the learnability of the broad, concisely-representable class of scoring rules. Our main result asserts that this class is efficiently learnable in the PAC model. We also discuss the limitations of our approach, and (along the way) we establish a lemma of independent interest regarding the number of distinct scoring rules.

1 Introduction

Voting is a well-studied method of preference aggregation, in terms of its theoretical properties, as well as its computational aspects [3, 2]; various practical, implemented applications exist [9, 8]. In an election, a set of n voters express their preferences over a set of m candidates or alternatives. To be precise, each voter is assumed to reveal linear preferences — a ranking of the candidates. The outcome of the election is determined according to a *voting rule*.

1.1 Scoring Rules

The predominant — ubiquitous, even — voting rule in real-life elections is the *Plurality* rule. Under Plurality, each voter awards one point to the candidate it ranks first, i.e., its most preferred alternative. The candidate that accumulated the most points, summed over all voters, wins the election. Another example of a voting rule is the *Veto* rule: each voter "vetoes" a single candidate; the candidate that was vetoed by the fewest voters wins the election. Yet a third example is the *Borda* rule: every voter awards $m-1$ points to its top-ranked candidate, $m-2$ points to its second choice, and so forth — the least preferred candidate is not awarded any points. Once again, the candidate with the most points is elected.

The abovementioned three voting rules all belong to an important family of voting rules known as *scoring rules*. A scoring rule can be expressed by a vector of parameters $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_m \rangle$, where each α_l is a real number and $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$. Each voter awards α_1 points to its most-preferred alternative, α_2 to its second-most-preferred alternative, etc. Predictably, the candidate with the most points wins. Under this unified framework, we can express our three rules as:

	Majority	Robustness	Manipulation	Communication
Plurality	Yes	$\geq \frac{m-2}{m-1}$ [11]	$\mathcal{P}$ [3]	$\Theta(n \log m)$ [6]
Borda	No	$\leq \frac{1}{m}$ [11]	$\mathcal{NP}$-complete [3]	$\Theta(nm \log m)$ [6]
Veto	No	$\geq \frac{m-2}{m-1}$ [11]	$\mathcal{NP}$-complete [2]	?

Table 1: Different scoring rules greatly differ in the desiderata they satisfy.

- *Plurality*: $\vec{\alpha} = \langle 1, 0, \ldots, 0 \rangle$.
- *Borda*: $\vec{\alpha} = \langle m-1, m-2, \ldots, 0 \rangle$.
- *Veto*: $\vec{\alpha} = \langle 1, \ldots, 1, 0 \rangle$.

1.2 Motivation

Voting rules are often compared on the basis of different criteria, which define potentially desirable properties. We outline below several important criteria, some theoretical, and some computational.

1. *Majority*: If there is a candidate that is most preferred by a majority of voters, does this candidate win the election?
2. *Robustness* [11]: What is the worst-case probability of the outcome of the election *not* changing as a result of a random mistake/fault in the preferences of the voters?
3. *Complexity of Manipulation*: Say a coalition of voters aims to improve its utility from the election by voting untruthfully. How computationally difficult is it to find an optimal vote?
4. *Communication Complexity*: How much communication is required in order to determine the winner of the election?

Impossibility theorems imply that one cannot expect one voting rule to satisfy all desirable criteria simultaneously. However, different voting rules, satisfy different subsets of criteria. In particular, scoring rules greatly differ in this respect. To put it differently, different choices of the parameters of a scoring rule yield significantly different voting rules in terms of their properties. As an example, Table 1 compares Plurality, Borda, and Veto, on the basis of the abovementioned four properties.

1.3 Our Approach

So, how would one go about designing a scoring rule with certain properties, configuring the parameters to one's needs? In this paper, we do so by learning from examples. The basic setup is as follows: the designer, or teacher, is presented with different constellations of voters' preferences, drawn according to a fixed distribution. For each such preference profile, the teacher answers

with the winning candidate. For example, if the designer wishes the voting rule to satisfy the majority criterion, and is presented with a profile where a candidate is ranked first by a majority of voters, the designer would answer that this candidate is the winner. More generally, it is possible to consider a setting where properties are represented by tables; for each preference profile, the table designates the set of possible winning candidates (candidates that do not violate the desired property). If a voting rule is to satisfy a given combination of properties, then the winner chosen in every profile is a candidate in the intersection of the different sets of possible winners.

Assuming that there exists a *target* scoring rule that meets all the requirements, we would like to produce a scoring rule that is as "close" as possible. This way, the designer could in principle translate the above cumbersome representation of possible winners using tables, to a concisely-represented voting rule that can be easily understood and computed.

By "close" we mean close with respect to the fixed distribution over preference profiles. More precisely, we would like to construct an algorithm that receives pairs of the form (preferences, winner) drawn according to a fixed distribution D over preferences, and outputs a scoring rule, such that the probability according to D that our scoring rule and the target rule agree is as high as possible. Some readers may have realized that, in fact, we wish to learn scoring rules in the framework of the formal learning model — the PAC (Probably Approximately Correct) model; a concise introduction to this model is given in Section 2.

The dimension of a function class is a combinatorial measure of the richness of the class. The dimension of a class is closely related to the number of examples needed to learn it. We give tight bounds on the dimension of the class of scoring rules: an upper bound of m, and a lower bound of $m - 3$, where m is the number of candidates in an election. In addition, we show that, given a set of examples, one can efficiently construct a scoring rule that is consistent with the examples, if one exists. Combined, these results imply that the class of scoring rules is efficiently learnable. In other words, given a combination of properties which is satisfied by some scoring rule, it is possible to construct a "close" scoring rule in polynomial time.

The main weakness of our approach is that there might be cases where there is no scoring rule that satisfies a given combination of properties, although there is a voting rule that does. In this case, there might not exist a scoring rule which is consistent with the given training set. We discuss the limitations of our approach, showing the there are voting rules which cannot be "approximated" by scoring rules. Along the way, we show that the number of distinct scoring rules is at most exponential in the number of voters and candidates (whereas the number of voting rules is double exponential).

1.4 Related Work

To the best of our knowledge, we are the first to study automated design of voting rules, and the first to suggest learning as a method of designing social choice mechanisms (although learning is known to be useful in economic settings; PAC learning has very recently been applied to computing utility functions that are rationalizations of given sequences of prices and demands [1]).

Conitzer and Sandholm [4] have studied automated mechanism design, in the more restricted setting where agents have numerical valuations for different alternatives. They propose automatically designing a truthful mechanism for every preference aggregation setting. However, they find that, under two solution concepts, even determining whether there exists a deterministic mechanism that guarantees a certain social welfare is an $\mathcal{NP}$-complete problem. The authors also show that the problem is tractable when designing a randomized mechanism. In more recent work [5], Conitzer and Sandholm put forward an efficient algorithm for designing deterministic mechanisms, which works only in very limited scenarios.

In short, our setting, goals, and methods are completely different — in the general voting context, even framing computational complexity questions is problematic, since the goal cannot be specified with reference to expected social welfare.

1.5 Structure of the Paper

In Section 2 we give an introduction to the PAC model. In Section 3, we describe our setting and rigorously prove that the class of scoring rules is efficiently learnable. In Section 4, we discuss the limitations of our approach, and in Section 5, we give our conclusions.

2 Preliminaries

In this section we give a very short introduction to the PAC model and the generalized dimension of a function class. A more comprehensive (and slightly more formal) overview of the model, and results concerning the dimension, can be found in [10].

In the PAC model, the learner is attempting to learn a function $f : X \to Y$, which belongs to a class $\mathcal{F}$ of functions from X to Y. The learner is given a *training set* — a set of points in X, $x_1, x_2, \ldots, x_t$, which are sampled i.i.d. (independently and identically distributed) according to a distribution D over the sample space X. D is unknown, but is fixed throughout the learning process. In this paper, we assume the "realizable" case, where a target function $f^*(x)$ exists, and the given training examples are in fact labeled by the target function: $\{(x_k, f^*(x_k))\}_{k=1}^t$. The *error* of a function $f \in \mathcal{F}$ is defined as

$$err(f) = \Pr_{x \sim D}[f(x) \neq f^*(x)]. \tag{1}$$

$\epsilon > 0$ is a parameter given to the learner that defines the *accuracy* of the learning process: we would like to achieve $err(h) \leq \epsilon$. Notice that $err(f^*) = 0$. The learner is also given an *accuracy* parameter $\delta > 0$, that provides an upper bound on the probability that $err(h) > \epsilon$:

$$\Pr[err(h) > \epsilon] < \delta. \tag{2}$$

We now formalize the discussion above:

Definition 1.

1. *A* learning algorithm *L is a function from the set of all training examples to $\mathcal{F}$ with the following property: given $\epsilon, \delta \in (0,1)$ there exists an integer $s(\epsilon, \delta)$ — the* sample complexity *— such that for any distribution D on X, if Z is a sample of size at least s where the samples are drawn i.i.d. according to D, then with probability at least $1 - \delta$ it holds that $err(L(Z)) \leq \epsilon$.*

2. *L is an* efficient *learning algorithm if it always runs in time polynomial in $1/\epsilon$, $1/\delta$, and the size of the representations of the target function, of elements in X, and of elements in Y.*

3. *A function class $\mathcal{F}$ is* (efficiently) PAC-learnable *if there is an (efficient) learning algorithm for $\mathcal{F}$.*

The sample complexity of a learning algorithm for $\mathcal{F}$ is closely related to a measure of the class's combinatorial richness known as the generalized dimension.

Definition 2. Let $\mathcal{F}$ be a class of functions from X to Y. We say $\mathcal{F}$ *shatters* $S \subseteq X$ if there exist two functions $g, h \in \mathcal{F}$ such that

1. For all $x \in S$, $g(x) \neq h(x)$.

2. For all $S_1 \subseteq S$, there exists $f \in \mathcal{F}$ such that for all $x \in S_1$, $f(x) = h(x)$, and for all $x \in S \setminus S_1$, $f(x) = g(x)$.

Definition 3. Let $\mathcal{F}$ be a class of functions from a set X to a set Y. The *generalized dimension* of $\mathcal{F}$, denoted by $D_G(\mathcal{F})$, is the greatest integer d such that there exists a set of cardinality d that is shattered by $\mathcal{F}$.

The generalized dimension of a function provides both upper and lower bounds on the sample complexity of algorithms.

Theorem 1. *[10, Theorem 5.1] Let $\mathcal{F}$ be a class of functions from X to Y of generalized dimension d. Let L be an algorithm such that, when given a set of t labeled examples $\{(x_k, f^*(x_k))\}_k$ of some $f^* \in \mathcal{F}$, sampled i.i.d. according to some fixed but unknown distribution over the instance space X, produces an*

output $f \in \mathcal{F}$ that is consistent with the training set. Then L is an (ϵ, δ)-learning algorithm for $\mathcal{F}$ provided that the sample size obeys:

$$s \geq \frac{1}{\epsilon}\left((\sigma_1 + \sigma_2 + 3)D_G(\mathcal{F})\ln 2 + \ln\left(\frac{1}{\delta}\right)\right) \tag{3}$$

where σ_1 and σ_2 are the sizes of the representation of elements in X and Y, respectively.

Theorem 2. *[10, Theorem 5.2] Let $\mathcal{F}$ be a function class of generalized dimension $d \geq 8$. Then any (ϵ, δ)-learning algorithm for $\mathcal{F}$, where $\epsilon \leq 1/8$ and $\delta < 1/4$, must use sample size $s \geq \frac{d}{16\epsilon}$.*

3 Learning Scoring rules

Before diving in, we introduce some notation. Let $N = \{1, 2, \ldots, n\}$ be the set of voters, and let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of candidates. Let $\mathcal{L}$ be the set of linear preferences[1] over C; each voter has preferences $\succ^i \in \mathcal{L}$. We denote the *preference profile*, consisting of the voters' preferences, by $\succ^N = \langle \succ^1, \succ^2, \ldots, \succ^n \rangle$.

Let $\vec{\alpha}$ be a vector of real numbers such that $\alpha_l \geq \alpha_{l+1}$ for all $l = 1, \ldots, m-1$. Let $f_{\vec{\alpha}} : \mathcal{L}^N \to C$ be the scoring rule defined by the vector $\vec{\alpha}$, i.e., each voter awards α_l points to the candidate it ranks in the l'th place, and the rule elects the candidate with the most points.

Since several candidates may have maximal scores in an election, we must adopt some method of tie-breaking. Our method works as follows: ties are broken in favor of the candidate that was ranked first by more voters; if several candidates have maximal scores and were ranked first by the same number of voters, the tie is broken in favor of the candidate that was ranked second by more voters; and so on.[2]

Let $\mathcal{S}_m^n$ be the class of scoring rules with n voters and m candidates. Our goal is to learn, in the PAC model, some target function $f_{\vec{\alpha}^*} \in \mathcal{S}_m^n$. To this end, the learner receives a training set $\{(\succ_k^N, f_{\vec{\alpha}^*}(\succ_k^N)\}_k$, where each $\succ_k^N$ is drawn from a fixed distribution over $\mathcal{L}^N$; let $c_{j_k} = f_{\vec{\alpha}^*}(\succ_k^N)$. For the profile $\succ_k^N$, we denote by $\pi_{j,l}^k$ the number of voters that ranked candidate c_j in place l. Notice that candidate c_j's score under the preference profile $\succ_k^N$ is $\sum_l \pi_{j,l}^k \alpha_l$.

Our main goal in this section is to prove the following theorem.

Theorem 3. *For all $n, m \in \mathbb{N}$, the class $\mathcal{S}_m^n$ is efficiently PAC-learnable.*

By Theorem 1, in order to prove Theorem 3 it is sufficient to validate the following two claims: that there exists an algorithm which, for any training set, runs in time polynomial in the size of the training set and in n, m, and outputs

[1] A binary relation which is antisymmetric, transitive, and total.

[2] In case several candidates have maximal scores and identical rankings everywhere, break ties arbitrarily — say, in favor of the candidate with the smallest index.

a scoring rule which is consistent with the training set (assuming one exists); and that the generalized dimension of the class $\mathcal{S}_m^n$ is polynomial in n and m.

It is rather straightforward to construct an efficient algorithm that outputs consistent scoring rules. Given a training set, we must choose the parameters of our scoring rule in a way that, for any example, the score of the designated winner is at least as large as the scores of other candidates. Moreover, if ties between the winner and a loser would be broken in favor of the loser, then the winner's score must be strictly higher than the loser's. Our algorithm, given as Algorithm 1, simply formulates all the constraints as linear inequalities, and solves the resulting linear program.

Algorithm 1 Given a training set, the algorithm returns a scoring rule which is consistent with the given examples, if one exists.

for $k \leftarrow 1 \dots t$ **do**
  $C_k \leftarrow \emptyset$
  for all $j \neq j_k$ **do** ▷ c_{j_k} is the winner in example k
    $\vec{\pi}^{\Delta} \leftarrow \vec{\pi}^k_{j_k} - \vec{\pi}^k_j$
    $l_0 \leftarrow \min\{l : \ \pi^{\Delta}_l \neq 0\}$
    if $\pi^{\Delta}_{l_0} < 0$ **then** ▷ Ties are broken in favor of c_j
      $C_k \leftarrow C_k \cup \{c_j\}$
    end if
  end for
end for
return a feasible solution $\vec{\alpha}$ to the following linear program:

$$\begin{array}{l} \forall k, \ \forall c_j \in C_k, \sum_l \pi^k_{j_k,l}\alpha_l > \sum_l \pi^k_{j,l}\alpha_l \\ \forall k, \ \forall c_j \notin C_k, \sum_l \pi^k_{j_k,l}\alpha_l \geq \sum_l \pi^k_{j,l}\alpha_l \\ \forall l = 1, \dots, m-1 \quad \alpha_l \geq \alpha_{l+1} \\ \forall l, \ \alpha_l \geq 0 \end{array}$$

A linear program can be solved in time that is polynomial in the number of variables and inequalities [12]; it follows that Algorithm 1's running time is polynomial in n, m, and the size of the training set.

So, it remains to demonstrate that the generalized dimension of $\mathcal{S}_m^n$ is polynomial in n and m. The following lemma shows this.

Lemma 4. *The generalized dimension of the class S_m^n is at most m:*

$$D_G(S_m^n) \leq m.$$

Proof. According to Definition 3, we need to show that any set of cardinality $m+1$ cannot be shattered by $\mathcal{F}$. Let $S = \{\succ_k^N\}_{k=1}^{m+1}$ be such a set, and let h, g be the two social choice functions that disagree on all preference profiles in S.

We shall construct a subset $S_1 \subseteq S$ such that there is no scoring rule $f_{\vec{\alpha}}$ that agrees with h on S_1 and agrees with g on $S \setminus S_1$.

Let us look at the first preference profile from our set, $\succ_1^N$. We shall assume without loss of generality that $h(\succ_1^N) = c_1$, while $g(\succ_1^N) = c_2$, and that in $\succ_1^N$ ties are broken in favor of c_1. Let $\vec{\alpha}$ be some parameter vector. If we are to have $h(\succ_1^N) = f_{\vec{\alpha}}(\succ_1^N)$, it must hold that

$$\sum_{l=1}^{m} \pi_{1,l}^1 \cdot \alpha_l \geq \sum_{l=1}^{m} \pi_{2,l}^1 \cdot \alpha_l, \tag{4}$$

whereas if we wanted $f_{\vec{\alpha}}$ to agree with g we would want the opposite:

$$\sum_{l=1}^{m} \pi_{1,l}^1 \cdot \alpha_l < \sum_{l=1}^{m} \pi_{2,l}^1 \cdot \alpha_l \tag{5}$$

More generally, we define, with respect to the profile $\succ_k^N$, the vector $\vec{\pi}_\Delta^k$ as the vector whose l'th coordinate is the difference between the number of times the winner under h and the winner under g were ranked in the l'th place:[3]

$$\vec{\pi}_\Delta^k = \vec{\pi}_{h(\succ_k)}^k - \vec{\pi}_{g(\succ_k)}^k. \tag{6}$$

Now we can concisely write necessary conditions for $f_{\vec{\alpha}}$ agreeing with h or g, respectively, by writing:[4]

$$\vec{\pi}_\Delta^k \cdot \vec{\alpha} \geq 0 \tag{7}$$

$$\vec{\pi}_\Delta^k \cdot \vec{\alpha} \leq 0 \tag{8}$$

Notice that each vector $\vec{\pi}_\Delta^k$ has exactly m coordinates. Since we have $m+1$ such vectors (corresponding to the $m+1$ profiles in S), there must be a subset of vectors that is linearly dependent. We can therefore express one of the vectors as a linear combination of the others. W.l.o.g. we assume that the first profile's vector can be written as a combination of the others with parameters β_k, not all 0:

$$\vec{\pi}_\Delta^1 = \sum_{k=2}^{m+1} \beta_k \cdot \vec{\pi}_\Delta^k \tag{9}$$

Now, we shall construct our subset S_1 of preference profiles, on which $f_{\vec{\alpha}}$ agrees with h, as follows:

$$S_1 = \{k \in \{2, \ldots, m+1\} : \ \beta_k \geq 0\} \tag{10}$$

[3]There is some abuse of notation; if $h(\succ_k^N) = c_l$ then by $\vec{\pi}_{h(\succ_k)}^k$ we mean $\vec{\pi}_l^k$.

[4]In all profiles except $\succ_1^N$, we are indifferent to the direction in which ties are broken.

Suppose, by way of contradiction, that $f_{\vec{\alpha}}$ agrees with h on $\succ_k^N$ for $k \in S_1$, and with g on the rest. We shall examine the value of $\vec{\pi}_\Delta^1 \cdot \vec{\alpha}$:

$$\vec{\pi}_\Delta^1 \cdot \vec{\alpha} = \sum_{k=2}^{m+1} \beta_k \cdot \vec{\pi}_\Delta^k \cdot \vec{\alpha} = \sum_{k \in S_1} \beta_k \cdot \vec{\pi}_\Delta^k \cdot \vec{\alpha} + \sum_{k \notin S_1 \cup \{1\}} \beta_k \cdot \vec{\pi}_\Delta^k \cdot \vec{\alpha} \geq 0 \qquad (11)$$

The last inequality is due to the construction of S_1 — whenever β_k is negative, the sign of $\vec{\pi}_\Delta^k \cdot \vec{\alpha}$ is non-positive ($f_{\vec{\alpha}}$ agrees with g), and whenever β_k is positive, the sign of $\vec{\pi}_\Delta^k \cdot \vec{\alpha}$ is non-negative (agreement with h).

Therefore, by equation (5), we have that $f(\succ_1^N) \neq c_2 = g(\succ_1^N)$. However, it holds that $1 \notin S_1$, and we assumed that $f_{\vec{\alpha}}$ agrees with g outside S_1 — this is a contradiction. □

Theorem 3 is thus proven. The upper bound on the generalized dimension of S_m^n is quite tight: in the next subsection we show a lower bound of $m - 3$.

3.1 Lower Bound for the Generalized Dimension of $\mathcal{S}_m^n$

Theorem 2 implies that a lower bound on the generalized dimension of a function class is directly connected to the complexity of learning it. In particular, a tight bound on the dimension gives us an almost exact idea of the number of examples required to learn a scoring rule. Therefore, we wish to bound $D_G(\mathcal{S}_m^n)$ from below as well.

Theorem 5. *For all $n \geq 4$, $m \geq 4$, $D_G(\mathcal{S}_m^n) \geq m - 3$.*

Proof. We shall produce an example set of size $m - 3$ which is shattered by $\mathcal{S}_m^n$. Define a preference profile $\succ_l^N$, for $l = 3, \ldots, m-1$, as follows. For all l, the voters $1, \ldots, n-1$ rank candidate c_j in place in place j, i.e., they vote $c_1 \succ_l^i c_2 \succ_l^i \cdots \succ_l^i c_m$. The preferences $\succ_l^n$ (the preferences of voter n in profile $\succ_l^N$) are defined as follows: candidate 2 is ranked in place l, candidate 1 is ranked in place $l+1$; the other candidates are ranked arbitrarily by voter n. For example, if $m = 5$, $n = 6$, the preference profile $\succ_3^N$ is:

$\succ_3^1$	$\succ_3^2$	$\succ_3^3$	$\succ_3^4$	$\succ_3^5$	$\succ_3^6$
c_1	c_1	c_1	c_1	c_1	c_3
c_2	c_2	c_2	c_2	c_2	c_4
c_3	c_3	c_3	c_3	c_3	c_2
c_4	c_4	c_4	c_4	c_4	c_1
c_5	c_5	c_5	c_5	c_5	c_5

Lemma 6. *For any scoring rule $f_{\vec{\alpha}}$ with $\alpha_1 = \alpha_2 \geq 2\alpha_3$ it holds that:*

$$f_{\vec{\alpha}}(\succ_l^N) = \begin{cases} c_1 & \alpha_l = \alpha_{l+1} \\ c_2 & \alpha_l > \alpha_{l+1} \end{cases}$$

Proof. We shall first verify that c_2 has maximal score. c_2's score is at least $(n-1)\alpha_2 = (n-1)\alpha_1$. Let $j \geq 3$; c_j's score is at most $(n-1)\alpha_3 + \alpha_1$. Thus, the difference is at least $(n-1)(\alpha_1 - \alpha_3) - \alpha_1$. Since $\alpha_1 = \alpha_2 \geq 2\alpha_3$, this is at least $(n-1)(\alpha_1/2) - \alpha_1 > 0$, where the last inequality holds for $n \geq 4$.

