A Non-Parametric Test of Stochastic Preferences

Bart Smeulders, Michel Regenwetter, Frits C.R. Spieksma

Abstract

In this paper we present algorithms to test a theory of stochastic preferences on binary choice data. For the general case, where preferences can be any strict linear order, a column generation algorithm is given. This algorithm can be easily adjusted to test for specific classes of preferences or to account for different kinds of data. We perform preliminary experiments showing the viability of this method. Furthermore, we look at the special case of single-peaked preferences and show that for this class, a test of stochastic preferences can be done in polynomial time by checking the data for a simple condition.

1 Introduction

Throughout the scientific literature there is no shortage of models of choice behaviour. Especially the notion of a rational person, who makes decisions based on his/her personal utility function or preferences, has been very influential in economics. However, it has often been observed that when a person is repeatedly given a choice between two alternatives a and b, the choices made are not always consistent. During some repetitions, the person will choose option a, while at other times choosing b. Obviously, such an observation is at odds with the idea of a rational person choosing according to an unchanging preference relation. One possible way to reconcile these observations with the idea of rationalality is to drop the condition of unchanging preferences. Let us consider the following example.

Jim and Jane often meet for dinner. Each time they meet, Jim chooses two of their three favourite restaurants and lets Jane decide at which of the two they will eat. After a year, Jim notices that when the choices are a pizzeria (A) or a sushi restaurant (B), Jane chooses the pizzeria two out of three times. Likewise, when the choice is between sushi (B) and steak (C), Jane prefers sushi two thirds of the time. Finally, when choosing between pizza (A) and steak (C), Jane picks the steak house two out of three times. An observer might think that because Jane seems to prefer pizza over sushi, sushi over steak and steak over pizza, she is irrational. However, a simple and rational explanation exists for her choices. Before dinner, Jane is either at home, at the library or at the gym with equal probability. From her house, the closest restaurant is A, then B and C is furthest. From the library, B is closest, followed by C and A. Finally, from the gym she can quickly get to C and then A, but B is far away. If she always pick the restaurant closest to her current location, this completely explains her choices. Instead of a single unchanging preference, she has multiple preference orderings, with each having an equal probability of determining her choices at any given time.

Such a model, which explains choices by multiple internally consistent preferences is referred to as a *mixture model* [10, 14, 15]. Notice that this model is equivalent to one that explains choices by multiple persons, each with their own preferences. In this paper, we will look at the situation where the decision maker is given pairs of objects and asked to choose between them. The resulting data are frequencies, showing for each pair of objects i, j, how often object i is chosen over j and vice versa. These frequencies can be interpreted as the probablility that i is chosen over j. The main question is then whether observed data are consistent with a mixture model. In this paper, we will compare two settings. First, we will look at the general case of linear orders. For this case we describe an algorithm which can be used to test rationalizability for small to medium size instances. In a second part, we restrict preferences to so-called single-peaked preferences. In this case, objects are ranked along an axis according to some characteristic. Preference orderings must have the characteristic that one object is the peak and the preference relation over two objects on the same side is determined solely by their relative proximity to the peak object along the axis [2]. In this special case, we can show that the rationalizability question becomes polynomially time solvable, even if the ranking of objects along the axis is unknown a priori.

1.1 Our contributions

We show that column generation is a viable technique for testing whether observed data are consistent with the mixture model. The pricing problem involved amounts to solving an instance of the linear ordering problem for which quite some knowledge is present in the literature; preliminary computations show that instances with up to 20 items can be solved in reasonable computing times. Furthermore, we show that when preferences are restricted to so-called single peaked preferences, the testing procedure simplifies to verifying a relatively simple condition.

1.2 A short review of literature

In the case of general linear orders, the question whether the observed data are consistent with the mixture model is equivalent to the question whether a point representing the data lies within the *linear ordering polytope* (LOP) [16]. This observation is interesting, as it allows the knowledge of the polytope from mathematics and operations research to be used for testing this model of choice behaviour on data. For example, known facet-defining inequalities of the polytope can easily be seen as necessary conditions for the choice model and full facet descriptions provide sufficient conditions. Several papers use this approach and identify necessary and sufficient conditions in this way, see amongst others [11, 8]. However, the studies cited use relatively small datasets, with up to 5 objects. Up to this size, the polytope is described by only two classes of facet-defining inequalities. However, this number rises rapidly to over a thousand classes for eight objects, for a total of more than 480 million facets [7, 12]. For more than eight objects, a full description is not known and thus can not be used to test inclusion of the data point. It is thus clear that using facet-defining inequalities as necessary and sufficient conditions becomes impractical very quickly. A second way of testing whether a given point falls within the LOP is through a vertex description, as every point within the polytope can be described as a convex combination of the polytope's vertices. Again this proves impractical, as the number of vertices rises rapidly with the size of the dataset. Specifically, there are $n!$ vertices for a dataset with n objects, making full descriptions cumbersome for even small numbers of objects.