Now, under preference profile $\succ_l^N$, c_1's score is $(n-1)\alpha_1 + \alpha_{l+1}$ and c_2's score is $(n-1)\alpha_1 + \alpha_l$. If $\alpha_l = \alpha_{l+1}$, the two candidates have identical scores, but c_1 was ranked first by more voters (in fact, by $n-1$ voters), and thus the winner is c_1. If $\alpha_l > \alpha_{l+1}$, then c_2's score is strictly higher — hence in this case c_2 is the winner. □

Armed with Lemma 6, we prove that the set $\{\succ_l^N\}_{l=3}^{m-1}$ is shattered by $\mathcal{S}_m^n$. Let $\vec{\alpha}^1$ such that $\alpha_1^1 = \alpha_2^1 \geq 2\alpha_3^1 = \alpha_4^1 = \cdots = \alpha_m^1$, and $\vec{\alpha}^2$ such that $\alpha_1^1 = \alpha_2^1 \geq 2\alpha_3^1 > \alpha_4^1 > \cdots > \alpha_m^1$. By the lemma, for all $l = 3, \ldots, m-1$, $f_{\vec{\alpha}^1}(\succ_l^N) = c_1$, and $f_{\vec{\alpha}^2}(\succ_l^N) = c_2$.

Let $T \subseteq \{3, 4, \ldots, m-1\}$. We must show that there exists $\vec{\alpha}$ such that $f_{\vec{\alpha}}(\succ_l^N) = c_1$ for all $l \in T$, and $f_{\vec{\alpha}}(\succ_l^N) = c_2$ for all $l \notin T$. Indeed, configure the parameters such that $\alpha_1 = \alpha_2 > 2\alpha_3$, and $\alpha_l = \alpha_{l+1}$ iff $l \in T$. The result follows directly from Lemma 6. □

4 Limitations

Heretofore, we have concentrated on trying to learn scoring rules. In particular, we have assumed that there is a scoring rule that is consistent with given training sets. We have motivated our attention to this specific family of rules by demonstrating that it is possible to obtain a variety of properties by adjusting the parameters that define scoring rules.

In this section, we push the envelope by asking the following question. Given examples that are consistent with some general voting rule, is it possible to learn a scoring rule that is "close" to the target rule? The natural definition of distance, in this case, would seem to be the fraction of preference profiles on which the two rules disagree.

Definition 4. A voting rule $f : \mathcal{L}^N \to C$ is a *c-approximation* of a voting rule g iff f and g agree on a c-fraction of the possible preference profiles:

$$|\{\succ^N \in \mathcal{L}^N : \ f(\succ^N) = g(\succ^N)\}| \geq c \cdot (m!)^n.$$

In other words, the question is: given a training set $\{(\succ_k^N, f(\succ_j^N)\}_k$, where $f : \mathcal{L}^N \to C$ is some voting rule, how hard is it to learn a scoring rule that c-approximates f, for c that is close to 1?

It turns out that the answer is: it is impossible. Indeed, there are voting rules that disagree with any scoring rule on half of all preference profiles; if the target rule f is such a rule, it is impossible to find, and of course impossible to learn, a scoring rule that is "close" to f.

Theorem 7. *Let $\epsilon > 0$. For large enough values of n and m, there is a voting rule $F : \mathcal{L}^n \to \{c_1, \ldots, c_m\}$ such that no scoring rule in $\mathcal{S}_m^n$ is a $(1/2 + \epsilon)$-approximation of F.*

In order to prove the theorem, we require the following lemma, which may be of independent interest.

Lemma 8. *There exists a polynomial $p(n, m)$ such that for all $n, m \in \mathbb{N}$, $|\mathcal{S}_m^n| \leq 2^{p(n,m)}$.*

Proof. It is true that there is an infinite number of ways to choose the vector $\vec{\alpha}$ that defines a scoring rule. Nevertheless, what we are really interested in is the number of *distinct* voting rules. For instance, if $\vec{\alpha}^1 = 2\vec{\alpha}^2$, then $f_{\vec{\alpha}^1} \equiv f_{\vec{\alpha}^2}$, i.e., the two vectors define the same voting rule.

It is clear that two scoring rules $f_{\vec{\alpha}^1}$ and $f_{\vec{\alpha}^2}$ are distinct only if the following condition holds: there exist two candidates $c_{j_1}, c_{j_2} \in C$, and a preference profile $\succ^N$, such that $f_{\vec{\alpha}^1}(\succ^N) = c_{j_1}$ and $f_{\vec{\alpha}^2}(\succ^N) = c_{j_2}$. This holds only if there exist two candidates c_{j_1} and c_{j_2} and a preference profile $\succ^N$ such that under α^1, c_{j_1}'s score is strictly greater than c_{j_2}'s, and under α_2, either c_{j_2}'s score is greater or the two candidates are tied, and the tie is broken in favor of c_{j_2}.

Now, assume $\succ^N$ induces rankings $\vec{\pi}_{j_1}$ and $\vec{\pi}_{j_2}$. The conditions above can be written as

$$\sum_l \pi_{j_1,l}\alpha_l^1 > \sum_l \pi_{j_2,l}\alpha_l^1, \tag{12}$$

$$\sum_l \pi_{j_1,l}\alpha_l^2 \leq \sum_l \pi_{j_2,l}\alpha_l^2, \tag{13}$$

where the inequality is an equality only if ties are broken in favor of c_{j_2}, i.e., if $l_0 = \min\{l : \pi_{j_1,l} \neq \pi_{j_2,l}\}$, then $\pi_{j_1,l} < \pi_{j_2,l}$.[5]

Let $\vec{\pi}_\Delta = \vec{\pi}_{j_1} - \vec{\pi}_{j_2}$. As in the proof of Lemma 4, equations (12) and (13) can be concisely rewritten as

$$\vec{\pi}_\Delta \cdot \vec{\alpha}^1 > 0 \geq \vec{\pi}_\Delta \cdot \vec{\alpha}^2, \tag{14}$$

where the inequality is an equality only if the first nonzero position in $\vec{\pi}_\Delta$ is negative.

In order to continue, we opt to reinterpret the above discussion geometrically. Each point in $\mathbb{R}^m$ corresponds to a possible choice of parameters $\vec{\alpha}$. Now, each possible choice of $\vec{\pi}_\Delta$ is the normal to a hyperplane. These hyperplanes partition the space into cells: the vectors in the interior of each cell agree on the signs of dot products with all vectors $\vec{\pi}_\Delta$. More formally, if $\vec{\alpha}_1$ and $\vec{\alpha}_2$ are two points in the interior of a cell, then for any vector $\vec{\pi}_\Delta$, $\vec{\pi}_\Delta \cdot \vec{\alpha}^1 > 0 \Leftrightarrow \vec{\pi}_\Delta \cdot \vec{\alpha}^2 > 0$. By equation (14), this implies that any two scoring rules $f_{\vec{\alpha}^1}$ and $f_{\vec{\alpha}^2}$, where $\vec{\alpha}^1$ and $\vec{\alpha}^2$ are in the interior of the same cell, are identical.

[5] W.l.o.g. we disregard the case where $\vec{\pi}_{j_1} = \vec{\pi}_{j_2}$; the reader can verify that taking this case into account multiplies the final result by an exponential factor at most.

What about points residing in the intersection of several cells? These vectors always agree with the vectors in one of the cells, as ties are broken according to rankings induced by the preference profile, i.e., according to the parameters that define our hyperplanes. Therefore, the points in the intersection can be conceptually annexed to one of the cells.

So, we have reached the conclusion that the number of distinct scoring rules is at most the number of cells. Hence, it is enough to bound the number of cells; we claim this number is exponential in n and m. Indeed, each $\vec{\pi}_\Delta$ is an m-vector, in which every coordinate is an integer in the set $\{-n, -n+1, \ldots, n-1, n\}$. It follows that there are at most $(2n+1)^m$ possible hyperplanes. It is known [7] that given k hyperplanes in d-dimensional space, the number of cells is at most $O(k^d)$. In our case, $k \leq (2n+1)^m$ and $d = m$, so we have obtained a bound of:

$$((2n+1)^m)^m \leq (3n)^{m^2} = \left(2^{\log 3n}\right)^{m^2} = 2^{m^2 \log 3n}. \tag{15}$$

□

Proof of Theorem 7. We will surround each scoring rule $f_{\vec{\alpha}} \in \mathcal{S}_m^n$ with a "ball" $B(\vec{\alpha})$, which contains all the voting rules for which $f_{\vec{\alpha}}$ is a $(1/2+\epsilon)$-approximation. We will then show that the union of all these balls does not cover the entire set of voting rules. This implies that there is a voting rule for which no scoring rule is a $(1/2+\epsilon)$-approximation.

For a given $\vec{\alpha}$, what is the size of $B(\vec{\alpha})$? As there are $(m!)^n$ possible preference profiles, the ball contains rules that do not agree with $f_{\vec{\alpha}}$ on at most $(1/2-\epsilon)(m!)^n$ preference profiles. For a profile on which there is disagreement, there are m options to set the image under the disagreeing rule.[6] Therefore,

$$|B(\vec{\alpha})| \leq \binom{(m!)^n}{(1/2-\epsilon)(m!)^n} m^{(1/2-\epsilon)(m!)^n}. \tag{16}$$

How large is this expression? Let $B'(\vec{\alpha})$ be the set of all voting rules that disagree with $f_{\vec{\alpha}}$ on *exactly* $(1/2+\epsilon)(m!)^n$ preference profiles. It holds that

$$\begin{aligned} |B'(\vec{\alpha})| &= \binom{(m!)^n}{(1/2+\epsilon)(m!)^n} (m-1)^{(1/2+\epsilon)(m!)^n} \\ &= \binom{(m!)^n}{(1/2-\epsilon)(m!)^n} ((m-1)^{1+2\epsilon})^{1/2(m!)^n} \\ &\geq \binom{(m!)^n}{(1/2-\epsilon)(m!)^n} m^{1/2(m!)^n}, \end{aligned} \tag{17}$$

where the last inequality holds for a large enough m. But since the total number of voting rules, $m^{(m!)^n}$, is greater than the number of rules in $B'(\vec{\alpha})$, we have:

$$\frac{m^{(m!)^n}}{B(\vec{\alpha})} \geq \frac{B'(\vec{\alpha})}{B(\vec{\alpha})} \geq \frac{\binom{(m!)^n}{(1/2-\epsilon)(m!)^n} m^{1/2(m!)^n}}{\binom{(m!)^n}{(1/2-\epsilon)(m!)^n} m^{(1/2-\epsilon)(m!)^n}} = m^{\epsilon(m!)^n}. \tag{18}$$

[6] This way, we also take into account voting rules that agree with $f_{\vec{\alpha}}$ on more than $(1/2+\epsilon)(m!)^n$ profiles.

Therefore

$$B(\vec{\alpha}) \leq \frac{m^{(m!)^n}}{m^{\epsilon(m!)^n}} = m^{(1-\epsilon)(m!)^n}. \tag{19}$$

If the union of balls is to cover the entire set of voting rules, we must have $|\mathcal{S}_m^n| \cdot m^{(1-\epsilon)(m!)^n} \geq m^{(m!)^n}$; equivalently, it must hold that $|\mathcal{S}_m^n| \geq m^{\epsilon(m!)^n}$. However, Lemma 8 implies that $|\mathcal{S}_m^n|$ is exponential in n and m, so for large enough values of n and m, the above condition does not hold. □

5 Conclusions

We have shown that the class of scoring rules is efficiently learnable in the PAC model. We have argued that, given properties the designer would like a voting rule to satisfy, learning from examples allows it to efficiently (albeit approximately) construct such a rule, if indeed one exists. Our basic assumption was that the designer can designate winning candidates in given preference profiles, by consulting some representation of the properties. So, the designer essentially translates a cumbersome representation of properties to a concisely represented voting rule which is easy to understand and apply.

We demonstrated that voting rules can capture a wide variety of properties. However, in Section 4 we explored the limitations of our approach, and showed that many voting rules cannot be approximated using scoring rules. This suggests that for some combinations of properties, there is no scoring rule that is close to satisfying all properties, whereas in general such a voting rule exists. On the other hand, we may have asked for too much. We did not attempt to characterize any of the disagreeing voting rules, and in practice they may be very bizarre. For example, consider the rule that sets the candidate that was most often ranked last as the winner. The abovementioned results raise two important questions, which we intend to investigate in the future:

1. Is there a class of voting rules that is significantly broader than the class of scoring rules, such that any voting rule in the former class can be approximated by a scoring rule?

2. Is there a class of voting rules that is significantly broader than the class of scoring rules, as well as efficiently learnable and concisely representable?

If the answer to one of the questions is "yes", we would be able to circumvent some of the alleged limitations of our approach.

References

[1] E. Beigman and R. Vohra. Learning from revealed preference. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 36–42, 2006.

[2] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the International Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 201–214, 2003.

[3] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence*, pages 314–319, 2002.

[4] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence*, pages 103–110, 2002.

[5] V. Conitzer and T. Sandholm. An algorithm for automatically designing deterministic mechanisms without payments. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 128–135, 2004.

[6] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 78–87, 2005.

[7] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1987.

[8] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: the anatomy of a recommender system. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 434–435, 1999.

[9] T. Haynes, S. Sen, N. Arora, and R. Nadella. An automated meeting scheduling system that utilizes user preferences. In *Proceedings of the First International Conference on Autonomous Agents*, pages 308–315, 1997.

[10] B. K. Natarajan. *Machine Learning: A Theoretical Approcach*. Morgan Kaufmann, 1991.

[11] A. D. Procaccia, J. S. Rosenschein, and G. A. Kaminka. On the robustness of preference aggregation in noisy environments. In *Proceedings of the First International Workshop on Computational Social Choice*, 2006.

[12] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, 2nd edition, 2001.

Ariel D. Procaccia, Aviv Zohar, Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Givat Ram, Jerusalem 91904, Israel
Email: {arielpro,avivz,jeff}@cs.huji.ac.il

Retrieving the Structure of Utility Graphs Used in Multi-Item Negotiation through Collaborative Filtering[1]

Valentin Robu, Han La Poutré
CWI, Dutch Center for Mathematics and Computer Science
Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands
{robu, hlp}@cwi.nl

Abstract

Graphical utility models represent powerful formalisms for modeling complex agent decisions involving multiple issues [2]. In the context of negotiation, it has been shown [8] that using utility graphs enables agents to reach Pareto-efficient agreements with a limited number of negotiation steps, even for high-dimensional negotiations over bundles of items involving complementarity/ substitutability dependencies. This paper considerably extends the results of [8], by proposing a method for constructing the utility graphs of buyers automatically, based on previous negotiation data. Our method is based on techniques inspired from item-based collaborative filtering, used in online recommendation algorithms. Experimental results show that our approach is able to retrieve the structure of utility graphs online, with a relatively high degree of accuracy, for complex, non-linear (k-additive) preference settings, even if a relatively small amount of data about concluded negotiations is available.

1 Introduction

Negotiation represents a key form of interaction between providers and consumers in electronic markets. One of the main benefi ts of negotiation in e-commerce is that it enables greater customization to individual customer preferences, and it supports buyer decisions in settings which require agreements over complex contracts. Automating the negotiation process, through the use of intelligent agents which negotiate on behalf of their owners, enables electronic merchants to go beyond price competition by providing flexible contracts, tailored to the needs of individual buyers.

Multi-issue (or multi-item) negotiation models are particularly useful for this task, since with multi-issue negotiations mutually benefi cial ("win-win") contracts can be

[1]This paper has been recently presented at the RRS'06 workshop, Hakodate, Japan [12] (proceedings to to appear as part of the Springer Lecture Notes in Computational Intelligence series). In this version of the paper, due to space limitations, the experimental set-up and tests performed to validate the model were not included. The full paper [12] (which is considerably longer, and includes the experimental results) is available at: `http://homepages.cwi.nl/~robu/rss2006.pdf`. We should also mention that the RRS'06 paper [12] represents complementary work to work on multi-issue negotiation model presented at the AAMAS'05 conference [8]. The interested reader can also consulte this paper at: `http://homepages.cwi.nl/~robu/aamas05negotiation.pdf`.

found [11, 4, 5, 8]. In this paper we consider the negotiation over the contents of a bundle of items (thus we use the term "multi-item" negotiation), though, at a conceptual level, the setting is virtually identical to previous work on multi-issue negotiation involving only binary-valued issues (e.g. [4]). A bottleneck in most existing approaches to automated negotiation is that they only deal with linearly additive utility functions, and do not consider high-dimensional negotiations and in particular, the problem of interdependencies between evaluations for different items. This is a significant problem, since identifying and exploiting substitutability/complementarity effects between different items can be crucial in reaching mutually profitable deals.

1.1 Using utility graphs to model negotiations over bundles of items

In our previous work [8], in order to to model buyer preferences in high-dimensional negotiations, we have introduced the concept of utility graphs. Intuitively defined, a utility graph (UG) is a structural model of a buyer, representing a buyer's perception of dependencies between two items (i.e. whether the buyer perceives two items to be as complementary or substitutable). An estimation of the buyer's utility graph can be used by the seller to efficiently compute the buyer's utility for a "bundle" of items, and propose a bundle and price based on this utility. The main result presented in [8] is that Pareto-efficient agreements can be reached, even for high dimensional negotiations with a limited number of negotiation steps, but provided that the seller starts the negotiation with a reasonable approximation of the *structure* of the true utility graph of the type of buyer he is negotiating with (i.e. he has a maximal structure of which issues could be potentially complimentary/substitutable in the domain).

The seller agent can then use this graph to negotiate with a specific buyer. During this negotiation, the seller will adapt the weights and potentials in the graph, based on the buyer's past bids. However, this assumes the seller knows a super-graph of the utility graphs of the class of buyers he is negotiating with (i.e. a graph which subsumes the types of dependencies likely to be encountered in a given domain - c.f. Sec. 2.2).

Due to space limitations, and to avoid too much overlap in content with our previous AAMAS paper [8], in this paper we do not describe the full negotiation model, the way seller weights are updated throughout the process, the initialization settings etc. These results have been described in [8], and we ask the interested reader to consult this work.

In this paper, we show this initial graph information can also be retrieved automatically, by using information from completed negotiation data. The implicit assumption we use here is that buyer preferences are in some way clustered, i.e. by looking at buyers that have shown interest for the same combinations of items in the past, we can make a prediction about future buying patterns of the current customer. Note that this assumption is not uncommon: it is a building block of most recommendation mechanisms deployed in Internet today [10]. In order to generate this initial structure of our utility graph, in this paper we propose a technique inspired by collaborative filtering.

1.2 Collaborative filtering

Collaborative filtering [10] is the main underlying technique used to enable personalization and buyer decision aid in today's e-commerce, and has proven very successful both in research and practice.

The main idea of collaborative filtering is to output recommendations to buyers, based on the buying patterns detected from buyers in previous buy instances. There are two approaches to this problem. The first of these is use of the preference database to discover, for each buyer, a neighborhood of other buyers who, historically, had similar preferences to the current one. This method has the disadvantage that it requires storing a lot of personalized information and is not scalable (see [10]). The second method, of more relevant to our approach, is **item-based collaborative filtering**. Item based techniques first analyze the user-item matrix (i.e. a matrix which relates the users to the items they have expressed interest in buying), in order to identify relationships between different items, and then use these to compute recommendations to the users [10]. In our case, of course, the recommendation step is completely replaced by negotiation. What negotiation can add to such techniques is that enables a much higher degree of customization, also taking into account the preferences of a specific customer. For example, a customer expressing an interest to buy a book on Amazon is sometimes offered a "special deal" discount on a set (bundle) of books, including the one he initially asked for. The potential problem with such a recommendation mechanism is that it's static: the customer can only take it, leave it or stick to his initial buy, it cannot change slightly the content of the suggested bundle or try to negotiate a better discount. By using negotiation a greater degree of flexibility is possible, because the customer can critique the merchant's sub-optimal offers through her own counter-offers, so the space of mutually profitable deals can be better explored.

1.3 Paper structure and relationship to previous work

The paper is organized as follows. In Section 2 we briefly present the general setting of our negotiation problem, define the utility graph formalism and the way it can be used in negotiations. Section 3 describes the main result of this paper, namely how the structure of utility graphs can be elicited from existing negotiation data. Section 4 discusses very briefly the experimental results from our model, fully presented in the RRS'06 paper [12]. Section 5 concludes the paper with a discussion.

An important issue to discuss is the relationship of this paper with our previous work. In our paper at the AAMAS'05 conference [8], we first introduced the utility graph formalism and present an algorithm that exploits the decomposable structure of such graphs in order to reach faster agreements during negotiation. That paper, however, uses the assumption that a minimal super-graph of individual buyer graphs is already available to the seller at the start of the negotiation. In the RRS'06 paper [12], we provide show how collaborative filtering could be used to build the structure of this super-graph and we propose a criteria for selecting the edges returned by the collaborative filtering process. This paper can be viewed as an extended abstract of these results.

For lack of space, we cannot present the full negotiation model from the AAMAS'05 paper [8] in this paper, except at a very general level. The interested reader is therefore asked to consult [8] for further details.

2 The multi-issue negotiation setting

2.1 Utility Graphs: Definition and Example

We consider the problem of a buyer who negotiates with a seller over a bundle of n items, denoted by $B = \{I_1, \ldots, I_n\}$. Each item I_i takes on either the value 0 or 1: 1 (0) means that the item is (not) purchased. The utility function $u : Dom(B) \mapsto \mathbb{R}$ specifies the monetary value a buyer assigns to the 2^n possible bundles ($Dom(B) = \{0,1\}^n$).

In traditional multi-attribute utility theory, u would be decomposable as the sum of utilities over the individual issues (items) [7]. However, in this paper we follow the previous work of [2] by relaxing this assumption; they consider the case where u is decomposable in *sub-clusters* of individual items such that u is equal to the sum of the *sub-utilities* of different clusters.

Definition: Let C be a set of (not necessarily disjoint) clusters of items $C_1, \ldots, C_r$ (with $C_i \subseteq B$). We say that a utility function is factored according to C if there exists functions $u_i : Dom(C_i) \mapsto \mathbb{R}$ ($i = 1, \ldots, r$ and $Dom(C_i) = \{0,1\}^{|C_i|}$) such that $u(\vec{b}) = \sum_i u_i(\vec{c}_i)$ where $\vec{b}$ is the assignment to the variables in B and $\vec{c}_i$ is the corresponding assignment to variables in C_i.We call the functions u_i sub-utility functions.

We use the following factorization, which is a relatively natural choice within the context of negotiation. Single-item clusters ($|C_i| = 1$) represent the individual value of purchasing an item, regardless of whether other items are present in the same bundle. Clusters with more than one element ($|C_i| > 1$) represent the *synergy effect* of buying two or more items; these synergy effects are positive for complementary items and negative for substitutable ones. In this paper, we restrict our attention to clusters of size 1 and 2 ($|C_i| \in \{1,2\}, \forall i$). This means we only consider binary item-item complementarity/substitutability relationships, though the case of retrieving larger clusters could form the object of future research.

The factorization defined above can be represented as an undirected graph $G = (V, E)$, where the vertexes V represent the set of items I under negotiation. An arc between two vertexes (items) $i, j \in V$ is present in this graph if and only if there is some cluster C_k that contains both I_i and I_j. We will henceforth call such a graph G a *utility graph*. *Example 1*Let $B = \{I_1, I_2, I_3, I_4\}$ and $C = \{\{I_1\}, \{I_2\}, \{I_1, I_2\}, \{I_2, I_3\}, \{I_2, I_4\}\}$ such that u_i is the sub-utility function associated with cluster i ($i = 1, \ldots, 5$). Then the utility of purchasing, for instance, items I_1, I_2, and I_3 (i.e., $\vec{b} = (1,1,1,0)$) can be computed as follows: $u((1,1,1,0)) = u_1(1) + u_2(1) + u_3((1,1)) + u_4((1,1))$, where we use the fact that $u_5(1,0) = u_5(0,1) = 0$ (synergy effect only occur when two or more items are purchased). The utility graph of this factorization is depicted in Fig. 1.

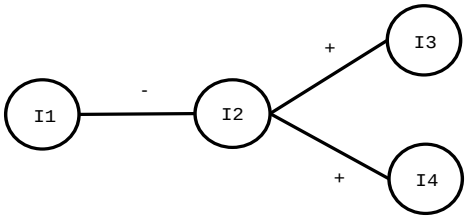


Figure 1: The utility graph that corresponds to the factorization according to C in Example 1. The $+$ and $-$ signs on the edges indicate whether the synergy represents a complementarity, respectively substitutability effect.

2.2 Minimal super-graph for a class of buyers

The definition of utility graphs given in Section 2.1 corresponds to the modeling the utility function of an individual buyer. In this paper, we call the utility graph of an individual buyer the *underlying* or *true* graph (to distinguish it from the *retrieved* or *learned* graph, reconstructed through our method). The underlying graph of any buyer remains hidden from the seller throughout the negotiation.

We do assume, however, that the buyers which negotiate with a given electronic merchant belong to a certain class or population of buyers. This means the utility buyers assign to different bundles of items follow a certain structure, specific to a buying domain (an assumption also used indirectly in [11, 10]). Buyers from the same population are expected to have largely overlapping graphs, though not all buyers will have all interdependencies specific to the class.

Definition: Let $A = \{A_1, ..A_n\}$ be a set (class, population) of n buyers. Each buyer $i = 1..n$ has a utility function u_i, which can be factored according to a set of clusters $C_i = \{C_{i,1}, C_{i,2}..C_{I,r(i)}\}$. We define the super-set of clusters for the class of buyers $A = \{A_1, ..A_n\}$ as: $C_A = C_1 \cup C_2 \cup .. \cup C_n$.

In graph-theoretic terms (as shown in Section 2.1), the set of clusters C_i according to which the utility a buyer A_i is structured is represented by a utility graph G_i, where each binary cluster from $\{C_{i,1}, ..C_{I,r(i)}\}$ represents a dependency or an edge in the graph. The super-set of buyer clusters C_A can also be represented by a graph G_A, which is the **minimal super-graph** of graphs G_i, $i = 1..n$. This graph is called **minimal** because it contains no other edges than those corresponding to a dependency in the graph of at least one buyer agent from this class. We illustrate this concept by a very simple example, which also relies on Fig. 1.

2.3 Application to negotiation

The negotiation, in our model, follows an alternating offers protocol. At each negotiation step each party (buyer/seller) makes an offer which contains an instantiation with 0/1 for all items in the negotiation set (denoting whether they are/are not included in the proposed bundle), as well as a price for that bundle. The decision process of the seller agent, at each negotiation step, is composed of 3 inter-related parts: (1) take into

account the previous offer made by the other party, by updating his estimated utility graph of the preferences of the other party, (2) compute the contents (i.e. item configuration) of the next bundle to be proposed, and (3) compute the price to be proposed for this bundle. In this model, the seller maintains of his buyer is represented by a utility graph, and tailors this graph towards the preferences of a given buyer, based on his/her previous offers.

The seller does not know, at any stage, the values in the actual utility graph of the buyer, he only has an approximation learned after a number of negotiation steps. However, the seller does have some prior information to guide his opponent modeling. He starts the negotiation by knowing a *super-graph of possible inter-dependencies* between the issues (items) which can be present for the class of buyers he may encounter. The utility graphs of buyers form subgraphs of this graph. Note that this assumption says nothing about values of the sub-utility functions, so the negotiation is still with double-sided incomplete information (i.e. neither party has full information about the preferences of the other).

2.4 Overview of our approach

There are two main stages of our approach:

1. Using information from previously concluded negotiations to construct the structure of the utility super-graph. In this phase the information used (past negotiation data) refers to a class of buyers and is not traceable to individuals.

2. The actual negotiation, in which the seller, starting from a super-graph for a class (population) of buyers, will negotiate with an individual buyer, drawn at random from the buyer population above. In this case, learning occurs based on the buyer's previous bids during the negotiation, so information is buyer-specific. However, this learning at this stage is guided by the structure of the super-graph extracted in the first phase.

3 Constructing the Structure of Utility Graphs Using Concluded Negotiation Data

Suppose the seller starts by having a dataset with information about previous concluded negotiations. This dataset may contain complete negotiation traces for different buyers, or we may choose, in order to minimize bias due to uneven-length negotiations, to consider only one record per negotiation. This can be either the first bid of the buyer or the bundle representing the outcome of the negotiation. The considered dataset is not personalized, i.e. the data which is collected online cannot be traced back to individual customers (this is a reasonable assumption in e-commerce, where storing a large amount of personalized information may harm customer privacy). However, in constructing of the minimal utility graph which the customers use, we implicitly assume that customers' preference functions are related - i.e. their corresponding utility graphs, have a (partially) overlapping structure.

Our goal is to retrieve the **minimal super-graph** of utility interdependencies which can be present for the class or population of buyers from which the negotiation data was generated. This past data can be seen as a $N * n$ matrix, where N is the number of previous negotiation instances considered and n is the number of issues (e.g. 50 for our tests). Item-based collaborative filtering [10] works by computing "similarity measures" between all pairs of items in the negotiation set. The steps used are:

1. Compute item-item similarity matrices (from the raw statistics)
2. Compute qualitative utility graph, by selecting which dependencies to consider from the similarity matrices.

In the following, we will use the following notations:

- N for the total number of previous negotiation outcomes considered
- For each item i=1..n, $N_i(1)$ and $N_i(0)$ represent the number of times the item was (respectively was not) asked by the buyer, from the total of N previous negotiations
- For each pair of issues $i, j = 1..n$ we denote by $N_{i,j}(0,0)$, $N_{i,j}(0,1)$, $N_{i,j}(1,0)$ and $N_{i,j}(1,1)$ all possibilities of joint acquisition (or non acquisition) of items i and j.

3.1 Computing the similarity matrices

The literature on item-based collaborative filtering defines two main criteria that could be used to compute the similarity between pairs of items: **cosine-based** and **correlation-based** similarity. In our work we have considered both, but experimental results showed that only correlation-based similarity seems to perform well for this task. Cosine-based similarity is conceptually simpler, and, from our experience, works well in detecting complementarity dependencies and only in the case when the data is relatively sparse (each buyer expresses interest only in a few items). Correlation-based similarity, however, does not have these limitations. Therefore, in this paper, we report the formulas and experimental results only for correlation-based similarity. Since the mathematical definitions (as presented in [10]) is given for real-valued preference ratings, we derive a more simplified form for the binary values case.

3.1.1 Correlation-based similarity

For correlation-based similarity, just one similarity matrix is computed containing both positive and negative values (to be more precise between -1 and 1). We first we define for each item $i = 1..n$, the average buy rate:

$$Av_i = \frac{N_i(1)}{N} \tag{1}$$

The following two terms are defined:

$$\psi_1 = N_{i,j}(0,0) * Av_i * Av_j - N_{i,j}(0,1) * Av_i * (1 - Av_j) \\ -N_{i,j}(1,0) * (1 - Av_i) * Av_j + N_{i,j}(1,1) * (1 - Av_i) * (1 - Av_j)$$

and the normalization factor:

$$\psi_2 = \sqrt{\frac{N_i(0) * N_i(1)}{N}} * \sqrt{\frac{N_j(0) * N_j(1)}{N}}$$

The values in the correlation-based similarity matrix are then computed as:

$$Sim(i,j) = \frac{\psi_1}{\psi_2} \tag{2}$$

3.2 Building the super-graph of buyer utilities

After constructing the similarity matrices, the next step is to use this information to build the utility super-graph for the class of buyers likely to be encountered in future negotiations. The item-item correlation similarity already provides a measure of how strong complementarity/substitutability dependencies are on average, by closeness to 1 or -1. However, we still need a method for deciding how many of the item-item relationships from the similarity matrices should be included in the final graph.

Ideally, all the inter-dependencies corresponding to the arcs in the graph representing the true underlying preferences of the buyer should feature among the highest (respectively the lowest) values in the retrieved correlation tables. When an interdependency is returned that was not actually in the true graph, we call this is an excess (extra, erroneous) arc or interdependency. Due to noise in the data, it is unavoidable that a number of such excess arcs are returned. For example, if item I_1 has a complimentary value with I_2 and I_2 is substitutable with I_3, it may be that items I_1 and I_3 often do not appear together, so the algorithm detects a substitutability relationship between them, which is in fact erroneous.

The question on the part of the seller is: how many dependencies should be considered from the ones with highest correlation, as returned by the filtering algorithm? There are two aspects that affect this cut-off decision:

- If too few dependencies are considered, then it is very likely that some dependencies (edges) that are in the true underlying graph of the buyer will be missed. This means that the seller will ignore some interdependencies in the negotiation stage completely, which can adversely affect the Pareto-efficiency of the reached agreements.

- If too many dependencies are considered, then the initial starting super-graph of the seller will be considerably more dense than the "true" underlying graph of the buyer (i.e. it contains many excess or extra edges). Actually, this is always the case to some degree, and in [8] we claim that Pareto-efficient agreements can be reached starting from a super-graph of the buyer graphs. However, this super-graph cannot be of unlimited size. For example, starting from a graph close to

full connectivity (i.e. with n^2 edges for a graph with n issues or vertexes) would be equivalent to providing no prior information to guide the negotiation process.

In the general case, we consider graphs whose number of edges (or dependencies) is a linear in the number of items (issues) in the negotiation set. Otherwise stated, we restrict our attention to graphs in which the number of edges considered is some linear factor k times the number of items (vertexes) negotiated on. Framed in this way, the problem becomes of choosing the optimal value for parameter k (henceforth denoted by k_{opt}).

3.3 Minimization of expected loss in Gains from Trade as cut-off criteria

Denote by $N_{missing}$ the number of edges that are in the "true", hidden utility graph of the buyer, but will not be present in the super-graph built through collaborative filtering. Similarly, we denote by N_{extra} the number of excess (or erroneous) edges, that will be retrieved, but are not in the true utility graph of the buyer.

The number of edges which are missing (not accurately retrieved) or excess (too many extra edges) depend on the accuracy and precision of the underlying collaborative filtering process. More precisely stated, the number of missing edges depends on 3 parameters: the type of filtering used (correlation or cosine-based), the amount of concluded negotiation records available for filtering (we denote this number by N_r) and the number of edges considered in the cut-off criteria, k. Formally, we can thus write: $N_{missing}(corr, N_r, k)$. In this section we focus, however, exclusively on choosing a value for k, and consider the other two parameters as already chosen at the earlier step. Thus we simplify the notation to: $N_{missing}(k)$ and $N_{extra}(k)$, respectively.

As discussed in Section 3.2, both having missing and too many extra edges influences the efficiency of outcome of the subsequent negotiation process. Our goal is to choose a value for k that minimizes this expected efficiency loss during the negotiation. The efficiency loss, in our case, is measured as the difference in Gains from Trade which can be achieved using a larger/smaller graph, compared to the Gains from Trade which can be achieved by using the "true" underlying utility graph of the buyer (in earlier work [11, 8], we have shown that maximizing the Gains from Trade in this setting is equivalent to reaching Pareto optimality).

In order to estimate this error rate, we consider a second **negotiation test set**, different from the one used for filtering. The purpose of this second test set is to obtain an estimation of the loss in gains from trade which occurs if we use a sparser/denser graph than the true underlying graph of the buyer. In more formal terms, the expected utility loss for using k edges can be written as:

$$E_{loss\_GT}(k) = max\langle E_{loss\_GT}(N_{missing}(k)), E_{loss\_GT}(N_{extra}(k))\rangle \quad (3)$$

The optimal choice of k can then be computed as:

$$k_{opt} = argmin_k E_{loss\_GT}(k) \quad (4)$$

Our criteria for choosing k presented in Equations 4 are not dissimilar to "min-max regret" decision criteria, often used in preference elicitation problems [1]. We could also use the name "regret" for the expected loss in gains from trade, but to keep the names consistent with our earlier AAMAS work [8] we prefer the term "GT loss".

4 Experimental evaluation

The model above was tested for a setting involving 50 binary-valued issues (items). For each set of tests, the structure of the graph was generated at random (with uniform distribution), by selecting at random the items (vertexes) connected by each edge representing a utility inter-dependency. For 50 issues, 75 random binary dependencies were generated for each test set, 50 of which were positive dependencies and 25 negative. Two sets of tests were performed: one for assessing the effi ciency of the collaborative fi ltering itself (i.e. the cosine and correlation similarity criteria) and one for detecting the cut-off limit for the maximal graph. In this paper we only report the results for correlation-based fi ltering, since this was found to perform considerably better than the cosine-based one. Next, we studied the effect of different cut-off criteria (values of k) on the negotiation process itself.

4.1 Results for the efficiency of the filtering criteria

There are two dimensions across which the two criteria need to be tested:

- **The strength of the interdependencies in the generated buyer profiles**. This is measured as a ratio of the average strength of the inter-dependency over the average utilities of an individual item. Each buyer profi le is generated as follows:

  First, for each item, an individual value is generated by drawing from identical, independent normal distributions (i.i.d.) of center $C_{individual-item} = 1$ and variance 0.5. Next, the substitutability/complementarity effects for each binary issue dependency (i.e. each cluster containing two items) are generated by drawing from a normal i.i.d-s with a centers $C_{non-linearity}$ and the same spread 0.5. The strength of the interdependency is then taken to be $\frac{C_{non-linearity}}{C_{individual-item}}$. The smaller this ratio is, the more diffi cult it will be to detect non-linearity (i.e. complementarity and substitutability effects between items). In fact, if this ratio takes the value 0, there are no effects to detect (which explains the performance at this point), at 0.1 the effects are very weak, but they become stronger as it approaches 1 and 2.

- **Number of previous negotiations from which information (i.e. negotiation trace) is available**.

The performance measure used is computed as follows. Each run of an algorithm (for a given history of negotiations, and a certain probability distribution for generating that history) returns an estimation of the utility graph of the buyer. Our performance measure is the recall, i.e. the percentage of the dependencies from the underlying

utility graph of the buyer (from which buyer profiles are generated) which are found in the graph retrieved by the seller.

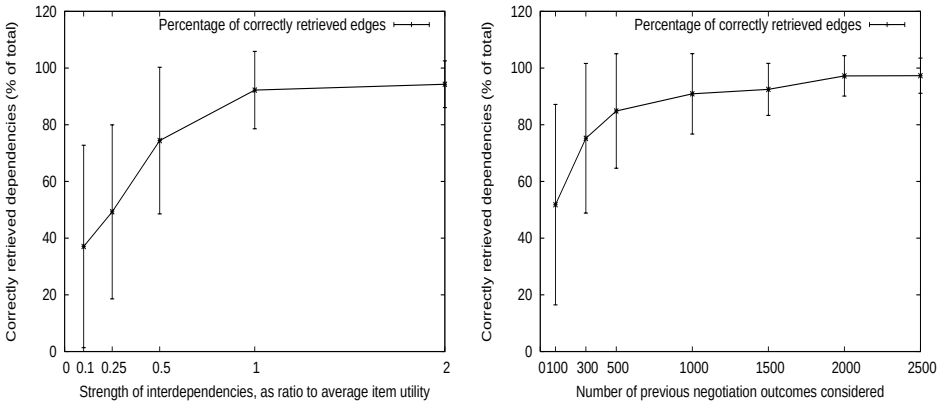


Figure 2: Results for the correlation-based similarity. Left-side graph gives the percentage of correctly retrieved dependencies, with respect to the average interdependency strength, while right-side graph gives the percentage of correctly retrieved dependencies with respect to the size of the available dataset of past negotiation traces.

4.2 Effect of the maximal graph size considered on the negotiation process

After measuring the effect of the two similarity criteria considered (i.e. cosine and correlation-based), as well as the effect of different amounts of data, we present results for different cut-off sizes for the maximal graph (i.e. the k parameter introduced in Section 3.3). For all tests reported in this Section, we used correlation-based similarity and we assumed 1000 records of previous negotiations are available for filtering. We chose to focus on correlation-based similarity since this criteria clearly performs better, in this setting, than cosine-based similarity. As shown in Sec. 4.1, 1000 records is a reasonable amount of data to ensure a good retrieval accuracy.

From Figs. 3 and 4, several conclusions can be drawn. First, missing edges from the graph the Seller starts the negotiation with has a considerably greater negative effect than adding too many extra (erroneous) edges.

Thus, as shown in Fig. 3, in order to get above 90% of the optimal Gains from Trade in future negotiations, the retrieval process cannot miss more than about 15% of the true inter-dependencies in the true graph of the Buyer. However, having a considerably denser starting graph does not degrade the performance so significantly. In fact, as we see in Fig. 4, having 3 times as many edges than in the original buyer graph (which means 2/3 of all edges are erroneous), only decreases performance with around 4%. The fact that there is still a decreasing effect can probably be explained

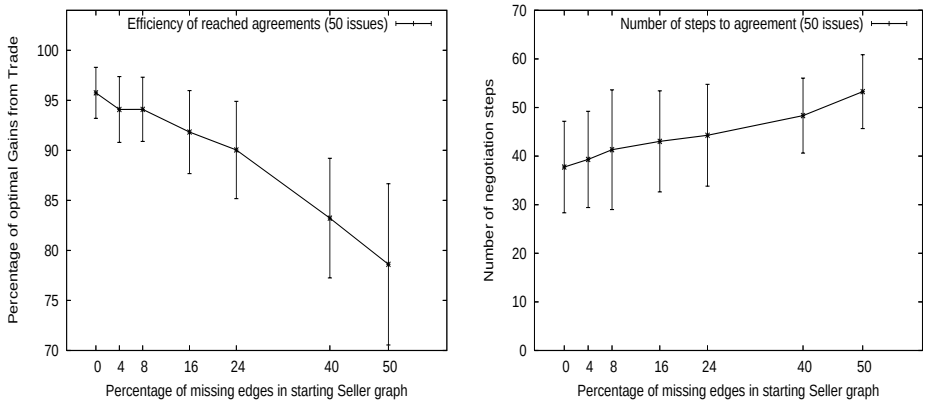


Figure 3: Effect of missing edges (dependencies) in the starting Seller graph on the Pareto-optimality of reached negotiation outcomes

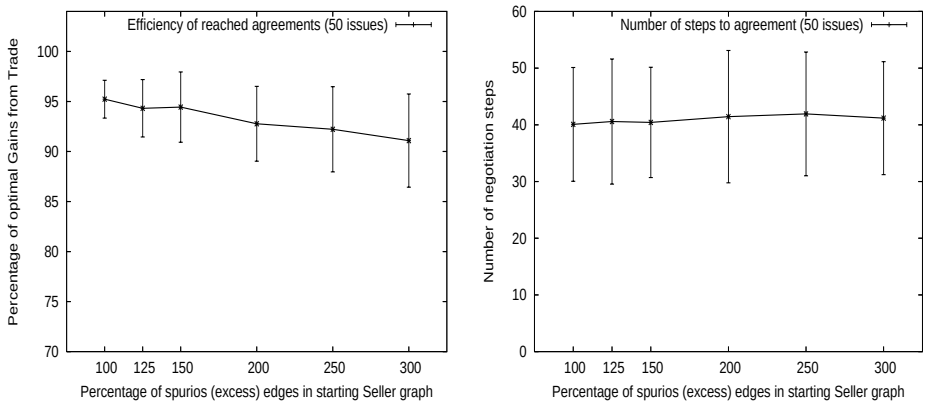


Figure 4: Effect of excess (erroneous) edges in the starting Seller graph on the Pareto-optimality of reached negotiation outcomes

from the interaction between the non-linear effects introduced by the structure and the non-linear effects introduced by the tails of normal distributions in each cluster. Finally, we observe that, in both cases, the negotiation speed does not seem to be very signifi cantly affected and it remains around 40 steps/thread, on average.

5 Discussion

Several previous results model automated negotiation as a tool for supporting the buyer's decision process in complex e-commerce domains [11, 4, 6]. Most of the work in multi-issue negotiations has focused on the independent valuations case. Faratin, Sierra & Jennings [5] introduce a method to search the utility space over multiple at-

tributes, which uses fuzzy similarity criteria between attribute value labels as prior information. These papers have the advantage that they allow flexibility in modeling and deal with incomplete preference information supplied by the negotiation partner. They do not consider the question of functional interdependencies between issues, however.