McGarvey [13] describes a related problem in which only the majority decision is observed and he proves that any pattern of majority decisions is rationalizable by a distribution over individual preferences. The difference with the problem described in this paper is that the exact ratio with which a is preferred over b need not be matched by the distribution over preferences in McGarvey's work. Debord describes a further variant, in which the difference between the number of times a is preferred over b and the number of times b is preferred over a is recorded, but not the number of trials. For this variant, he describes conditions for rationalizability [5]. [1]

1.3 Paper Organization

The rest of this paper is organized as follows. In section 2, we propose a method of testing inclusion of the data point using only a partial vertex description of the LOP. Subsection 2.1 will formally lay out the problem, alongside the notation used in this paper. In Subsection 2.2, we will then describe a column generation algorithm which may be used to solve instances of the problem. Subsection 2.3, is then used to discuss some preliminary computational results. Next, Section 3 will formally introduce single-peaked preferences. In this section, we will then give a condition for rationalizability by single-peaked preferences which can easily be checked in polynomial time. Finally, section 4 concludes.

2 General Linear Orders

In this section, we propose a method of testing inclusion of the data point using only a partial vertex description of the LOP. Obviously, any data point that can be described as a convex combination of a subset of the LOP's vertices is also included in the LOP. If the point falls outside of this portion of the polytope, we will use column generation to identify new vertices to expand the partial LOP in the direction of the data point. All this will be done using a linear program which describes, if possible, the data point as a convex combination of vertices in the partial description and a pricing problem, the linear ordering problem, which is used to generate additional vertices. We note that this general scheme may be used in many similar situations, both when preference orders are restricted to specific classes and situations where the data differs from choices between two objects.

2.1 Problem Description

We consider a set N, consisting of n objects. We further consider strict linear orders over these objects. In an individual strict linear order, objects are ranked as follows. For each pair of objects, $a, b \in N$, we have either $a \succ b$ or $b \succ a$. This relation is transitive, i.e. if $a \succ b$ and $b \succ c$, it is also the case that $a \succ c$. An individual strict linear order is denoted by $\succ_m$. The set of all possible strict linear orders is O, we note that $|O| = n!$. We furthermore consider subsets O_{ij} for each pair of objects $i, j \in N$, with $\succ_m \in O_{ij}$ if and only if $i \succ_m j$. Notice that $O_{ij} \cup O_{ji} = O$ and $O_{ij} \cap O_{ji} = \emptyset$. The observed data is gathered from forced binary choices and represented by p_{ij}, the probability that i is chosen over j. As in each situation, either i or j must be chosen, $p_{ij} + p_{ji} = 1$. We are now in a position to formally state the conditions for rationalizability of the observed data p_{ij} by a mixture model [8].

Definition 1. *Observed data can be rationalized by a model of stochastic strict linear ordering preferences if and only if there exist numbers $x_m \geq 0, m = 1, \ldots, n!$ for which.*

$$\sum_{m:\succ_m \in O_{ij}} x_m = p_{ij}, \qquad \forall i, j \in N \tag{1}$$

A test of rationalizability is thus to check whether a solution exists to this system of equalities (1). A straightforward computation of this system is, as mentioned in the introduction, difficult due to the number of preferences. It corresponds to using the vertex description of the LOP to test inclusion of the observed data in the polytope.

[1] See Charon and Hudry [3] for these results in English

2.2 Column Generation Scheme

In this section, we will lay out a linear programming formulation of the problem and show that it is equivalent to the system (1). This formulation uses an exponential number of variables. It is well-known from LP-theory [4] that it is not necessary to consider all of these variables simultaneously, in particular when the number of constraints is relatively small. Indeed, column generation can be used to solve this LP.

2.2.1 Linear Programming Problem

In the previous section, we provided a system of equalities, which are both necessary and sufficient for the data to be rationalizable by a model of stochastic strict linear ordering preferences. This system can be easily rewritten as an linear programming problem as follows.

$$\text{Minimize} \quad \sum_{m:\succ_m \in O} x_m \tag{2}$$

$$\sum_{m:\succ_m \in O_{ij}} x_m \geq p_{ij} \qquad \forall i,j \in N \tag{3}$$

$$x_m \geq 0 \qquad \forall m = 1,\ldots,n! \tag{4}$$

Claim 1. *The optimal solution value of (2)-(4) is equal to 1 if and only if a solution exists to the system of equalities (1).*

Proof. The claim can be easily checked as follows. First, we show that no feasible solution to (2)-(4) can have a solution value lower than 1. Next, we show that if there exists a solution with this value, it is also a solution to (1). Finally, we show that any solution to (1) is also a solution to (2)-(4), with value 1.