A negotiation approach that specifically address the problem of complex interdependencies between multiple issues is Klein et al. [4]. They consider a setting similar to the one considered in this paper, namely bilateral negotiations over a large number of boolean-valued issues with binary interdependencies. In this setting, they compare the performance of two search approaches: hill-climbing and simulated annealing and show that if both parties agree to use simulated annealing, then Pareto-efficient outcomes can be reached. By comparison to our work, this approach does not try to use prior information, in the form of the clustering effect between the preference functions of different buyers, in order to shorten individual negotiation threads.

Our approach to modeling multi-issue negotiation relies on constructing an explicit model of the buyer utility function - in the form of a utility graph. A difference of our approach (presented both in this paper and in [8]) from other existing negotiation approaches is that we use information from previous negotiations in order to aid buyer modeling in future negotiation instances. This does not mean that personalized negotiation information about specific customers needs to be stored, only aggregate information about all customers. The main intuition behind our model is that we explicitly utilize, during the negotiation, the clustering effect between the structure of utility functions of a population of buyers. This is an effect used by many Internet product recommendation engines today, in order to shorten the period required for customers to search for items (though it comes under different names: collaborative filtering, social filtering etc.). When adapted and used in a negotiation context, such techniques enable us to handle high dimensional and complex negotiations efficiently (with a limited number of negotiation steps).

The main contribution of this paper, in addition to the one highlighted in [8], is that it shows that the whole process can be automatic: no human input is needed in order to achieve efficient outcomes. We achieve this by using techniques derived from collaborative filtering (widely used in current e-commerce practice) to learn the structure of utility graphs used for such negotiations. We thus show that the link between collaborative filtering and negotiation is a fruitful research area, which, we argue, can lead to significant practical applications of automated negotiation systems.

As future work, there are several directions which could be explored in this area. An immediate one is to obtain a precise definition of the classes of non-linearity (in terms of utility graph structure and density) for which it is possible to reach efficient agreements with a linear number of negotiation steps. To this end, we intend to make use of results from random graph theory [9] and constraint processing [3].

References

[1] D. Brazunias and C. Boutilier. Local utility elicitation in gai models. In *Proc. of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 42–49, 2005.

[2] U. Chajewska and D. Koller. Utilities as random variables: Density estimation and structure discovery. In *Proceedings of sixteenth Annual Conference on Uncertainty in Artificial Intelligence UAI-00*, pages 63–71, 2000.

[3] R. Dechter. *Constraint Processing*. Morgan Kaufmann Publishers, San Francisco, USA, 2003.

[4] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12:111–125, 2003.

[5] N. R. Jennings P. Faratin, C. Sierra. Using similarity criteria to make issue tradeoffs in automated negotiations. *Journal of Artificial Intelligence*, 142(2):205–237, 2002.

[6] P. Maes R. Gutman. Agent-mediated integrative negotiation for retail electronic commerce. In *Agent Mediated Electronic Commerce, Springer LNAI vol. 1571*, pages 70–90, 1998.

[7] H. Raiffa. *The art and science of negotiation*. Harvard University Press, Cambridge, Massachussets USA, 1982.

[8] V. Robu, D.J.A. Somefun, and J. A. La Poutré. Modeling complex multi-issue negotiations using utility graphs. In *4th Int. Conf. on Autonomous Agents & Multi Agent Systems (AAMAS), Utrecht, The Netherlands*, 2005.

[9] A. Rucinski S. Janson, T. Luczak. *Random Graphs*. Wiley, New York, USA, 2000.

[10] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative fi ltering recommendation algorithms. In *Tenth International WWW Conference (WWW10), Hong Kong*, 2001.

[11] D.J.A. Somefun, T.B. Klos, and J.A. La Poutré. Online learning of aggregate knowledge about nonlinear preferences applied to negotiating prices and bundles. In *Proc. 6th Int Conf. on E-Commerce, Delft*, pages 361–370, 2004.

[12] J.A. La Poutré V. Robu. Retrieving the structure of utility graphs used in multi-item negotiations through collaborative fi ltering of aggregate buyer preferences. In *Proc. of the 2nd Int. Wk. on Rational, Robust and Secure Negotiations in MAS, Hakodate, Japan*. Springer LNCI (to appear), 2006.

On Determining Dodgson Winners by Frequently Self-Knowingly Correct Algorithms and in Average-Case Polynomial Time*

Jörg Rothe† and Holger Spakowski

Abstract

In their study of an efficient greedy heuristic for determining Dodgson winners, Homan and Hemaspaandra [HH06] introduced the notion of frequently self-knowingly correct algorithm. We show that this notion is closely related to Impagliazzo's notion of polynomial-time benign algorithm scheme [Imp95], which provides a model of average-case polynomial time. In particular, we show that every distributional problem (with respect to the uniform distribution) that has a polynomial-time benign algorithm scheme also has a frequently self-knowingly correct algorithm. Furthermore, we discuss Homan and Hemaspaandra's greedy heuristic with respect to AvgP, Levin's notion of average polynomial time [Lev86].

Key words: Preference aggregation, election systems, Dodgson elections, frequently self-knowingly correct algorithms, average-case complexity.

1 Introduction

This paper studies a novel type of algorithm, called frequently self-knowingly correct algorithm, and its relation to average-case polynomial time. Frequently self-knowingly correct algorithms were introduced by Homan and Hemaspaandra [HH06], who designed efficient such algorithms for solving the winner problem for Dodgson elections (which is known to be hard in the worst-case complexity model) with a guaranteed high frequency of success.

Before we turn to the main purpose of this paper, let us give a brief overview of recent results on complexity-theoretic issues related to voting in order to motivate this paper's topic in a broader framework. For more background on computational politics and the complexity of electoral problems, we refer to the comprehensive surveys [FHHR06, HH00].

Preference aggregation and election systems have been studied for centuries in social choice theory, political science, and economics, see, e.g., Arrow [Arr63],

*Supported in part by the German Science Foundation under grants RO 1202/9-1 and RO 1202/9-3.

†Supported in part by the Alexander von Humboldt Foundation in the TransCoop program.

Black [Bla58], Brams and Fishburn [BF83], and McLean and Urken [MU95]. Recently, these topics have become the focus of attention in various areas of computer science as well, such as artificial intelligence (especially with regard to distributed AI in multi-agent settings), computational complexity, and operations research. In the field of computational complexity, much work has been done during the past few years on the following four important classes of problems for various election systems.

Let $\mathcal{E}$ be a given election system. The *winner problem* for $\mathcal{E}$ asks whether a designated candidate has won a given election under $\mathcal{E}$. Bartholdi, Tovey, and Trick [BTT89b] proved that the winner problem for both Dodgson elections [Dod76] and Kemeny elections [Kem59, KS60] is NP-hard. Hemaspaandra, Hemaspaandra, and Rothe [HHR97] optimally improved the former result by proving the Dodgson winner problem complete (under polynomial-time many-one reductions) for $\mathrm{P}_{\|}^{\mathrm{NP}}$, the class of problems solvable via parallel access to NP. This class is known to be equal to a number of other classes (including $\mathrm{P}_{\text{truth-table}}^{\mathrm{NP}}$ and $\mathrm{P}^{\mathrm{NP[log]}}$, see [PZ83, Wag90]) and forms the Θ_2^p level of the polynomial hierarchy.

Rothe, Spakowski, and Vogel [RSV03] proved that the winner problem for Young elections [You77] is also $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete, and Hemaspaandra, Spakowski, and Vogel [HSV05] obtained the analogous result for the Kemeny winner problem. Each of these three election systems is based on a certain combinatorial optimization problem, and while each of these systems avoids the Condorcet Paradox, it does respect the Condorcet Principle [Con85, Fis77], i.e., it selects the Condorcet winner whenever one exists.[1] All the above results focus on "winner problems" (which ask if the designated candidate wins regardless of whether there are other winners as well) as opposed to "unique winner problems," and we also adopt the traditional model of winner problem here.[2]

The *control problem* for election system $\mathcal{E}$ asks whether it is feasible for an election chair to alter the outcome of an election by changing its agenda (see, e.g., [BTT92, HHR05]). The *manipulation problem* (a.k.a. the *strategic voting problem*) for $\mathcal{E}$ asks whether it is feasible that the outcome of an election can be altered by having voters strategically change their preferences (see, e.g., [BTT89a, BO91, CS02, CLS03, HH05]). The *bribery problem* for $\mathcal{E}$ (which is somewhat related to manipulation) asks whether it is feasible that the outcome of an election can be altered by an agent who bribes voters to change their votes. Of course, most desirable are voting systems whose winner problem is easy yet which resist electoral control, manipulation, and bribery.

We are concerned with the winner problem only. Since for some election systems with otherwise useful properties the winner problem is hard (in the worst-case complexity model), it is natural to wonder if one at least can find

[1] A Condorcet winner is a candidate who beats all other candidates in pairwise majority-rule contests.

[2] Note, however, that Hemaspaandra, Hemaspaandra, and Rothe [HHR06] have recently shown that also the unique winner problems for Dodgson, Young, and Kemeny elections are $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete.

a heuristic algorithm solving the problem for "most of the inputs occurring in practice." Examples of such heuristics are known, e.g., for the Dodgson winner problem [HH06] and for the web page ranking problem in close relation to the Kemeny winner problem [DKNS01]. In particular, we study a heuristic called `GreedyWinner` for the problem of determining the winners of Dodgson elections. This heuristic is due to Homan and Hemaspaandra [HH06], who proved that if the number of voters greatly exceeds the number of candidates (which in many real-world cases is a very plausible assumption), then their heuristic is a "frequently self-knowingly correct algorithm," a notion they introduced in [HH06]. We show that this notion is closely related to average-case complexity.

This paper is organized as follows. In Section 2, we give a brief introduction to social choice theory (and, in particular, to Dodgson elections), we present some foundations of average-case complexity theory, and we define the notion of frequently self-knowingly correct algorithm. Section 3 provides our main result: Every problem in AvgP has a frequently self-knowingly correct algorithm. In Section 4, we discuss the relation of Homan and Hemaspaandra's greedy heuristic for finding Dodgson winners to average-case polynomial time. Finally, in Section 5, we conclude by raising some related open questions.

2 Preliminaries

2.1 Some Background from Social Choice Theory

An election $E = (C, V)$ is given by a set C of candidates and a set V of votes, where each vote is specified by a preference order on all candidates. As is common, we assume that the underlying preference relation is strict (i.e., irreflexive and antisymmetric), transitive, and complete.

In this paper, we focus on Dodgson elections. In 1876, Dodgson proposed an election system [Dod76] that is based on a combinatorial optimization problem: An election is won by those candidates who are "closest" to being a *Condorcet winner*, the unique candidate (if one exists) who defeats each other candidate in pairwise comparison by a strict majority of votes.

More precisely, given a Dodgson election $E = (C, V)$, every candidate c in C is assigned a score, which is denoted by DodgsonScore(C, V, c) and is defined to be the smallest number of sequential exchanges of adjacent preferences in the voters' preference orders needed to make c a Condorcet winner with respect to the resulting preference orders. Whoever has the lowest Dodgson score wins.

The problem `DodgsonWinner` is defined as follows: Given an election $E = (C, V)$ and a designated candidate c in C, is c a Dodgson winner in E? (The search version of this decision problem can easily be derived.) As mentioned above, Hemaspaandra, Hemaspaandra, and Rothe [HHR97] have shown that this problem is $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete.

It certainly is not desirable to have an election system whose winner problem is hard, as only systems that can be evaluated efficiently are actually used in practice. Fortunately, there are a number of positive results on Dodgson

elections and related systems as well. In particular, Bartholdi, Tovey, and Trick [BTT89b] proved that for elections with a bounded number of candidates or voters Dodgson winners are easy to determine. Fishburn [Fis77] proposed a "homogeneous" variant of Dodgson elections that Rothe, Spakowski, and Vogel [RSV03] proved to have a polynomial-time winner problem. McCabe-Dansted, Pritchard, and Slinko [MDPS06] proposed a scheme (called Dodgson Quick) that approximates Dodgson's rule with an exponentially fast convergence. Finally, Homan and Hemaspaandra [HH06] proposed a greedy heuristic that finds Dodgson winners with a guaranteed high frequency of success. In particular, they introduced the notion of "frequently self-knowingly correct algorithm," which we define in Section 2.3 below, and they noted:

> "The closest related concepts [...] are probably those involving proofs to be verified, such as NP certificates and the proofs in interactive proof systems."

This statement notwithstanding, we will show that in fact it is the theory of average-case complexity (which Homan and Hemaspaandra also mention in their paper) that is even more closely related to their notion.

2.2 Foundations of Average-Case Complexity Theory

The theory of average-case complexity was initiated by Levin [Lev86]. A problem's average-case complexity can be viewed a more significant measure than its worst-case complexity in many cases, for example in cryptographic applications. We here follow Goldreich's presentation [Gol97]. Another excellent introduction to this theory is due to Wang [Wan97].

Fix the alphabet $\Sigma = \{0, 1\}$, let Σ^* denote the set of strings over Σ, and let Σ^n denote the set of all length n strings in Σ^*. For any $x, y \in \Sigma^*$, $x < y$ means that x precedes y in lexicographic order, and $x - 1$ denotes the lexicographic predecessor of x.

Intuitively, Levin observed that many hard problems—including those that are NP-hard in the traditional worst-case complexity model—may nonetheless be easy to solve "on the average," i.e., for "most" inputs or for "most practically relevant" inputs. He proposed to define the complexity of problems with respect to some suitable distribution on the input strings.

We now define the notion of a distributional problem and the complexity class AvgP. In subsequent sections, we consider two heuristic algorithms: the algorithm `GreedyWinner` intended to solve the decision problem `DodgsonWinner`, and the algorithm `GreedyScore` intended to compute the Dodgson score of some given candidate. Both heuristics work well sufficiently often, provided that the number of voters greatly exceeds the number of candidates.

Here, we define only distributional search problems; the definition of distributional decision problems is analogous.

Definition 1 *1. A* distribution function $\mu : \Sigma^* \to [0,1]$ *is a nondecreasing function from strings to the unit interval that converges to one (i.e.,* $\mu(0) \geq 0$ *and* $\mu(x) \leq \mu(y)$ *for each* $x < y$*, and* $\lim_{x\to\infty} \mu(x) = 1$*). The* density function associated with μ *is defined by* $\mu'(0) = \mu(0)$ *and* $\mu'(x) = \mu(x) - \mu(x-1)$ *for each* $x > 0$*. That is, each string* x *gets weight* $\mu'(x)$ *with this distribution.*

2. A distributional (search) problem *is a pair* (f, μ)*, where* $f : \Sigma^* \to \Sigma^*$ *is a function and* $\mu : \Sigma^* \to [0,1]$ *is a distribution function.*

3. A function $t : \Sigma^* \to \mathbb{N}$ *is* polynomial on the average with respect to some distribution μ *if there exists a constant* $\epsilon > 0$ *such that*

$$\sum_{x\in\Sigma^*} \mu'(x) \cdot \frac{t(x)^\epsilon}{|x|} < \infty.$$

4. The class AvgP *consists of all distributional problems* (f, μ) *for which there exists an algorithm* $\mathcal{A}$ *computing* f *such that the running time of* $\mathcal{A}$ *is polynomial on the average with respect to the distribution* μ*.*

In this paper, we focus on the standard uniform distribution μ on Σ^*, which is defined by

$$\mu'(x) = \frac{1}{|x|(|x|+1)2^{|x|}}.$$

That is, we first choose an input size n at random with probability $1/n(n+1)$, and then we choose an input string of that size n uniformly at random. For each $n \in \mathbb{N}$, let μ_n be the restriction of μ to strings of length at most n.

Impagliazzo [Imp95] introduced the notion of polynomial-time benign algorithm scheme to present an alternative view on the definition of average polynomial time.

Definition 2 ([Imp95]) *1. An algorithm computes a function* f *with* benign faults *if it either ouputs an element of the image of* f *or "?," and if it outputs anything other than ?, it is correct.*

2. A polynomial-time benign algorithm scheme *for a function* f *on* μ_n *is an algorithm* $\mathcal{A}(x, \delta)$ *such that:*

(a) $\mathcal{A}$ *runs in time polynomial in* $|x|$ *and* $1/\delta$*.*

(b) $\mathcal{A}$ *computes* f *with benign faults.*

(c) For each δ*,* $0 < \delta < 1$*, and for each* $n \in \mathbb{N}^+$*,*

$$\mathrm{Prob}_{\mu'_n}[\mathcal{A}(x,\delta) = \ ?] \leq \delta.$$

2.3 A Frequently Self-Knowingly Correct Greedy Algorithm

Homan and Hemaspaandra [HH06] define the following notion.

Definition 3 ([HH06]) *1. Let $f : S \to T$ be a function, where S and T are sets. We say an algorithm $\mathcal{A} : S \to T \times \{$"definitely", "maybe"$\}$ is* self-knowingly correct for f *if, for each $s \in S$ and $t \in T$, whenever $\mathcal{A}$ on input s outputs $(t,$ "definitely"$)$ then $f(s) = t$.*

2. An algorithm $\mathcal{A}$ that is self-knowingly correct for $g : \Sigma^* \to T$ is said to be frequently self-knowingly correct for g *if*

$$\lim_{n\to\infty} \frac{||\{x \in \Sigma^n \mid A(x) \in T \times \{\text{"maybe"}\}\}||}{||\Sigma^n||} = 0.$$

In their seminal paper [HH06], Homan and Hemaspaandra present two frequently self-knowingly correct polynomial-time algorithms, which they call `GreedyScore` and `GreedyWinner`. Since `GreedyWinner` can easily be reduced to `GreedyScore`, we focus on `GreedyScore` only and briefly describe the intuition behind this algorithm; for full detail, we refer to [HH06].

If $E = (C, V)$ is an election and c is some designated candidate in C, we call (C, V, c) a *Dodgson triple*. Given a Dodgson triple (C, V, c), `GreedyScore` determines the Dodgson score of c with respect to the given election (C, V). We will see that there are Dodgson triples (C, V, c) for which this problem is particularly easy to solve.

For any $d \in C - \{c\}$, let Deficit$[d]$ be the number of votes c needs to gain in order to have more votes than d in a pairwise contest between c and d.

Definition 4 *Any Dodgson triple (C, V, c) is said to be* nice *if for each candidate $d \in C - \{c\}$, there are at least* Deficit$[d]$ *votes for which candidate c is exactly one position below candidate d.*

Given a Dodgson triple (C, V, c), the algorithm `GreedyScore` works as follows:

1. For each candidate $d \in C - \{c\}$, determine Deficit$[d]$.
2. If (C, V, c) is not nice then output ("anything","maybe"); otherwise, output
$$(\textstyle\sum_{d \in C-\{c\}} \text{Deficit}[d], \text{"definitely"}).$$

Note that, for nice Dodgson triples, we have

$$\text{DodgsonScore}(C, V, c) = \sum_{d \in C-\{c\}} \text{Deficit}[d],$$

It is easy to see that `GreedyScore` is a self-knowingly correct polynomial-time bounded algorithm. To show that it is even *frequently* self-knowingly correct, Homan and Hemaspaandra prove the following key lemma. Their proof uses a variant of Chernoff bounds.

Lemma 5 (see Thm. 4.1(3) in [HH06]) *Let (C, V, c) be a given Dodgson triple with $n = ||V||$ votes and $m = ||C||$ candidates, chosen uniformly at random among all such Dodgson elections. The probability that (C, V, c) is not nice is at most*

$$2(m-1)e^{-\frac{n}{8m^2}}.$$

Homan and Hemaspaandra [HH06] show that the heuristic `GreedyWinner`, which is based on `GreedyScore` and which solves the winner problem for Dodgson elections, also is a frequently self-knowingly correct polynomial-time algorithm. This result is stated formally below.

Theorem 6 ([HH06]) *For all $m, n \in \mathbb{N}^+$, the probability that a Dodgson election E selected uniformly at random from all Dodgson elections having m candidates and n votes (i.e., all $(m!)^n$ Dodgson elections having m candidates and n votes have the same likelihood of being selected) has the property that there exists at least one candidate c such that* `GreedyWinner` *on input (E, c) outputs "maybe" as its second output component is less than $2(m^2 - m)e^{-\frac{n}{8m^2}}$.*

3 On AvgP and Frequently Self-Knowingly Correct Algorithms

Our main result relates polynomial-time benign algorithm schemes to frequently self-knowingly correct algorithms. We show that every distributional problem (with respect to the uniform distribution) that has a polynomial-time benign algorithm scheme must also have a frequently self-knowingly correct algorithm. It follows that all AvgP problems have a frequently self-knowingly correct algorithm.

Theorem 7 *Suppose that $\mathcal{A}(x, \delta)$ is a polynomial-time benign algorithm scheme for a distributional problem f on μ_n. Then there is a frequently self-knowingly correct algorithm $\mathcal{A}'$ for f.*

Proof For each $n \in \mathbb{N}$, let $\delta(n) = 1/n^3$. Define the algorithm $\mathcal{A}'$ as follows:

1. On input $x \in \Sigma^*$, simulate $\mathcal{A}(x, \delta(|x|))$.
2. If $\mathcal{A}(x, \delta(|x|))$ outputs ?, then output (*anything*, *"maybe"*).
3. If $\mathcal{A}(x, \delta(|x|))$ outputs $y \in T$, where $y \neq ?$, then output (y, *"definitely"*).

By definition of "polynomial-time benign algorithm scheme," algorithm $\mathcal{A}'$ is polynomial-time bounded. It remains to show that $\mathcal{A}'$ is frequently self-knowingly correct.

Fix an arbitrary $n \in \mathbb{N}$. Then

$$\begin{aligned}
&\frac{||\{x \in \Sigma^n \mid \mathcal{A}'(x) \in T \times \{\text{“maybe”}\}\}||}{||\Sigma^n||} \\
&= \quad \text{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ? \mid |x| = n] \\
&= \quad \frac{\text{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ? \text{ and } |x| = n]}{\text{Prob}_{\mu'_n}[|x| = n]} \\
&\leq \quad \frac{\text{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ?]}{\text{Prob}_{\mu'_n}[|x| = n]}
\end{aligned}$$

$$\begin{aligned}
&\leq \quad \frac{\text{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ?]}{1/n(n+1)} && (1) \\
&= \quad n(n+1) \cdot \text{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ?] \\
&\leq \quad n(n+1) \cdot \delta(|x|) && (2) \\
&= \quad \frac{n(n+1)}{n^3} \\
&\leq \quad \frac{n+1}{n^2}.
\end{aligned}$$

Here, (1) holds because, by definition of μ,

$$\text{Prob}_{\mu'_n}[|x| = n] \geq 1/n(n+1),$$

and (2) is true by the definition of polynomial-time benign algorithm scheme.

We have shown that

$$\lim_{n \to \infty} \frac{||\{x \in \Sigma^n \mid \mathcal{A}'(x) \in T \times \{\text{“maybe”}\}\}||}{||\Sigma^n||} = 0,$$

which completes the proof. ∎

Corollary 8 *Every distributional problem (under the standard uniform distribution) that is in* AvgP *has a frequently self-knowingly correct algorithm.*

Proof Impagliazzo proved that any distributional problem on input ensemble μ_n is in AvgP if and only if it has a polynomial-time benign algorithm scheme; see Proposition 2 in [Imp95]. The claim now follows from Theorem 7. ∎

4 Dodgson Winners and Average-Case Polynomial Time

Because of the close relationship between the notion of frequently self-knowingly correctness and average-case complexity, one might think that Homan and Hemaspaandra's algorithm could be used to devise an algorithm, call it

DodgsonWinner-AverageGood, witnessing that the problem DodgsonWinner is in AvgP (assuming the uniform probability distribution on all elections with the same number of candidates and voters). A tempting approach towards this goal would be as follows. Given an election, run the GreedyWinner algorithm on it. If that fails (i.e., if GreedyWinner outputs "maybe"), then use the exhaustive algorithm that works in time $O(m^n)$.

We now show that this approach does not work in any obvious way. The reason is that the algorithm is not "frequently enough" self-knowingly correct. That is, DodgsonWinner-AverageGood would have to spend too much time (namely, $O(m^n)$) on a too large fraction of the inputs, namely,

$$2(m^2 - m)e^{-\frac{n}{8m^2}}.$$

Let $\mu'(m,n)$ be the probability that an election chosen randomly under the uniform distribution has m candidates and n voters. We assume that the elections are encoded into strings in a reasonable way. Use, for instance, the encoding scheme given by Homan and Hemaspaandra [HH06].

To prove that DodgsonWinner is in AvgP, we would have to show that there is an $\epsilon > 0$ such that

$$\sum_{x \in \Sigma^*} \mu'(x) \frac{t(x)^\epsilon}{|x|} < \infty,$$

where $t(x)$ is the computation time of the algorithm DodgsonWinner-AverageGood on input x, see Definition 1.

Let $E(m,n)$ be the set of strings in Σ^* that encode elections having m candidates and n voters.

For any $x \in E(m,n)$,

$$\mu'(x) = \frac{1}{||E(m,n)||} \cdot \mu'(m,n).$$

By Theorem 6, for fixed m and n, we obtain

$$\sum_{x \in E(m,n)} \mu'(x) \frac{t(x)^\epsilon}{|x|} \leq \mu'(m,n) \cdot \left(\frac{p(|x|)^\epsilon}{|x|} + \alpha(m,n) \cdot (cm^n)^\epsilon \right),$$

where $\alpha(m,n) = 2(m^2 - m)e^{-\frac{n}{8m^2}}$. Here, p is the polynomial that bounds the computation time of GreedyWinner. We can clearly choose $\epsilon > 0$ small enough such that $p(|x|)^\epsilon/|x|$ contributes only a constant in the above term. However, the crucial fact is that the term

$$\alpha(m,n) \cdot (cm^n)^\epsilon = 2(m^2 - m)e^{-\frac{n}{8m^2}} \cdot (cm^n)^\epsilon$$

grows exponentially, no matter how small we choose ϵ.

Thus, DodgsonWinner-AverageGood does not run in average-case polynomial time with respect to any interesting distribution on the inputs.

5 Conclusions

Homan and Hemaspaandra [HH06] proposed a greedy heuristic for finding Dodgson winners, and they proved that this heuristic is a frequently self-knowingly correct algorithm. We have shown that every distributional problem (with respect to the standard uniform distribution) in AvgP has a frequently self-knowingly correct algorithm. It is easy to see that the converse implication is not true. (For instance, one can define a problem L that consists only of strings in $\{0\}^*$ encoding the halting problem. This problem is clearly not in AvgP, yet it is frequently self-knowingly correct.)

Furthermore, we have argued why it might be hard to show that the problem of finding Dodgson winners—at least via Homan and Hemaspaandra's heuristic—can be solved in polynomial time on the average. We suspect that this problem is hard on the average, but a rigorous proof for this claim has eluded us so far, and we raise this as an open question. In particular, it would be interesting to study the average-case complexity of the Dodgson winner problem with respect to suitable distributions (see, e.g., Procaccia and Rosenschein [PR06]).

Other interesting issues remain open as well. For example, one might study the approximability of the Dodgson winner problem; some first steps in this direction have been taken by McCabe-Dansted, Pritchard, and Slinko [MDPS06]. Also, one could investigate other voting systems with a hard winner problem in the worst-case model, such as the problem of determining Young winners [RSV03] or the problem of determining Kemeny winners [HSV05], and seek to find algorithms that can be shown to be frequently self-knowingly correct or even average-case polynomial-time, or else seek to prove these problems hard on the average.

Acknowledgments: We thank Lane A. Hemaspaandra and Chris Homan for their interest in this work, and for many inspiring discussions on computational issues related to voting. We also thank the COMSOC '06 referees for their helpful comments.

References

[Arr63] K. Arrow. *Social Choice and Individual Values.* John Wiley and Sons, 1951 (revised edition 1963).

[BF83] S. Brams and P. Fishburn. *Approval Voting.* Birkhäuser, Boston, 1983.

[Bla58] D. Black. *The Theory of Committees and Elections.* Cambridge University Press, 1958.

[BO91] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[BTT89a] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT89b] 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(2):157–165, 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.

[CLS03] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 201–214. ACM Press, 2003.

[Con85] 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. English translation appears in I. McLean and A. Urken, *Classics of Social Choice*, University of Michigan Press, 1995, pages 91–112.

[CS02] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 314–319. AAAI Press, 2002.

[DKNS01] 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.

[Dod76] C. Dodgson. A method of taking votes on more than two issues. Pamphlet printed by the Clarendon Press, Oxford, and headed "not yet published" (see the discussions in [MU95, Bla58], both of which reprint this paper), 1876.

[FHHR06] 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, 2006. To appear.

[Fis77] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[Gol97] O. Goldreich. Note on Levin's theory of average-case complexity. Technical Report TR97-058, Electronic Colloquium on Computational Complexity, November 1997.

[HH00] E. Hemaspaandra and L. Hemaspaandra. Computational politics: Electoral systems. In *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science*, pages 64–83. Springer-Verlag *Lecture Notes in Computer Science #1893*, 2000.

[HH05] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. Technical Report TR-861, University of Rochester, Department of Computer Science, Rochester, NY, April 2005.

[HH06] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science*, pages 528–539. Springer-Verlag *Lecture Notes in Computer Science #4162*, August/September 2006.

[HHR97] 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, November 1997.

[HHR05] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 95–101. AAAI Press, 2005.

[HHR06] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report TR-900, Department of Computer Science, University of Rochester, Rochester, NY, June 2006. Revised, August 2006.

[HSV05] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, December 2005.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society Press, 1995.

[Kem59] J. Kemeny. Mathematics without numbers. *Dædalus*, 88:571–591, 1959.

[KS60] J. Kemeny and L. Snell. *Mathematical Models in the Social Sciences.* Ginn, 1960.

[Lev86] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[MDPS06] J. McCabe-Dansted, G. Pritchard, and A. Slinko. Approximability of Dodgson's rule. Technical Report TR-551, Auckland University, Department of Mathematics, Auckland, New Zealand, June 2006.

[MU95] I. McLean and A. Urken. *Classics of Social Choice*. University of Michigan Press, Ann Arbor, Michigan, 1995.

[PR06] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 497–504. ACM Press, May 2006.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings of the 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.

[RSV03] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, June 2003.

[Wag90] K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[Wan97] J. Wang. Average-case computational complexity theory. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 295–328. Springer-Verlag, 1997.

[You77] H. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16(2):335–353, 1977.

Jörg Rothe
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: `rothe@cs.uni-duesseldorf.de`

Holger Spakowski
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: `spakowsk@cs.uni-duesseldorf.de`

Voting cycles in a computational electoral competition model with endogenous interest groups

Vjollca Sadiraj*, Jan Tuinstra† and Frans van Winden‡

Abstract

We develop a computational electoral model by extending the benchmark model of spatial competition in two directions. First, political parties do not have complete information about voter preferences but behave adaptively and use polls to find policy platforms that maximize the probability of winning an election. Second, we allow voters to organize in different interest groups endogenously and depending upon the incumbent's policy platform. These interest groups transmit information about voter preferences to political parties and coordinate voting behavior. We use computational methods to investigate the convergence properties of this model. We find that the introduction of endogenous interest groups increases the separation between parties platforms, inhibits convergence to the center of the distribution of voter preferences, and increases the size of the winning set. Moreover, the presence of interest groups in an environment with adaptively searching political parties increases the likelihood of voting cycles, even when a dominant point exists. We also investigate the dynamics of this agent-based spatial model of electoral competition by looking at the mean-dynamics, i.e. by replacing stochastic variables by their expected values. The resulting Markov process shows that voting cycles exist. The mechanism driving these voting cycles may explain some empirical regularities found in the political science literature.

Keywords: Computational political economy, interest groups, spatial competition, polling, campaign contributions.

JEL classification code: D72; D83.

*Department of Economics, Andrew Young School of Policy Studies, Georgia State University, vsadiraj@gsu.edu.

†Department of Quantitative Economics and CeNDEF, University of Amsterdam, the Netherlands, J.Tuinstra@uva.nl. This research has been supported by the Netherlands Organisation for Scientific Research (NWO) under a MaGW-Pionier grant.

‡Department of Economics and CREED, University of Amsterdam, the Netherlands, F.A.A.M.vanWinden@uva.nl.

1 Introduction

Existing models of electoral competition typically make strong assumptions about the information political parties and voters use. In Downs-Hotelling spatial competition models, for example, the preferred policy of a voter is modeled as a point in an issue space and voters vote for the party whose policy platform is 'closest' to this ideal point. Each voter is assumed to be able to evaluate the consequences of all policy positions and each party is assumed to have complete information about the distribution of the voters' ideal points. These assumptions are not very realistic and in this paper we will take the informational constraints in politics explicitly into account, using a spatial competition model with two office motivated parties. Starting point is the observation that parties have to find out about voter preferences through some kind of polling. However, this search activity is costly. Voters may be willing to contribute in the form of effort or money because this allows them to affect the election outcome as well as policy platforms. Conditioning takes place by making contributions only available for polling in that part of the political issue space that the voter is mostly concerned about. For simplicity, we will have voters contribute to an 'interest group' which conditionally transfers the contributions to the parties. In line with recent literature contributions are assumed to be primarily driven by dissatisfaction with existing policies on issues of particular concern to the voter. Note that by getting politically involved in this way voters are likely to identify themselves with the policy stances they go for. In our model it is assumed, therefore, that some coordination of voting will occur. This coordination of voting may affect policies.

Our study is related to Kollman *et al.* (1992), which investigates the relevance of the theoretical "chaos" results for multi-dimensional issue spaces, which predict that, in general, the challenging party can always defeat the incumbent. They found convergence of the parties' platforms to the center of the distribution of voters' ideal positions. Sadiraj *et al.* (2006) presents extensive simulation studies of a spatial competition model with endogenous emergence of interest groups and shows that their presence increases separation between policy platforms and increases the probability of winning for the challenger. In this paper we provide a theoretical underpinning of these results by considering the *mean dynamics*, where we replace the stochastic elements of the model by their expected values and study the asymptotic properties of the resulting Markov model. It turns out that the steady state distribution of policy outcomes depends critically upon the way interest groups transmit information about the electoral landscape to the political parties. The model with interest groups may help explain some "stylized facts" concerning empirical data on policy outcomes.

The rest of the paper is organized as follows. Section 2 introduces the computational electoral competition model and the mean dynamics are introduced and studied in Section 3. Section 4 presents a general result on voting cycles and Section 5 concerns a replication of some stylized facts. Section 6 concludes.

2 The spatial competition model[1]

2.1 Incompletely informed political parties

Policy platforms are represented as points in a discrete two-dimensional issue spaces $\mathcal{X} = \{1, \ldots, K\} \times \{1, \ldots, K\}$. There is a population of N voters, where utility voter j attaches to policy outcome $y = (y_1, y_2)$ is given by

$$u_j(y) = -\sum_{i=1}^{2} s_{ji} (x_{ji} - y_i)^2, \tag{1}$$

with $x_j \in \mathcal{X}$ the voter's *ideal point* and s_{ji} the *strength* voter j attaches to issue i, where $s_j = (s_{j1}, s_{j2}) \in \mathcal{S} \times \mathcal{S}$ with $\mathcal{S} = \{s_0, s_1, \ldots, s_c\}$ and $0 \leq s_0 < \ldots < s_c \leq 1$. A configuration of voters is generated by drawing, for each j, an ideal position x_j from the discrete uniform distribution on $\mathcal{X}$ and strengths s_{ji} from a discrete distribution on $\mathcal{S}$. There are two political parties entering the election, the incumbent and the challenger. The incumbent does not change its policy position y from the previous period. Each voter votes for the political candidate yielding him the highest utility as given by (1). Then for each position z the height of the *electoral landscape*, $h(z \mid y)$, is given by the fraction of voters voting for the challenger, if it would select that position. For every z with $h(z \mid y) > (<) \frac{1}{2}$, the challenger wins (loses) the election. (If $h(z \mid y) = \frac{1}{2}$, the challenger wins with probability $\frac{1}{2}$). The objective for the challenger is to find maxima of the electoral landscape. Instead of assuming that political parties or candidates have complete information about the electoral landscape, we follow Kollman, Miller and Page (1992) in assuming that political parties have incomplete information about voter preferences and select policy platforms *adaptively* as follows. The challenger randomly draws a number of positions from the issue space and runs a poll there. Such a poll consists of, for example, a randomly drawn 10% of the voters. The challenger observes the fraction of this poll which favors his policy over the incumbents policy and uses this as an estimate of the true height of the electoral landscape at that position. If the best polling result indicates a height of at least $\frac{1}{2}$ then the challenger chooses that position. Otherwise it chooses the incumbent position, where it has probability $\frac{1}{2}$ of winning the election. If the true height of the landscape at the position selected by the challenger is above (below) $\frac{1}{2}$, the challenger (incumbent) wins the election. If the height is exactly $\frac{1}{2}$, each political party has a probability $\frac{1}{2}$ of winning the election. This procedure is then repeated for each election. Figure 1 shows some simulation results (taken from Sadiraj *et al.* 2006) for $K = 5$, $\mathcal{S} = \{0, 0.5, 1\}$, $N = 301$, 20 elections, 10 polls of 30 voters (10%) per election and 100 trials.

The solid lines in Figure 1 show the value of four different measures describing the outcomes of the model averaged over 100 trials. For each trial a new configuration of voters is drawn. The measure 'convergence' (upper left panel)

[1] The model, results and discussion in this section are taken from Sadiraj *et al.* (2006).

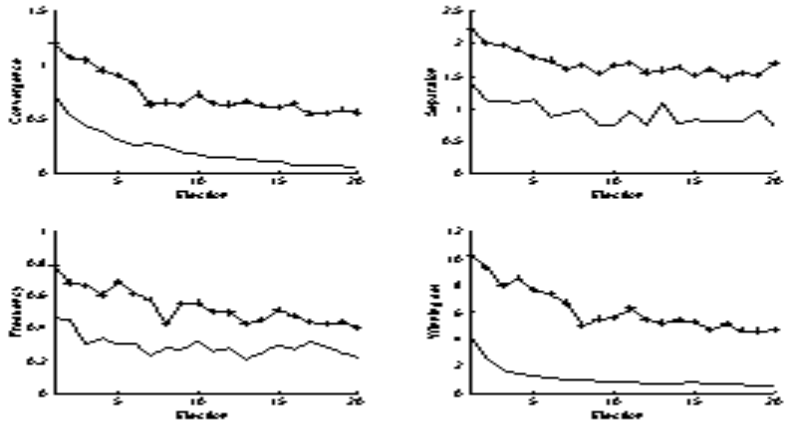


Figure 1: Time series of different measures, averaged over 100 trials, for the basic moden (−) and for the interest group model (+).

gives the Euclidean distance between the election outcome and the *center* of the distribution and decreases in the number of elections. The measure 'separation' (upper right panel) gives the Euclidean distance between the incumbent and challenger, which is also decreasing over time. The lower left panel shows the empirical frequency of election victories for the challenger and the lower right panel shows the size of the winning set (i.e. the number of positions that defeats the incumbent). These simulation results replicate the finding of Kollman *et al.* (1992) that policy platforms tend to converge to the center of the distribution of voter preferences.

2.2 Interest groups

We model interest groups as endogenously emerging institutions, arising from social interaction and dissatisfaction. Our approach differs from most of the literature which focuses on lobbying and campaign contributions and uses game theoretic models to describe the interaction between political parties and interest groups.

Interest groups emerge as follows. Voters with the same ideal position on one of the issues may decide to organize in interest groups in order to play a role in determining the election outcome. Now let n_k^i be the total number of voters having position $k \in \{1, \ldots, K\}$ on issue $i \in \{1, 2\}$. Clearly, $\sum_{k=1}^{K} n_k^1 = \sum_{k=1}^{K} n_k^2 = N$. Along each of the $2K$ "lines" in the issue space an interest group emerges. Prior to each election interest group formation takes place, where each voter determines whether he\she joins zero, one or two interest groups. After this process of interest group formation is over, total funds collected by the interest groups determines which interest groups become "active".

Joining an interest group provides a means to exert some political influence. Since this influence in interest group size, so is the willingness to join, which might even be reinforced by identification with the group. Furthermore, we assume voters are more inclined to join if the current policy position is farther away from their own position on that issue. On the other hand, there may be costs c of joining, which we assume to be exogenous and fixed. The process is now modeled as follows. Potential members are drawn in a random order and sequentially determine whether to join or not. This procedure is repeated once, so each voter decides whether to join or not one or two times. Let $m^i_{k,s-1}$ (with $m^i_{k,0} = 0$) be the size of the interest group at position k of issue i after $s-1$ voters have decided. The s'th voter then decides on the basis of a decision rule of the form $v^i_{js}(k) = V\left(\frac{m^i_{k,s-1}}{n^i_k}, k - y_i, c\right)$ which is increasing in s_{ji}, $k - y_i$ and $m^i_{k,s-1}/n^i_k$ and decreasing in c. This process leads to $2K$ different interest groups with typically different sizes. The total size of the interest groups decides which of them become active.

Interest groups try to influence the election process by coordinating voting behavior of their members and, conditionally, providing information about the electoral landscape to the candidates. Each active interest group finances some polls of the challenger conditional on: *i)* that the challenger runs these polls in policy positions coinciding with the interest group's position on the relevant issue; *ii)* that the challenger commits to select the platform with the highest poll result, provided this platform has a height of at least $\frac{1}{2}$. The interest group's members vote according to the interest group's advice, which is determined as follows. If one candidate is closer to the interest group's position than the other candidate, the former is supported. If the distance of the candidates from the interest groups positions on the relevant issue is the same,interest group members votes according to (1).

During an electoral campaign the challenger also runs some polls at randomly selected policy positons, next to those financed by the interest groups. It then chooses the position with the best polling result. All voters organized in interest groups vote according to the interest group's advice (if they belong to two interest groups they follow the interest group with the highest value of $v^i_{js}(\cdot)$, all other voters vote according to (1). The party with the majority of votes wins the election.

The simplifying assumptions about the symmetry and the uniform distribution of preferences, as well as the small number of issues and positions typically imply that the *generalized median voter* exists. The position of this median voter, once located, can not be defeated by any other platform, and will be reached with probability 1 because of the finiteness of the issue space. The model therefore predicts that, in the absence of interest groups, the incumbent converges to the median in the long run. One effect of interest groups is that typically they increase the winning set (see Sadiraj *et al.* 2005), since interest groups are more likely to form far away from the incumbent and hence tilt the electoral landscape at the expense of the incumbent. This leads to a higher

probability for the challenger to win an election. Furthermore, if the location of the incumbent favors the organization of the median voters, conditional polling makes sure that the median is located much faster than in the benchmark model. On the other hand, if the distribution of voters allows for formation of interest groups asymmetric to the median and cycles in winning platforms may appear. Consider for example the case where, once the incumbent is at the median, two groups located on different issues and different from the median organize in interest groups. The policy position corresponding to their intersection may then in fact defeat the center, only due to the fact that interest groups coordinate voting behavior. Figure 1 confirms this intuition conveyed above. The winning set for the interest group model is larger, and consequently, so is separation between platforms and the probability that the challenger wins.

3 Mean dynamics

The dynamics of the computational model depends critically upon the random initial configuration of the population of voters. We now derive analytical results by replacing stochastic variables by their expected values and study the resulting stationary Markov process, which can be seen as the deterministic skeleton of the original process. These so-called *mean dynamics* can give us useful information about the stochastic electoral competition model.

3.1 Electoral competition as a Markov process

Consider again a population of N voters, with ideal positions x_j drawn from the uniform discrete distribution on $\mathcal{X} = \{1, 2, \ldots, K\} \times \{1, 2, \ldots, K\}$ (with K odd). The distribution of strengths $s_j = (s_{j1}, s_{j2}) \in \mathcal{S} \times \mathcal{S}$ satisfies $p_s = \Pr(S_{j1} = s_{j1}, S_{j2} = s_{j2}) = \Pr(S_{j1} = s_{j1}) \Pr(S_{j2} = s_{j2}) = p_{s_1} p_{s_2}$. Let $y^{t-1} \in \mathcal{X}$ be the winning platform of the election at time $t-1$.

Definition 1 *Let* $\mathcal{R} = \{R : \exists i_1, i_2 \in \{1, \ldots, K\} \text{ s.t. } R^2 = i_1^2 + i_2^2\}$ *and* $\mathcal{U} = \{U(R), R \in \mathcal{R}\}$ *the family of subsets of* $\mathcal{X}$ *with* $U(R) = \{x \in \mathcal{X} : \|x - C\| = R\}$.

We use $\mathcal{U}$ as the new state space since, due to symmetry, all platforms that belong to the same element $U(R)$ can be treated equivalently. Notice that $\mathcal{U}$ has only $n = \sum_{k=1}^{\frac{1}{2}(K+1)} k << K^2 = |\mathcal{X}|$ elements and that each element of $\mathcal{U}$ contains 1, 4 or 8 elements of $\mathcal{X}$.

Proposition 2 *The family* $\mathcal{U}$ *satisfies the following properties.*

i) It forms a partition for the space $\mathcal{X}$.

ii) For all R *and* R' *and for all* $y^t, (y^t)' \in U_R$,

$$\Pr\left(y^{t+1} \in U_{R'} \mid y^t\right) = \Pr\left(y^{t+1} \in U_{R'} \mid \left(y^t\right)'\right).$$

The second property states that the probability of moving from any platform z in U_R to platforms in $U_{R'}$ is independent of the particular platform z. The electoral competition model corresponds to a Markov process with stationary transition probabilities on $\mathcal{U}$. Denote the n elements of $\mathcal{U}$ as $\{U_1, \ldots, U_n\} \equiv \left\{U(0), U(1), U\left(\sqrt{2}\right), \ldots, U\left(\sqrt{2}K\right)\right\}$. We can then, for given political institutions, voter preferences and interest group formation process, compute the $n \times n$ transition matrix P_r, where r indicates the number of polls. Element (i, j) of P_r gives the probability that, if the incumbent is in U_i, the election outcome will be in U_j.

Let the initial policy platform, y^0, follow some discrete distribution π_0 on $\mathcal{U}$. Then, at election t, the distribution of policy platforms over the different states U_i is given by $\pi_t = \pi_0 (P_r)^t$. We are, for every time period t, interested in two variables: the distance of the incumbent from the center of the issue space, which is given by $E(\|y^t - C\|) = \sum_{R \in \mathcal{R}} R\pi_t$ and the probability that the challenger wins the election, which is $\Pr(\text{the challenger wins at time } t) = \pi_t w$, where $w = (w_R)_{R \in \mathcal{R}}$ with $w_R = \Pr(\text{challenger wins} \mid \text{incumbent} \in U_R)$. An algorithm outlining how to compute the transition matrix P_r and the vector w, for the different models can be found in Sadiraj (2002).

In the next two subsections we will investigate the mean dynamics for the benchmark model and the interest group model for $K = 5$, and $\mathcal{S} = \left\{0, \frac{1}{2}, 1\right\}$. with $\Pr(s_{ji} = 0) = \Pr(s_{ji} = 1) = \frac{1}{4}$ and $\Pr\left(s_{ji} = \frac{1}{2}\right) = \frac{1}{2}$, for $j \in \{1, 2, \ldots, N\}$. The state space becomes $\mathcal{U} = \left\{U_R \mid R \in \left\{0, 1, \sqrt{2}, 2, \sqrt{5}, 2\sqrt{2}\right\}\right\}$ and we assume that the initial policy position is drawn from the uniform distribution on $\mathcal{X}$ which implies $\pi_0 = \left[\frac{1}{25}, \frac{4}{25}, \frac{4}{25}, \frac{4}{25}, \frac{8}{25}, \frac{4}{25}\right]$.

3.2 Dynamics for the benchmark model

For the benchmark model without interest groups and $r = 10$ random polls we obtain

$$P_{10} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.400 & 0.600 & 0 & 0 & 0 & 0 \\ 0.400 & 0.543 & 0.057 & 0 & 0 & 0 \\ 0.250 & 0.495 & 0.253 & 0.002 & 0 & 0 \\ 0.152 & 0.533 & 0.308 & 0.006 & 0.001 & 0 \\ 0.090 & 0.407 & 0.422 & 0.001 & 0.080 & 0 \end{pmatrix}, w_{(10)} = \begin{pmatrix} 0.50000 \\ 0.70000 \\ 0.97174 \\ 0.99878 \\ 1.00000 \\ 1.00000 \end{pmatrix}$$

Let $P_r(i, j)$ be the element in the i-th row and j-th column of P_r. Then $P_r^n(i, i) = [P_r(i, i)]^n$, since P_r is a lower diagonal matrix. Hence, for all $i = 2, \ldots, 6$, $\sum_n P_r^n(i, i) < \infty$ as a geometric series with term $|P_r(i, i)| < 1$. Thus all states U_R, $R > 0$ are transient since transience (persistence) of a state j is equivalent to $\sum_n P^n(j, j) < \infty$ $(= \infty)$ Furthermore, $P_{11} = 1$ implies that $\{U_0\}$ is a closed set[2] and U_0 a persistent state. Thus, the stationary distribution

[2] A set B in S is *closed* if $\sum_{j \in B} P(i, j) = 1$ for $i \in B$: once the system enters B it cannot leave.

is $\pi^* = [1, 0, 0, 0, 0, 0]$, and in the long run: *(i)* the policy platform ends up in C and stays there forever; and *(ii)* the probability the challenger wins at an election converges to $0.5 (= \lim_{t\to\infty} \pi_t w = \pi^* w = w_0)$.[3]

3.3 Dynamics for the model with interest groups

Interest groups influence the election process by: *(i)* coordinating voting behavior; *(ii)* providing information about the electoral landscape to the candidates; and *(iii)* putting conditions on polling. In order to be able to disentangle the impact of the latter from the first two we present results of the model with interest groups for 'unconditional' and 'conditional' polling separately. For analytical tractability we assume that all voters organize in interest groups and all interest groups become active.

3.3.1 Unconditional polling

Our first research question is to investigate the effects of the new (if any) properties of the electoral landscape in the dynamics of the electoral outcomes. For this we assume that the challenger runs r random polls. It should be clear by now that this case is exactly the same as the benchmark model, corrected for the fact that strength profiles of interest group members change from s to $(1, 0)$ or $(0, 1)$. The transition matrix, P^I_{10} and the vector, $w^I_{10^u}$ of winning probabilities for the model with interest groups, turns out to be

$$P^I_{10^u} = \begin{pmatrix} 1.000 & 0 & 0 & 0 & 0 & 0 \\ 0.152 & 0.848 & 0 & 0 & 0 & 0 \\ 0.400 & 0.425 & 0.176 & 0 & 0 & 0 \\ 0.007 & 0.443 & 0.407 & 0.142 & 0 & 0 \\ 0.152 & 0.444 & 0.307 & 0.007 & 0.090 & 0 \\ 0.028 & 0.407 & 0.542 & 0.000 & 0.023 & 0 \end{pmatrix}, w^I_{(10^u)} = \begin{pmatrix} 0.50000 \\ 0.99878 \\ 0.99985 \\ 1.00000 \\ 1.00000 \\ 1.00000 \end{pmatrix}$$

As for the basic model, we find that there is one and only one closed set, the elements of which are all persistent states, which is $\{U_0\}$. All states $U \in \mathcal{U} \backslash U_0$ are transient. However, there is a difference in the speed with which the system convergence to the center as the following shows. Figure 2 gives, for the 3 different cases, diagrams with $E(\|y^t - C\|)$ and $\Pr$ (the challenger wins at time t), respectively. First consider the left panel of Figure 2. From the highest to the lowest curve we have: benchmark model with 2 random polls, interest group model with 10 random polls, benchmark model with 10 random polls. From this we find that an increase in the number of (unconditional) polls decreases the expected separation between the winning platform and the center of the distribution. Secondly, for the interest group model expected separation is larger than for the basic model with the same number of polls. For the right panel of Figure 2 the highest to the lowest curve (as measured at election 6)

[3] Recall that, if the challenger does not find a platform with $h(z \mid y) > 0.5$, it chooses the incumbent's platform y.

are respectively: the interest group model with 10 random polls, the basic model with 10 polls and the basic model with 2 polls. From this it follows that the presence of interest groups increases the probability of winning an election. One of the findings in Sadiraj *et al.* (2006) was that the presence of interest groups appears to increase the winning set. That result is confirmed here as well. Given the state of the incumbent, we find that the size of the winning set equals: *(a)* $\begin{pmatrix} 0 & 1 & 5 & 9 & 14 & 21 \end{pmatrix}'$ for the basic model, and *(b)* $\begin{pmatrix} 0 & 9 & 11 & 17 & 19 & 22 \end{pmatrix}'$ for the model with interest groups (recall that $|\mathcal{X}| = 25$). Figure 2 suggests that typically the size of the winning set increases in the presence of interest groups.[4]

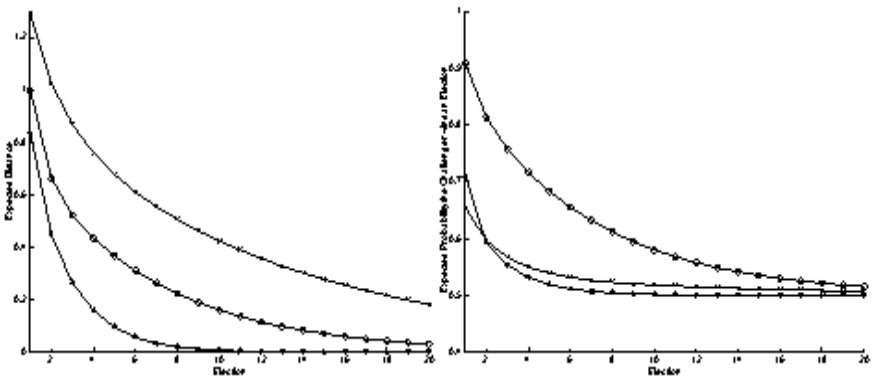


Figure 2: **Left panel**: Time series of the expected distance between the incumbent and the center of the space. The curve $-\times$ denotes the benchmark model with 2 polls, the curve $-\circ$ denotes the interest group model with 10 polls and $-*$ denotes the benchmark model with 10 polls. **Right panel**: Time series of the expected probabilities with which the challenger defeats the incumbent. The curve $-\times$ denotes the benchmark model with 2 polls, the curve $-\circ$ denotes the interest group model with 10 polls and $-*$ denotes the benchmark model with 10 polls.

3.3.2 Conditional polling

As mentioned above, the interest groups influence the election process by providing information about the electoral landscape to the political parties. The interest groups finance polls ran by the challenger conditional on: *i)* running a number of polls[5] in policy positions coinciding with the interest group's position

[4] The result is robust to changes in all parameter settings we have investigated. We have derived similar results for different distributions p on $\mathcal{S}$, and different number of positions per issue ($K \in \{3, \ldots, 11\}$).

[5] Remember that the number of polls that an interest group can finance is determined by the cost of running a poll and the size of the fund that the group possesses.

	5	×	×	.538	×	×
	4	×	×	.590	.538	.508
Issue 2	*3*	×	.500	.575	.500	×
	2	×	×	.590	.538	.508
	1	×	×	.538	×	×
		1	*2*	*3*	*4*	*5*

Issue 1

Table 1: Fractions of voters who prefer a position $z = (i, j)$ to $(2, 3)$ ($\times$ refers to fractions less than 0.5).

on the relevant issue; *ii)* commitment of the challenger to select the platform with the highest poll result, if this platform has a height of at least $\frac{1}{2}$. Furthermore, it is assumed that each interest group knows the median of the distribution of its group's members on the other issue and finances a poll there. Let r_1 be the number of random polls and r_2 the number of conditional polls. Let the challenger first run r_2 conditioned polls and then r_1 random polls. Removing from the policy space the positions where the conditioned polls are run one can compute the transition probabilities for the conditional polling procedure.

For the specified model and $r_2 = 8, r_1 = 2$, we find

$$P^I_{10^c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0.882 & 0.118 & 0 & 0 & 0 \end{pmatrix}, w^I_{10^c} = \begin{pmatrix} 0.5 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

A new persistent state appears. In addition to state U_0 which remains a persistent state with the property '$\{U_0\}$ is a closed set', state U_1 becomes a persistent state as well with the property '$\{U_1\}$ is a closed set'. This can be derived as follows. The transition matrix shows that if the system at election t is in one of the states $U_R, R \in \{1, 2, \sqrt{5}\}$, then at election $t + 1$ it will be in U_1 and stay there forever. If the system starts at $U_{2\sqrt{2}}$ then, with probability 0.882, in the coming election it will end up in U_1 and never leave that state. The probability that the system will settle in U_1 is given by the first coordinate of $\pi_0 P^I_{10^c}$ and equals 0.781. In the same way one can derive that the system will settle in U_0 with probability 0.219. Furthermore, let the incumbent platform be $y = (2, 3) \in U_1$.[6] Table (1) shows the fraction of votes that the challenger gets if he selects a position $z = (i, j)$, $i, j = 1, \ldots, 5$, ($\times$ refers to fractions of votes smaller than 0.5). Thus, the winning set that corresponds to a position $y \in U_1$ has always at least two elements from U_1 with the highest fraction of votes. Let us now consider the interest group located at position 2 on the second

[6] It should be clear (for reasons of symmetry) that Table 1 for a $y \in U_1$ is the same as the one derived by rotating Table 1 around the center $(3, 3)$ until $(2, 3)$ reaches y.

issue. From the uniformity of the distribution of voters in the space and the homogeneity[7] of voters within types, it follows that the median of the members of this interest group related to the first issue is located at 3. Hence, that interest group will finance a poll at position $(3, 2)$. Note that the altitude at $(3, 2)$ is .59, which is the highest value in Table 1. Thus, the incumbent platform in the coming election will be either $(3, 2)$ or $(3, 4)$. This means that although the incumbent does not leave the U_1 set, a voting cycle appears. Therefore we may conclude that, with probability .781, *(i)* a cycle emerges and *(ii)* the challenger wins with probability 1.

4 Voting cycles driven by interest groups

The 'mean dynamic' analysis from the previous section shows that for the specified parameters of the models, there is only one closed set, $\{U_0\}$ in the benchmark model. However, under conditional polling, there are two closed sets, $\{U_0\}$ and $\{U_1\}$, for the model with interest groups. This raises the question of the dependence of this result on the parameter specification, like the size of the space, the set of strengths, the probability distribution of strengths on that set and so on. The following result provides an answer to that question (for a proof, see Sadiraj, 2002).

Proposition 3 *Assume the distribution of strengths satisfies*

$$\frac{(1-\frac{1}{K})\sum_{s\in\mathcal{S}\setminus\{0\}} p_s^2 + \left(\frac{3}{2}-\frac{1}{K}\right)\sum_{(s_1/s_2)=2} p_{s_1}p_{s_2}}{\sum_{s\in\mathcal{S}\setminus\{0\}} p_s} > \frac{1}{2}. \tag{2}$$

Then, for both models, with and without interest groups, $\{U_0\}$ is a closed set and U_0 is a persistent state. For the model without interest groups, all other states, $U \in \mathcal{U} \setminus U_0$ are transient and in the presence of interest groups and given conditional polling, $\{U_1\}$ is a closed set and U_1 is a persistent state.

It is straightforward to check that (2) holds for the models from the previous section. Hence, the results shown in Section 3 follow directly from Proposition 3. We conclude that voting cycles emerge, once the incumbent visits U_1.

5 Simulations and empirical illustration

The law of large numbers ensures us that the mean-analysis is relevant for populations that are large enough to correct for random deviations. However, the population of voters may not be large enough to cancel out random fluctuations, and therefore, the law of large numbers may not always apply. This may have consequences at the macro-level. That is why in this section we will consider

[7] Voters of some type s and with the same ideal positions on some issue i, make the same decisions to join the relevant interest group.

some simulations for different realizations of voter preferences and investigate whether the predictions of Proposition 3 are valid. Furthermore, we will compare these simulation results to some empirically observed policy outcomes.

Each trial starts with drawing a population of 1000 voters from the uniform distribution on $\mathcal{X}$, where we again assume $K = 5$. The initial position of the incumbent is chosen to be the center, in order to be able to investigate the closeness property of this center for the different models. Each trial was run for 20 elections and we have done 20 different trials. Typical results are represented in panels *(a)-(d)* of Figure 3. Panels *(a)* and *(b)* show that in the basic model the incumbent remains at the center for all elections. This is a robust feature of all trials with the basic model. Panels *(c)* and *(d)* of Figure 3 show that in the interest group model something different occurs: counter to the first statement in Proposition 3, the incumbent leaves the center and positions itself at some other position. This happens in more than half of all trials. From these figures it is apparent that for the basic model, the set that contains the center of the issue space, $\{C\}$, is a closed set even for the stochastic model. However, for the interest group model, the center loses that property for certain realizations of the distribution of voter preferences. For our issue space of 25 positions, simulations show that: for the basic model the property that $\{U_0\}$ is a closed set is maintained if the size of the population is larger than 300; for the model with interest groups, $\{U_0\}$ and $\{U_1\}$ are closed sets if the size of the population is larger than 10000. For populations with size smaller than 1000, neither $\{U_0\}$ nor $\{U_1\}$ are closed sets. Our next step is to relate these simulation results to some empirical data on policy outcomes. An analysis of the policy outcomes for 20 European countries was done in Woldendorp, Keman and Budge (1998). They classified the composition of the government as falling into one of 5 categories, ranging from extreme left (category 1) to extreme right (category 5). The graphs represented in panels *(e)* and *(f)* of Figure 3, respectively, correspond to Iceland data and Finland data, starting with the first time the composition of the government is in the center (position 3) after 1960. We draw attention to two features present in the data from both countries: *(i)* the government composition stays longer at position 3 than at the other positions, that is, the center presents a position which is hard to defeat; *(ii)* although the government composition locates at position 3 it does not stay there forever, that is, the center can be defeated. Comparing these graphs to the graphs generated by the simulations it is clear that the data generated by the interest group model represents the empirical data best. In our view, this provides some support for the model with interest groups presented in Section 2.

6 Concluding remarks

Although simulations provide a valuable aid in characterizing the behavior of the electoral competition model, their power is limited to the domain of the selected parameters. An understanding of the more generic properties of

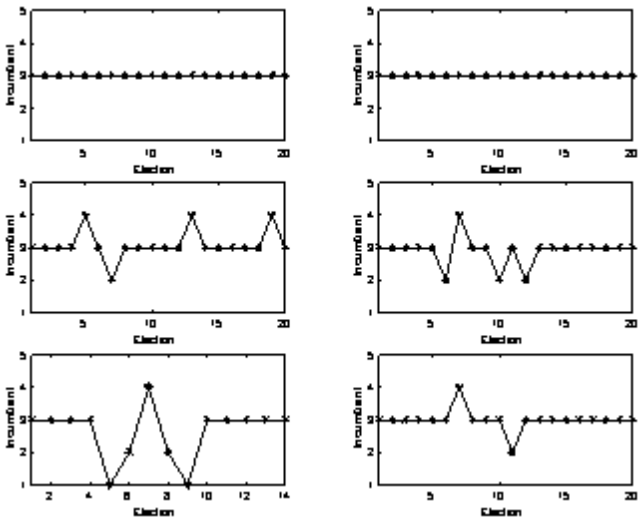


Figure 3: The stability of the center:simulation and empirical data. Panels (a) and (b) show data generated from the benchmark model in simulations 13 and 14, respectively. Panels (c) and (d) show data generated from the model with interest groups in simulations 13 and 14, respectively. Panels (e) and (f) show data generated from the composition of the governments in Finland and Iceland, respectively.

individual-based models requires the use of deterministic approximation models. In this paper we have applied a mean-field approximation to the stochastic models presented in Section 2, by replacing the values of the random variables by their expected values. This leads to deterministic dynamic models of the Markov–type. The main results obtained from the analysis of the deterministic models are as follows. The dynamics of the distance between the policy outcome and the center of the space, and of the probability that the challenger wins an election, replicate qualitatively the respective dynamics generated by the stochastic computational model. For both models, with and without interest groups, the set consisting of the center of the space presents a closed set. For a broad class of probability distributions on a set of strengths $\mathcal{S}$ and under conditional polling, it is shown that *(i)* the set of positions at distance 1 from the center is a closed set for the model with interest groups, and *(ii)* a voting cycle emerges. For the specified model the voting cycle appears with probability .781. To our knowledge, this is the first study pointing at, and providing a micro-foundation for, the possibility of a voting cycle in the presence of a dominant point. A further investigation shows, that if the size of the population is lower than some threshold (1000 for our specified model) voting cycles become frequent phenomena and expand all over the issue space. Our model

positions itself in the series of models that point at the electoral instability of voting outcomes.

The inherent property driving our results is that the winning set (i.e., the set of policy platforms that will defeat the current incumbent) increases in the presence of interest groups. This happens in all the stochastic and numerical simulations. Moreover, in Sadiraj *et al.* (2005) it is rigorously shown that, in a slightly different spatial competition framework and under certain mild conditions on the incumbent's position, the winning set for the challenger indeed increases when interest groups are present to coordinate voting behavior.

References

[1] K. Kollman, J.H. Miller and S.E. Page (1992): Adaptive parties in spatial elections. *American Political Science Review* **86**, 929-937.

[2] V. Sadiraj (2002): *Essays in Political and Experimental Economics.* PhD Thesis, University of Amsterdam.

[3] V. Sadiraj, J. Tuinstra and F. van Winden (2005): On the size of the winning set in the presence of interest groups. *CeNDEF Working paper 05-08* University of Amsterdam.

[4] V. Sadiraj, J. Tuinstra and F. van Winden (2006): A computational electoral competition model with social clustering and endogenous interest groups as information brokers. *Public Choice*, **129**, 169-187.

[5] J. Woldendorp, H. Keman and I. Budge (1998): Party government in 20 democracies: an update (1990-1995). *European Journal of Political Research* **33**, 125-164.

Domains of social choice functions on which coalition strategy-proofness and Maskin monotonicity are equivalent[1]

Koji Takamiya

Abstract

It is known that on some social choice and economic domains, a social choice function is coalition strategy-proof if and only if it is Maskin monotonic. This equivalence provides a computational merit: Replacing coalition strategy-proofness with Maskin monotonicity significantly reduces the time required to check whether the social choice function is coalition strategy-proof, or not. This paper studies the foundation of those equivalence results. I provide a set of conditions which is sufficient for the equivalence between coalition strategy-proofness and Maskin monotonicity. Further, applying these conditions, I provide a class of domains, called "essentially strict domains," on which this equivalence holds. This yields some known and new results. An "essentially strict domain" is a domain such that each individual is endowed with a partition over the set of alternatives, and the preferences admissible to this individual are exactly the strict rankings over this partition.
Keywords— social choice function, coalition strategy-proofness, Maskin monotonicity.

0 Introduction

This paper examines logical relations between coalition strategy-proofness and Maskin monotonicity of social choice functions. Coalition strategy-proofness is a strong requirement of incentive compatibility. A social choice function is said to be **coalition strategy-proof** if no group of individuals can benefit from jointly misrepresenting their preferences, in other words, cannot manipulate the final outcome. A social choice function is said to be **Maskin monotonic** if the outcome to be chosen by the function does not vary whenever each individual switches his preference keeping or improving the relative ranking of that outcome. This property is very important in implementation theory. For example, it is well-known as a necessary condition for Nash implementation (see Maskin, 1985, 1999).

[1]This paper is based on my previous paper with the same title (Institute of Social and Economic Research Discussion Paper Series, ♯668), which is forthcoming in *Economics Letters*. Part of the research has been done while I visited to the Indian Statistical Institute, Delhi Centre. I am very grateful to Arunava Sen for his hospitality and helpful comments. I thank the people of the institute for their hospitality. And I am grateful to two reviewers of COMSOC 2006 for valuable comments. All errors are my own responsibility.

It has been observed that these two properties are strongly related to each other. The classical result by Muller and Satterthwaite (1977) asserts that on the unrestricted strict preference domain, a social choice function is strategy-proof if and only if it is Maskin monotonic.[2] Since on this domain, strategy-proofness is equivalent to coalition strategy-proofness, the theorem states the equivalence between coalition strategy-proofness and Maskin monotonicity.

In more recent studies, it has been pointed out that coalition strategy-proofness and Maskin monotonicity are equivalent even in some environments where coalition strategy-proofness is strictly stronger than strategy-proofness. Takamiya (2001, 2003) pointed out the equivalence between coalition strategy-proofness and Maskin monotonicity holds for allocation rules in a certain broad class of economies with indivisible goods, which includes some notable problems such as "marriage problems" (Gale and Shapley, 1962) and "housing markets" (Shapley and Scarf, 1974). (See also Svensson (1999) for related results.) Also in some other environments, for example the classical exchange economies, it is known that coalition strategy-proofness implies Maskin monotonicity (e.g. Barberà and Jackson, 1995).

There is an evident **computational** merit in replacing coalition strategy-proofness with Maskin monotonicity, in particular in such cases where coalition strategy-proofness is strictly stronger than strategy-proofness. To check whether the function is coalition strategy-proof by the straightforward method, one needs to check whether each **coalition** has a chance of manipulation. On the other hand, to check whether the function is Maskin monotonic or not, it only requires to check for each **individual** whether a monotonic change of his preference alters the outcome to be outputted. Evidently, the number of coalitions grows exponentially as the number of individuals grows. Thus checking coalition strategy-proofness requires exponentially larger time than checking Maskin monotonicity. For this reason, replacing coalition strategy-proofness with Maskin monotonicity significantly reduces the time required to check whether the social choice function is coalition strategy-proof, or not.

The purpose of this paper is to study the foundation of these close relationships between coalition strategy-proofness and Maskin monotonicity. I examine what conditions the domain of the social choice function should satisfy in order to have the property that coalition strategy-proofness implies Maskin monotonicity, and its converse.

The main results of this paper provide two sufficient conditions. The first condition, which is referred to as Condition A, is a sufficient condition for that coalition strategy-proofness implies Maskin monotonicity. This condition requires the domain of the social choice function to satisfy two properties. The first property says that for any individual, and any preference admissible to him, if any two alternatives are indifferent under this preference, then these alternatives are indifferent under all the preferences admissible to him. In other

[2] "Strategy-proofness" requires that the social choice function cannot be manipulated by any single individual. That is, unlike coalition strategy-proofness, manipulations by groups are not necessarily ruled out.

words, this requires that every individual has a partition of the set of alternatives, and his admissible preferences contains only (but not necessarily all of) such preferences that any two alternatives are indifferent if, and only if, these alternatives are in the same cell of the partition. The second property that Condition A requires is that if for any preference profile in the domain, there is no pair of alternatives such that all individuals are indifferent between them. That is to say, if the alternative to be chosen shifts from one to another, then there is always someone who cares about this shift. This is equivalent to the requirement that the "join" (the coarsest common refinement) of the partitions of all the individuals that arise in the first half of this condition equals to the finest partition (i.e. the one in which each cell contains exactly one element).

The second condition, referred to as Condition B, is a sufficient condition for that Maskin monotonicity implies coalition strategy-proofness. This condition is defined as follows: Let any coalition be given. And pick up any preference profile for this coalition, which I call the first profile. Then let us fix any two alternatives, say x and y, such that y (weakly) Pareto dominates x within this coalition under the first profile. Further, pick up another arbitrary preference profile for this coalition, the second profile. Then this domain satisfies Condition B if the domain contains at least one preference profile for this coalition such that x keeps or improves its relative ranking from the first profile to this profile, and so does y from the second profile to this profile. Speaking very roughly, the third profile is a mixture of the first and the second profiles in the sense of the desirability of x and y. And Condition B requires such a mixture always exists.

Given these two sufficient conditions, we present a class of domains which satisfies both of these conditions. This class is the collection of those domains in which (i) every individual has a partition of the set of alternatives, and his admissible preferences are **exactly** such preferences that any two alternatives are indifferent if, and only if, these alternatives are in the same cell of the partition, and (ii) the join of all these partitions equals to the finest partition. Paraphrasing, such a domain is the maximal domain among those satisfying Condition A, given a list of partitions. I call these domains **essentially strict domains**.

I point out that essentially strict domains are assumed in some previous results. This observation unifies the the Muller-Satterthwaite theorem and the similar equivalence theorem by Takamiya (2003) in the context of the "generalized indivisible good allocation problems" (Sönmez, 1999), which cover various problems including well-known "housing markets" (Shapley and Scarf, 1974) and "marriage problems" (Gale and Shapley, 1962).

Furthermore, I present an application of the result regarding essentially strict domains, which, to my knowledge, has never been reported elsewhere.

1 Preliminaries

Let N denote the set of **individuals**. Assume that N is a nonempty finite set. Call any nonempty subset of N a **coalition**. Let X be the set of **alternatives** (social outcomes). X is nonempty and may be finite or infinite.

Let Q be a nonempty set. Then denote by $W(Q)$ the set of weak orderings (i.e. complete and transitive binary relations) on Q. And denote by $L(Q)$ the set of linear orderings (i.e. complete, tansitive and anti-symmetric binary relations) on Q.

For $i \in N$, call $R^i \in W(Q)$ a **preference relation** on Q of individual i. And a list $(R^i)_{i\in N}$ is called a **preference profile**. For $x, y \in Q$, xR^iy reads that to individual i, x is at least as good as y. As usual, P^i denotes the asymmetric part, and I^i denotes the symmetric part of R^i. Let $R^i \in W(Q)$ and $Q' \subset Q$. Then $\max R^i(Q)$ denotes the set of R^i-maximal elements in Q', $\{x \in Q' \mid \forall y \in Q', xR^iy\}$.

For $i \in N$, D^i denotes the set of **admissible preferences** of individual i. Assume that $D^i \subset W(X)$ for any $i \in N$. For $S \subset N$, D^S denotes the Cartesian product $\prod_{i\in S} D^i$. A **social choice function** (SCF) is a single-valued function $f : D^N \to X$. D^N is called the **domain** (of f).

Let f be a SCF. Let S be a coalition. Then we say that S manipulates f at a preference profile $R \in D^N$ if there exists some $R'^S \in D^S$ such that

$$[\forall i \in S,\ f(R^{-S}, R'^S)R^i f(R)] \ \&\ [\exists j \in S :\ f(R^{-S}, R'^S)P^j f(R)]. \tag{1}$$

Call f **coalition strategy-proof** if no coalition manipulates f at any $R \in D^N$.

For $R^i \in W(X)$, and $x \in X$, denote by $L(x, R^i)$ the set $\{y \in X \mid xR^iy\}$. That is, $L(x, R^i)$ is the lower-contour set of x relative to R^i. Call f **Maskin monotonic** if for any $i \in N$, any $R \in D^N$, and any $R^i, R'^i \in D^i$,

$$[f(R) = x \ \&\ \ L(x, R^i) \subset L(x, R'^i)] \Rightarrow f(R^{-i}, R'^i) = x. \tag{2}$$

We note that checking coalition strategy-proofness requires exponentially larger time than checking Maskin monotonicity: Let us bound the number of possible preferences of each individual, that is, for each i, $\left|D^i\right| \leq c$. And let the number of individuals, denoted by n, vary. Then checking coalition strategy-proofness requires to check $O(2^n c^{2n})$ times the relation (1) in the above. On the other hand, checking Maskin monotonicity requires to check the relation (2) $O(nc^{n+1})$ times. Although the both grow exponentially as n gets larger (this is because the preference domain grows larger exponentially), the former still grows exponentially faster with relative to the latter. This fact gives a **computational merit** in replacing coalition strategy-proofness with Maskin monotonicity.

2 Main results

This section presents the main results. These results provide sufficient conditions that the domain of the social choice function should satisfy to have the

property that coalition strategy-proofness implies Maskin monotonicity, and its converse. I introduce these two sufficient conditions. Let a SCF $f : D^N \to X$ be given.

Let $\mathcal{P}^i$ be a partition of X. For $x \in X$, let us denote the cell of $\mathcal{P}^i$ that contains x by $\mathcal{P}^i(x)$. Let $(\mathcal{P}^i)_{i\in N}$ be a profile of partitions of X. And let us denote by "$\bigvee$" the operation of taking the "join" (the coarsest common refinement) of the partitions.

Conditon A. There exists some profile of partitions $(\mathcal{P}^i)_{i\in N}$ such that for any $R \in D^N$, any $i \in N$ and any $x, y \in X$,

$$x \in \mathcal{P}^i(y) \Leftrightarrow xI^i y, \tag{3}$$

and

$$\bigvee_{i\in N} \mathcal{P}^i = \{\{x\} \mid x \in X\}. \tag{4}$$

In words, Condition A consists of two parts, which correspondence to the formulas (3) and (4), respectively:
(i) For any individual, and for any preference admissible to him, any two alternatives are indifferent under this preferences if, and only if, these alternatives are indifferent under all the preferences admissible to him; and
(ii) If for any preference profile, for any pair of alternatives there is at least one individual who is not indifferent between these alternatives.

Condition B. For any $S \subset N$ with $S \neq \emptyset$, any $\tilde{R}^S, \hat{R}^S \in D^S$, and any $x, y \in X$ such that $(\forall i \in S,\ y\tilde{R}^i x)$ and $(\exists i \in S :\ y\tilde{P}^i x)$, there exists $R^{\star S} \in D^S$ such that

$$\forall i \in S,\ L(x, \tilde{R}^i) \subset L(x, R^{\star i}) \ \&\ L(y, \hat{R}^i) \subset L(y, R^{\star i}). \tag{5}$$

In words, Condition B condition is defined as follows: Let any coalition be given. And pick up any preference profile for this coalition, which I call the first profile. Then let us fix any two alternatives, say x and y, such that y (weakly) Pareto dominates x within this coalition under the first profile. Further, pick up another arbitrary preference profile for this coalition, the second profile. Then the domain satisfies Condition B if the domain contains at least one preference profile for this coalition such that x keeps or improves its relative ranking from the first profile to this profile, and so does y from the second profile to this profile. Roughly speaking, the third profile is a mixture of the first and the second profiles in the sense of the desirability of x and y. And the condition requires such a mixture always exists.

Theorem 1 *Let D^N satisfy Condition A. Then if f is coalition strategy-proof, then f is Maskin monotonic.*

Proof Suppose that D^N satisfies Condition A and that f is not Maskin monotonic. Then I will show that f is not coalition strategy-proof. Since f is not Maskin monotonic, we have for some $i \in N$, some $R \in D^N$, and some $\tilde{R}^i \in D^i$,

$$L(x, R^i) \subset L(x, \tilde{R}^i) \ \& \ f(R^{-i}, \tilde{R}^i) \neq x, \tag{6}$$

where x denotes the alternative $f(R)$. Let us denote $f(R^{-i}, \tilde{R}^i)$ by y. Since D^N satisfies Condition A, there must be at least one individual j such that xP^jy or yP^jx. Thus the set $T = \{j \in N \mid \neg xI^jy\}$ is nonempty. Then there are two cases.

(i) Assume that $i \in T$. Then either xP^iy or yP^ix. Suppose that xP^iy holds true. Then $L(x, R^i) \subset L(x, \tilde{R}^i)$ and Condition A together imply $x\tilde{P}^iy$. That is, $f(R^{-i}, R^i)\tilde{P}^i f(R^{-i}, \tilde{R}^i)$, which says i manipulate at $(R^{-i}, \tilde{R}^i)$ by reporting R^i. Thus f is not coalition strategy-proof.

In turn, suppose that yP^ix holds true. Then similarly, i manipulate at R by reporting $\tilde{R}^i$, which violates coalition strategy-proofness again.

(ii) Assume that $i \notin T$. Then xI^iy. Let $j \in T$, which means either xP^jy or yP^jx. Suppose that xP^jy. Then similarly to the case (i), $\{i, j\}$ manipulates at $(R^{-\{i,j\}}, \tilde{R}^i, R^j)$ by reporting (R^i, R^j). In turn suppose that yP^jx. Then $\{i, j\}$ manipulates at $(R^{-\{i,j\}}, R^i, R^j)$ by reporting $(\tilde{R}^i, R^j)$. In either way, f is not coalition strategy-proof. □

Theorem 2 *Let D^N satisfy Condition B. Then if f is Maskin monotonic, then f is coalition strategy-proof.*

Proof Suppose that D^N satisfies Condition B and that f is Maskin monotonic but not coalition strategy-proof. Then there is some coalition S which manipulates at some $\tilde{R} \in D^N$ by reporting $\hat{R}^S \in D^S$. Let us denote the alternative $f(\tilde{R})$ by x, and $f(\tilde{R}^{-S}, \hat{R}^S)$ by y. Then clearly, $\forall i \in S,\ y\tilde{R}^ix$ and $\exists i \in S :\ y\tilde{P}^ix$. Thus Condition B implies that there is $R^{\star S} \in D^S$ such that for all $i \in S$,

$$L(x, \tilde{R}^i) \subset L(x, R^{\star i}), \tag{7}$$

$$L(y, \hat{R}^i) \subset L(y, R^{\star i}). \tag{8}$$

Since f is Maskin monotonic, (6) implies $f(\tilde{R}^{-S}, R^{\star S}) = x$. On the other hand, (7) implies $f(\tilde{R}^{-S}, R^{\star S}) = y$. These imply $x = y$, a contradiction. □

3 Further results

This section provides a class of domains which satisfies both Conditions A and B. Then applying the results presented in Section 2, some previous and new results will be derived.

Essentially strict domain. Let $(\mathcal{P}^i)_{i\in N}$ be a profile of partitions of X. Then D^N is said to be the **essentially strict domain with respect to** $(\mathcal{P}^i)_{i\in N}$ if D^N is the collection of all the preference profiles R that satisfy for any $x, y \in X$,

$$x \in \mathcal{P}^i(y) \Leftrightarrow xI^iy, \tag{9}$$

and $(\mathcal{P}^i)_{i\in N}$ satisfies

$$\bigvee_{i\in N} \mathcal{P}^i = \{\{x\} \mid x \in X\}. \tag{10}$$

To paraphrase, the essentially strict domain with respect to $(\mathcal{P}^i)_{i\in N}$ is the (inclusion) maximal domain among those which satisfy Condition A given $(\mathcal{P}^i)_{i\in N}$.

It is less obvious that such a domain satisfies Condition B.

Lemma 1 *If D^N is the essentially strict domain with respect to $(\mathcal{P}^i)_{i\in N}$, then D^N satisfies Condition B.*

Proof Let $S \subset N$ with $S \neq \emptyset$, and $x, y \in X$. Let $\tilde{R}^S \in D^S$ such that $(\forall i \in S,\ y\tilde{R}^i x)$ & $(\exists i \in S,\ y\tilde{P}^i x)$, and $\hat{R}^S \in D^S$. To show that D^N satisfies Condition B, we will give $R^{\star S} \in D^S$ such that

$$\forall i \in S,\ L(x, \tilde{R}^i) \subset L(x, R^{\star i}) \ \&\ L(y, \hat{R}^i) \subset L(y, R^{\star i}). \tag{11}$$

Let $S^+ = \{i \in S \mid y\tilde{P}^i x\}$, and $S^0 = \{i \in S \mid y\tilde{I}^i x\}$. Let $R^{\star S}$ be such that for each $i \in S^+$,

$$\max R^{\star i}(X) = \mathcal{P}^i(y), \text{ and} \tag{12}$$

$$\max R^{\star i}(X \setminus \max R^{\star i}(X)) = \mathcal{P}^i(x), \tag{13}$$

and for each $i \in S^0$,

$$\max R^{\star i}(X) = \mathcal{P}^i(y). \tag{14}$$

Note that for $i \in S^0$, $\mathcal{P}^i(x) = \mathcal{P}^i(y)$. Then evidently, $L(y, R^{\star i}) = X$ for any $i \in S$; $L(x, R^{\star i}) = X$ for any $i \in S^0$; and $L(x, R^{\star i}) = X \setminus \mathcal{P}^i(y)$ for any $i \in S^+$. Note that $L(x, \tilde{R}^i) \cap \mathcal{P}^i(y) = \emptyset$. Thus $R^{\star S}$ satisfies (11). □

Now we obtain the following result applying Theorems 1 and 2.

Theorem 3 *Let D^N be an essentially strict domain. Then f is coalition strategy-proof if, and only if, f is Maskin monotonic.*

In the following, we will derive two known results and one new result as special cases of Theorem 3. First, let the partition profile $(\mathcal{P}^i)_{i\in N}$ be such that for each i, $\mathcal{P}^i$ is $\{\{x\} \mid x \in X\}$. Then the essentially strict domain with respect to $(\mathcal{P}^i)_{i\in N}$ coincides with $L(X)^N$. This yields the well-known Muller-Satterthwaite theorem.

Corollary 1 (Muller and Satterthwaite, 1977) *Let $D^N = L(X)^N$. Then f is coalition strategy-proof if, and only if, it is Maskin monotonic.*

The second application is for "generalized indivisible good allocation problems," as defined in Sönemz (1999). This class of allocation problems contains well-known "marriage problems" (Gale and Shapley, 1962) and "housing markets" (Shapley and Scarf, 1974) as special cases. A **generalized indivisible good allocation problem** (an **allocation problem**, henceforth) is a list $(N, \Omega, \mathcal{A}, R)$.[3] Here N is the set of individuals, as we have defined in Section 1. Ω is the set of goods, which is assumed to be a nonempty finite set. An "allocation" is a set-valued function $x : N \to\!\!\to \Omega$ such that $\{x(i) \mid i \in N\}$ is a partition of Ω. $\mathcal{A}$ is the set of feasible allocations. And R is a preference profile belonging to the domain D^N defined as follows:

$$D^N := \{R \mid \forall i \in N,\ R^i \in W(\mathcal{A})\ \&\ (\forall x, y \in \mathcal{A},\ xI^i y \Leftrightarrow x(i) = y(i))\}. \quad (15)$$

That is, every individual has preferences that exhibit no consumption externalities, and are strict over their own assignments.

Obviously, in this case, D^N is an essentially strict domain with respect to $(\mathcal{P}^i)_{i\in N}$, where for each $i \in N$, $\mathcal{P}^i$ is the partition such that for any $x, y \in \mathcal{A}$, $x \in \mathcal{P}(y) \Leftrightarrow x(i) = y(i)$.

In this setting, we consider the set of allocation problems $\{(N, \Omega, \mathcal{A}, R) \mid R \in D^N\}$, and SCFs $f : D^N \to \mathcal{A}$. Then we have the following known result as a corollary to Theorem 3.

Corollary 2 (Takamiya, 2003) *Let f be a SCF in a setting of allocation problems. Then f is coalition strategy-proof if, and only if, it is Maskin monotonic.*

The third application is a class of settings which are interpreted as environments of "distributive work." Consider a domain D^N defined as follows: Fix an individual $d \in N$. And assume $D^d = L(X)$. This yields the partition $\mathcal{P}^d = \{\{x\} \mid x \in X\}$ which satisfies (9). For each $i \in N \setminus \{d\}$, let $\mathcal{P}^i$ be an arbitrary partition of X. Then let D^N be the essentially strict domain with respect to $(\mathcal{P}^i)_{i\in N}$. In words, D^N is an essentially strict domain in which at least one individual has exactly all the linear orderings over X as the admissible preferences.

This domain has an interpretation like the following: Imagine a team of people who are to complete one complex work. There is one distinct individual called a "director" who takes care of the whole picture. Each of the other members of the team is assigned his part of the work for which he takes the responsibility, and he only cares about the components of outcomes relevant to the part assigned to him. Thus he is indifferent over any two outcomes between which the components relevant to his part are the same. But the director takes care of all components thus his admissible preferences are strict over all outcomes.

[3] The complete definition of the problem includes initial endowments, which are superfluous for the present purpose thus dropped here.

Corollary 3 *Let f be a SCF in a setting of the distributive work environment. Then f is coalition strategy-proof if, and only if, it is Maskin monotonic.*

To my knowledge, this result has never been reported elsewhere. This shows that the introduction of essentially strict domains not only synthesizes known results but produces some new insights.

References

[1] Barberà S and Jackson MO (1995) Strategy-proof exchange. *Econometrica* 63: pp51–87

[2] Gale D and Shapley L (1962) College admissions and the stability of marriage. *American Mathematical Monthly* 69: pp9–15.

[3] Maskin E (1985) The theory of implementation in Nash equilibrium: A survey. Hurwicz L, Schmeidler D and Sonnenschein H (eds) *Social Goals and Social Organization.* Cambridge University Press, Cambridge, pp 173–04

[4] Maskin E (1999) Nash equilibrium and welfare optimality. *Review of Economic Studies* 66: pp23–38.

[5] Muller E and Satterthwaite M (1977) The equivalence of strong positive association and strategy-proofness. *Journal of Economic Theory* 14: pp412–18.

[6] Shapley L and Scarf H (1974) On cores and indivisibilities. *Journal of Mathematical Economics* 1: pp23–27.

[7] Sönmez T (1999) Strategy-proofness and essentially single-valued cores. *Econometrica* 67: pp677–89.

[8] Svensson L-G (1999) Strategy-proof allocation of indivisible goods. *Social Choice and Welfare* 16: pp557–67.

[9] Takamiya K (2001) Coalition strategy-proof and monotonicity in Shapley-Scarf housing markets. *Mathematical Social Sciences* 41: pp201–13.

[10] Takamiya K (2003) On strategy-proofness and essentially single-valued cores: A converse result. *Social Choice and Welfare* 20: pp77–83.

Koji Takamiya
Institute of Social and Economic Research
Osaka University
6-1 Mihogaoka Ibaraki Osaka 567-0047, Japan
Email: takamiya@iser.osaka-u.ac.jp

Small Binary Voting Trees

Michael Trick

Abstract

Sophisticated voting on a binary tree is a common form of voting structure, as exemplified by, for example, amendment procedures. The problem of characterizing voting rules that can be the outcome of this procedure has been a longstanding problem in social choice. We explore rules over a small number of candidates, and discuss existence and nonexistence properties of rules implementable over trees.

1 Introduction

The problem of characterizing voting rules implementable by backward induction (or *sophisticated voting*) has been a longstanding problem in social choice. Consider a set C of candidates from which an election will choose a winner. A *sophisticated voting tree* is a binary tree where at each node of the tree, voters choose between the two candidates who have survived the process to that node, where the process begins at the leaves and works towards the root. For instance, in the tree in figure 1, the voters begin by choosing between candidates b and c and the winner then is compared with a.

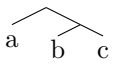


Figure 1: Simple Tree on Three Candidates

The relationship of this tree to backwards induction was developed by Dutta and Sen [4].

Given a voting tree, the winner of the election is clearly a function of the underlying majority tournament (for simplicity, we will assume that preferences are strict and there are an odd number of voters, so the majority tournament is complete and the winner is therefore well defined). But what functions, or voting rules, are implementable on trees (or, in this paper *implementable*, for short)? For instance, while any voting rule over three candidates that chooses from the top cycle of the tournament is implementable, similar results do not hold for four candidates. In particular, in figure 2, no voting rule that chooses candidate b for tournament 1 and a for tournament 2 is implementable on trees (in these diagrams, we draw an arc from i to j if i is preferred to j by the electorate). In fact, of the 16 pairs of possible winners over these two tournaments, only five pairs are implementable (reasons for this will be given later).

There have been many partial results on characterizing implementable rules. There are particular rules for which implementations are known. Moulin [9]

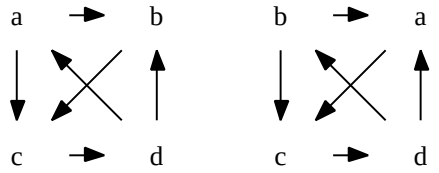


Figure 2: Pair of Tournaments over 4 Candidates

and Mueller [11] showed particular veto-type social choice functions are implementable, Herrero and Srivastava [6] showed that every rule over three candidates is implementable, Dutta and Sen [4] showed a particular rule choosing from the uncovered set is implementable, and Coughlan and Le Breton [2] extended this to a particular choice in the ultimate uncovered set.

A general characterization continues to be elusive. McKelvey and Niemi [7] showed that any onto voting rule must choose from the top cycle, and recognized that such a restriction is not sufficient. An effort towards sufficiency was Srivastava and Trick [13]. They characterized implementable voting rules defined on pairs of tournaments, and conjectured that pairwise implementation was sufficient for rules defined over all tournaments. A decade has passed since this result, and the conjecture remains open. The purpose of this paper is to provide computational results to both support the conjecture and to identify possible locations of counterexamples.

Section 2 of this paper outlines the known results on implementable rules. Section 3 outlines a computational approach to generating implementable rules and shows that there are exactly three non-isomorphic rules over three candidates (among all rules which choose the Condorcet winner when it exists). Section 4 then provides a series of results over tournaments on four candidates. The final section then outlines a research agenda for finally characterizing implementable rules.

2 Basic Results

Given a set C of n candidates, a *tournament* over C is a complete binary irreflexive relationship over C (so, for two candidates i and j, either iTj or jTi, but not both). In our case, the tournament summarizes the voting outcome between any pair of candidates giving which candidate is preferred by the electorate.

A *voting tree* is a binary tree where each leaf of the tree is labeled with some candidate from C. Given a tournament T, applying T to a voting tree means iteratively finding two leaves with a common parent, removing those leaves, and labeling the parent with the winner under T between the two leaf labels. This continues until the root of the tree is labeled. The resulting label is the winner of T relative to the tree.

Let $\mathcal{T}$ be a set of tournaments over C. A voting rule over T is a function $f : T \in \mathcal{T} \rightarrow \mathcal{C}$. f is an *implementable* rule if there exists a voting tree such

that when the tournament T is applied to the tree, $f(T)$ wins, for all $T \in \mathcal{T}$.

For a tournament T, the *top cycle* of T is the minimal subset of candidates with the property that every candidate in the subset beats every candidate outside the subset. If the top cycle of T is a singleton a, then a is the *Condorcet winner* for T.

It is clear that for an implementable rule f, if a is the Condorcet winner for a tournament $T_1 \in \mathcal{T}$, then either $f(T_1) = a$ or $f(T) \neq a$ for all $T \in \mathcal{T}$. The former occurs whenever the label a is applied anywhere in the voting tree; the latter when the label a is excluded from the tree. For this latter case, the rule is then defined on a subset of C. For simplicity, this paper will only be concerned with voting rules that choose the Condorcet winner when it exists. Equivalently, we are concerned with onto voting rules.

For Condorcet voting rules, an implementable rule must always choose from the top cycle. The example in figure 2 shows that condition is not sufficient for implementability, however. Srivastava and Trick [13] give a necessary and sufficient condition for implementability for rules defined over two tournaments. The condition is as follows:

Let T and T' be tournaments defined on a ground set C. We are concerned with voting rules defined over (T, T'). If there exists a binary voting tree that implements (i, j) (so i wins for T and j for T'), we write $(i, j) \in I$.

A set $S \subseteq C$ is *prime* (relative to T and T') if there does not exist a partition of S into two or more nonempty subsets such that $S = S_1 \cup S_2 \cup \ldots S_k$ and

1. $a \in S_i, b \in S_j, i \neq j$ implies either $aTb, aT'b$ or $bTa, bT'a$, and
2. $a \in S_i, b \in S_j, i \neq j, aTb$ implies $a'Tb'$ for all $a' \in S_i$ and $b' \in S_j$.

Intuitively, a prime set is a set that cannot be decomposed into subsets such that T and T' agree and are consistent in the relations between items in different subsets.

Let the top cycle of T restricted to a set $S \subseteq C$ be denoted as $tc(S)$ and the corresponding top cycle at T' as $tc'(S)$.

Theorem 1 *[13]* $(a, b) \in I$ *if and only if there exists a prime set* S *with* $a \in tc(S), b \in tc'(S)$.

If we return to the four candidate examples in figure 2, we can see what can and cannot be implemented over these two tournaments. The set of all candidates is not prime, due to the decomposition illustrated in figure 3.

So the only pairs that are implementable over these two tournaments are $(a, a), (b, b), (c, c), (d, d)$ and (a, b). This is the smallest case of a non-prime set.

Srivastava and Trick further conjecture that the condition in the theorem is sufficient to define implementable rules. They conjecture that any rule defined over all tournaments is implementable if and only if it is implementable over every pair of tournaments. We'll denote this conjecture the Pairwise Conjecture.

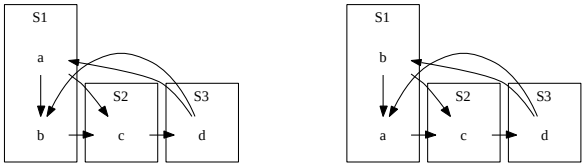


Figure 3: Decomposition

For tournaments with a small number of candidates, this conjecture implies a specific set of implementable rules. For three candidates, there are only two tournaments that do not include Condorcet winners, as shown in figure 4.

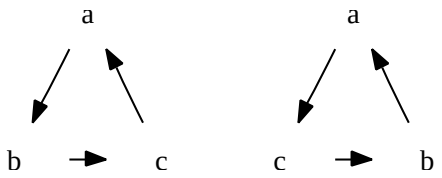


Figure 4: Three candidate tournaments

Since C is prime relative to these two tournaments, the Pairwise Conjecture implies there are exactly 9 implementable Condorcet rules. We will show the trees for these rules in the next section.

For four candidates, the situation is much more complex. We will show that the Pairwise conjecture implies there are exactly $5^{12}3^8 = 1{,}601{,}806{,}640{,}625$ implementable Condorcet rules. While a direct search for these seems beyond current capabilities, we explore aspects of this set of rules in Section 4.

3 Computational Procedure

In this section, we provide a computational approach to generating all implementable rules over a set of tournaments $\mathcal{T}$. If there are m tournaments in $\mathcal{T}$, then we can arbitrarily order that set as $T_1, T_2, \ldots T_m$ and represent a rule by an m-tuple $a_1 a_2 \ldots a_m$ where a_i is the winner for tournament T_i.

We iteratively generate all rules by beginning with the n rules (for $|C| = n$) $jj \ldots j$ for each $j \in C$. Then, given two rules $j_1 j_2 \ldots j_m$ and $k_1 k_2 \ldots k_m$, we can generate a new rule by choosing the winners comparing j_1 and k_1 under T_1, j_2 and k_2 under T_2 and so on. This has the effect of creating a new tree where the j rule is the left branch and the k rule is the right branch.

The procedure may generate the same rule repeatedly, so it is important to identify an already-generated rule quickly. This can be done with a hashing function on the rules. The exact hashing function is unimportant providing the number of trees assigned to any particular hash value is relatively low. In

our implementation, we use a hash function that is a function of the number of times each candidate appears in the rule along with some lesser terms.

We also want to generate the smallest tree for each rule (in terms of number of leaves in the tree). This is done by always combining trees that result in the minimum number of leaves in the combined tree. We begin with 1 leaf in each of the $jj \dots j$ trees. Combining a rule with n_1 leaves with one of n_2 leaves results in a tree of $n_1 + n_2$ leaves. This lets us generate all trees of k leaves by combining the $k-1$ leaf trees with the 1 leaf trees, the $k-2$ leaf trees with the 2 leaf trees and so on. Once all the k leaf trees are generated, the routine can move onto $k+1$ leaf trees.

The final optimization to be done is to identify all isomorphic rules, where one rule is isomorphic to another if a permutation of the candidates applied to one rule's tree results in the other rule as winners. Identifying isomorphic rules allows the presentation of a smaller number of trees. As long as the number of candidates is not large, this can be done by enumeration.

The resulting code is able to generate millions of rules in a few hours. While this is not fast enough for a complete enumeration of the more than 1,000,000,000,000 (conjectured) four candidate rules, it is enough to determine the set of implementable rules over smaller sets of tournaments.

To begin, it is simple to calculate the trees over three candidates. There are exactly three minimum-sized, non-isomorphic onto voting trees on three candidates. These are shown in figure 5.

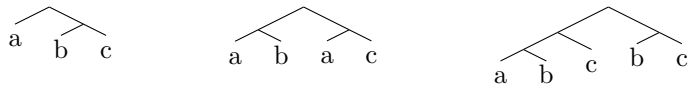


Figure 5: Trees on Three Candidates

Since all the trees contain all three candidates, the rules they implement are Condorcet. If there is no Condorcet winner, the first tree always chooses a, the second tree chooses the loser between b and c, and the third tree chooses the winner between b and c (it is interesting the tree for choosing the winner is larger than the tree choosing the loser). For each tree, there are three relabelings that result in different rules, so these three trees give 9 rules, as required by the Pairwise Conjecture.

These trees show an intriguing sort of agenda manipulation: the choice of tree leads to strong effects on the candidate. In the first tree, neither b nor c can win (unless they are Condorcet winners), but it is obvious that the tree is biased towards a. It is not obvious that the other two trees are biased against a, but in neither case can a win without being the Condorcet winner.

4 Results on Four Candidates

While we cannot generate all rules for all tournaments on four candidates, we can do so for some interesting subsets of tournaments.

The key insight into analyzing four-candidate voting rules is that, for two tournaments T_1 and T_2 over four candidates, if T_1 and T_2 are not isomorphic to the tournaments in figure 2 (by relabeling candidates), then the entire 4-candidate set C is prime. So for pairs not isomorphic to those in figure 2, all pairs of candidates are implementable. A brute force calculation shows that every one of the 24 tournaments with all candidates in the top cycle has exactly one other tournament for which the pair is isomorphic to those in figure 2. As stated before, over a pair like that in figure 2, there are only 5 implementable rules (not $4^2 = 16$), so there are 5^{12} =244,140,625 rules over the those 24 tournaments (assuming the Pairwise conjecture).

In addition to the 24 tournaments where the top cycle contains all four candidates, there are 8 tournaments with three candidates in the top cycle. Pairing each of these with every other tournament results in a prime set consisting of at least the candidates in the top cycle, so for each of the 5^{12} rules on tournaments with 4-candidate top cycles, there are 3^8 choices from the three-candidate top cycles. Since all remaining tournaments have a condorcet winner, this gives a total of $5^{12}3^8$ =1,601,806,640,625 onto rules over all tournaments.

While this number is beyond current capability to handle directly, we are able to analyze different classes of rules. These classes are of independent interest since they give interesting trees in their own right, and they provide indirect confirmation of the Pairwise Conjecture since they give sets where no counterexample exists.

All tournaments with a Condorcet Loser. There are eight such tournaments, and each has three candidates in the top cycle. All pairs are prime over their top cycles, so the Pairwise Conjecture implies there are $3^8 = 6561$ implementable voting rules. The computational procedure shows that is, indeed the case. Table 1 gives the number of rules of each size; figure 6 gives an example of a size 16 voting rule.

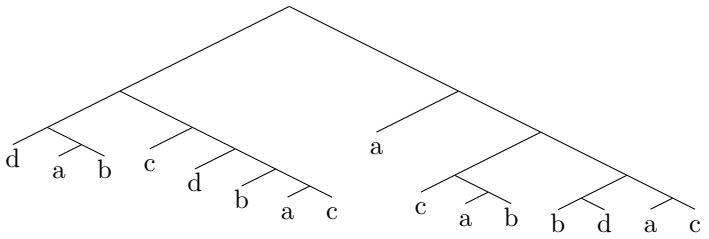


Figure 6: Size 16 Voting Tree

All Tournaments with a Condorcet Loser plus 2. All tournaments on four candidates without a Condorcet winner or loser have the same structure: there

Size	Number	Non-Iso.
4	15	2
5	102	7
6	144	10
7	264	13
8	507	25
9	852	38
10	936	47
11	1152	49
12	1089	49
13	732	31
14	504	22
15	192	8
16	72	5

Table 1: Voting Rules on 4 Candidates - 8 Condorcet Loser Tournaments

are two candidates who beat two others and two that beat one other. Associated with each tournament is another tournament that is identical except for one reversal in the majority tournament. This is illustrated in figure 2. As shown in section 2, just five of the sixteen paired outcomes is implementable on this pair. Taking one such pair of tournaments together with the eight tournaments with a Condorcet loser gives 10 tournaments, over which the Pairwise Conjecture predicts $5(3^8) = 32805$ implementable rules. Again, the computational procedure confirms this number with a maximum tree size of 24. The table is shown in table 2 and a sample tree of size 24 is given in figure 7.

All Voting Rules over 4 candidate, no Condorcet Loser tournaments. The most interesting rules on four candidates involve the tournaments for which all candidates are in the top cycle. As mentioned, the 24 such tournaments break into 12 pairs, and there are five choices of pairs of winners for each pair. The Pairwise Conjecture predicts that this will lead to exactly $5^{12} = 244{,}140{,}625$ implementable rules. It is possible that these rules might be enumerated to provide further evidence for the Pairwise Conjecture.

There are some structured rules for which the tree would have independent interest. To describe these rules, note that for every four candidate tournament with all candidates, there are two candidates who beat two others (so have Copeland score 2), while two candidates beat only one other (Copeland score 1). If we let $w_1(T)$ and $w_2(T)$ be the two candidates with Copeland score 2 under T such that $w_1(T)Tw_2(T)$ and let $l_1(T)$ and $l_2(T)$ be the corresponding candidates with Copeland score 1, $l_1(T)Tl_w(T)$, then if T and T' are decomposable pairs (as shown in figure 2), then

- $w_1(T) = w_1(T')$
- $w_2(T) = l_1(T')$

Size	Number	Non-Iso.
4	15	2
5	102	7
6	169	19
7	345	45
8	693	109
9	1268	207
10	1837	391
11	2715	681
12	3335	951
13	3643	1223
14	3807	1430
15	3500	1489
16	3110	1441
17	2691	1307
18	2348	1173
19	1583	791
20	977	488
21	475	238
22	156	78
23	34	17
24	2	1

Table 2: Voting Rules on 4 Candidates - 8 Condorcet Loser Tournaments plus 2

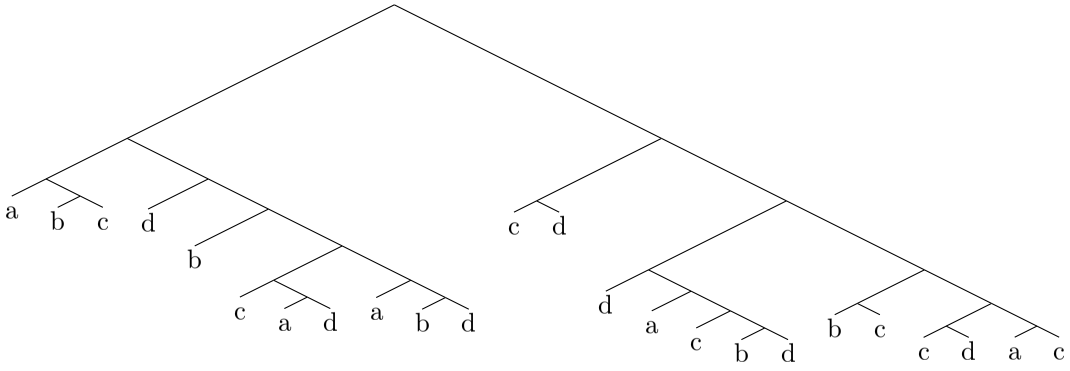


Figure 7: Size 24 Voting Tree

- $l_1(T) = w_2(T')$
- $l_2(T) = l_2(T')$

Using Theorem 1, we then see the only implementable rules on T and T' are $(w_1(T), w_1(T'))$, $(w_2(T), l_1(T'))$, $(l_1(T), w_2(T'))$, $(l_2(T), l_2(T'))$, and $(w_2(T), w_2(T'))$. So, if we look for Copeland winners, the only tiebreaking rules possible are to choose w_1 for both T and T' or w_2 for both T and T': it is not permitted to choose w_1 from one and w_2 for the other. Even stronger, the only possible tie-breaking rule among Copeland losers (in the top cycle) is l_2 for both T and T'.

This implies there are $2^{12} = 4096$ rules that choose Copeland winners, and only one rule that chooses Copeland losers. The structure of such trees would be of independent interest.

Initial runs on this set of tournaments are limited to trees of size 21 or less. The table gives the number of rules and the number of Copeland rules found. At this point, we have found 5,608,475 rules, of which 1746 choose from Copeland winners (the rule that chooses a Copeland Loser has not yet been found). We display a 21 node tree that chooses among Copeland winners.

Size	Number	Copeland
4	15	3
5	102	0
6	424	0
7	1104	0
8	2377	19
9	5486	4
10	11232	18
11	21768	36
12	40420	36
13	70600	96
14	116670	60
15	187560	96
16	294510	240
17	439102	192
18	633986	138
19	895648	292
20	1231551	368
21	1655920	148

Table 3: Rules over 4 Candidates, no Condorcet Losers

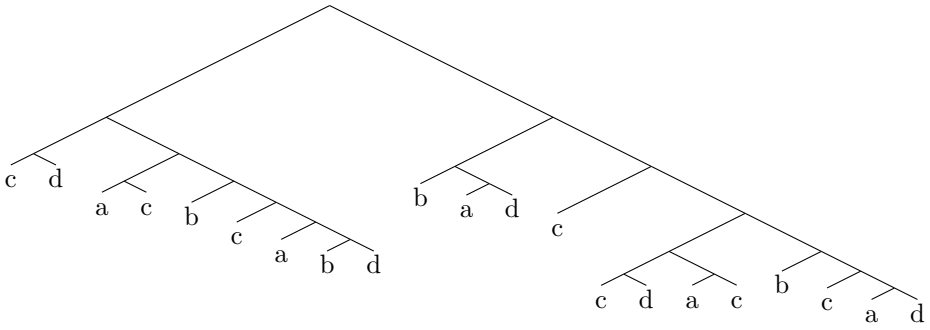


Figure 8: 21 Node Tree Choosing among Copeland Winners

5 Conclusions

There are some conclusions that can be drawn from these tests. First, it is clear that any counter-example to the Pairwise Conjecture will either have to be extremely involved or involve a larger number of candidates than we have considered here. Second, minimal trees implementing rules can be extraordinarily complex, involving the repeated comparison of candidates. Even allowing a candidate to appear six times (on average) in a tree generates only a small fraction of the possible rules on four candidates. It may be possible to use this as a measure of the complexity of rule. This measure would be somewhat counterintuitive, since the smallest trees generate rules that are difficult to describe, while easy to describe rules (like choosing the l_2 candidate for each tournament) seem to generate very complex trees.

Generating all rules over 4 candidate tournaments without a Condorcet Loser (so all candidates are in the top cycle) seems within reach and should lead to insight into possible tie breaking rules among these tournaments.

One limit to this computational approach is the limited use of symmetry-breaking. While we generate many rules and trees, many of them are isomorphic to others, and exploiting this fact may lead to significant computational speedup. Such an improvement is needed if there is any possibility of moving onto five candidates or more.

References

[1] J.S. Banks, "Sophisticated Voting Outcomes and Agenda Control," *Social Choice and Welfare*, **1:** 295–306 (1985).

[2] P.J. Coughlan and M. Le Breton, "A Social Choice Function Implementable via Backward Induction with Values in the Ultimate Uncovered Set," *Review of Economic Design*, **4:** 153–160 (1999).

[3] B. Dutta, "Covering Sets and a New Condorcet Choice Correspondence," *Journal of Economic Theory*, **44:** 63–80 (1988).

[4] B. Dutta and A. Sen, "Implementing Generalized Condorcet Social Choice Functions via Backward Induction," *Social Choice and Welfare*, **10:** 149–160 (1993).

[5] R. Farquharson, *Theory of Voting*, Yale University Press, New Haven (1969).

[6] M. Herrero and S. Srivastava, "Decentralization by Multistage Voting Procedures," *Journal of Economic Theory*, **56:** 182–201 (1992).

[7] R.D. McKelvey and R.G. Niemi, "A multistage game representation of sophisticated voting for binary procedures," *Journal of Economic Theory*, **18:** 1–22 (1978).

[8] N. 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:** 68–96 (1980).

[9] H. Moulin, "Prudence versus Sophistication in Voting Strategy," *Journal of Economic Theory*, **24:** 498–417 (1981).

[10] H. Moulin, "Choosing from a tournament," *Social Choice and Welfare*, **3:** 271–291 (1986).

[11] D. Mueller, "Voting by Veto," *Journal of Public Economics*, **10:** 57–76 (1978).

[12] T. Schwartz, "Cyclic Tournaments and Cooperative Majority Voting: A Solution," *Social Choice and Welfare*, **7:** 19–29 (1990).

[13] S. Srivastava and M.A. Trick, "Sophisticated Voting Rules: The Case of Two Tournaments," *Social Choice and Welfare*, **13:** 275–289 (1996).

Michael Trick
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA, USA
Email: `trick@cmu.edu`