Notice that $O = O_{ij} \cup O_{ji}$ and $O_{ij} \cap O_{ji} = \emptyset$, which implies $\sum_{m:\succ_m \in O} x_m = \sum_{m:\succ_m \in O_{ij}} x_m + \sum_{m:\succ_m \in O_{ji}} x_m$, so for any solution satisfying 3, we have $obj = \sum_{m:\succ_m \in O} x_m \geq p_{ij} + p_{ji} = 1$. Now suppose we have a solution to (2)-(4) with solution value 1. As $1 = \sum_{m:\succ_m \in O} x_m = \sum_{m:\succ_m \in O_{ij}} x_m + \sum_{m:\succ_m \in O_{ji}} x_m$ and $\sum_{m:\succ_m \in O_{ij}} x_m \geq p_{ij}$ and $\sum_{m:\succ_m \in O_{ji}} x_m \geq p_{ji}$, we must have $\sum_{m:\succ_m \in O_{ij}} x_m = p_{ij}$ and $\sum_{m:\succ_m \in O_{ji}} x_m = p_{ji}$. The values of x_m are thus a solution to (1). Finally, suppose we have a solution to the system of equalities(1). The values of x_m can then be put into the linear programming problem. As for each pair of i,j, we have $\sum_{m:\succ_m \in O_{ij}} x_m = p_{ij}$ it is clear that the constraints are met. Furthermore, as $p_{ij} + p_{ji} = 1$, we have $1 = \sum_{m:\succ_m \in O_{ij}} x_m + \sum_{m:\succ_m \in O_{ji}} x_m = \sum_{m:\succ_m \in O} x_m$ and thus we have a feasible solution to the LP with solution value 1, thus an optimal solution. □

Clearly this LP-formulation still has a large number of variables ($n!$), as each variable represents one vertex in the vertex description of the LOP. However, this formulation has a relatively small number of constraints (n^2). Since an optimal solution to an LP can be found with a number of non-zero variables less than or equal to the number of constraints, it is clear that not all variables are needed. We will therefore use a column generation approach. In this context, we will refer to the linear problem (2)-(4) as the primal or master problem. As a starting point, the master problem is solved with only a subset of variables, if these are sufficient to find a solution of value 1, we have proven rationalizability. If this is not the case, we will use a pricing problem described below, which allows us to find additional variables which may improve the solution. We use this process until we either find a solution with value 1, or we conclude that the given data-point is not rationalizable.

2.2.2 Column Generation

The dual of the above LP-formulation is as follows.

$$\text{Maximize} \quad \sum_{i,j \in N} p_{ij} y_{ij} \tag{5}$$

$$\sum_{ij:\succ_m \in O_{ij}} y_{ij} \leq 1 \qquad \forall k = 1, \ldots, n! \tag{6}$$

$$y_{ij} \geq 0 \qquad \forall i,j \in N \tag{7}$$

Given a solution $x = \{x_1, x_2, \ldots, x_m\}$ to the primal problem, a solution $y = \{y_1, y_2, \ldots, y_{n^2}\}$ can easily be found. It is well-known that feasibility of the dual solution y is equivalent to optimality of the primal solution x. Thus, if we want to test optimality of x, we may test whether the dual solution y satisfies (6) and (7). This is done by solving a pricing problem. In this case, there exists a violated inequality if and only if there exists a linear order $\succ_m$, for which $\sum_{ij:\succ_m \in O_{ij}} y_{ij} > 1$. This gives us the following pricing problem.

$$\text{Maximize} \quad \sum_{i,j \in N} y_{ij} \times a_{ij} \tag{8}$$

$$a_{ij} + a_{ji} = 1 \qquad \forall i,j \in N, i \neq j \tag{9}$$

$$a_{ij} + a_{jk} + a_{ki} \leq 2 \qquad \forall i,j,k \in N, i \neq j \neq k \neq i \tag{10}$$

$$a_{ij} \in \{0,1\} \qquad \forall i,j \in N, i \neq j \tag{11}$$

A solution of the problem consists of the a_{ij} variables, which given the constraints (9) - (11) encode a strict linear order. Any such solution for which the objective value (8) is greater than 1, corresponds to a variable which when added can improve the solution of the master problem. We notice that the pricing problem is the well known linear ordering problem. This problem is known to be NP-Hard (Garey and Johnson [9]). However, it is not necessary to solve the pricing problem using exact algorithms, as long as violated constraints can be found using heuristic solution methods.

2.3 Implementation

To illustrate some characteristics of the problem, we provide preliminary computational results for a number of instances with 20 objects. These instances are split into 2 classes, for the first class, all p_{ij} values were chosen completely at random. Due to the small size of the LOP in comparison to the unit hypercube of which it is a part, all instances of this class fall outside of the LOP. For the second class, a number of preference orderings were randomly generated and randomly assigned weights which sum to 1. In this way, a convex combination of these vertices is found, providing a point within the LOP. Table 1 provides an overview of computation times needed and number of vertices generated by the column generation algorithm to answer the rationalizability question. Instances of the first class are denoted by "R", those of the second class by the number of preference generated from which the convex combination was taken.

All computation times were obtained on a dual core 2.5 GHz computer with 4 GB RAM. Starting columns for the linear programming problem were obtained in a heuristic fashion. Both the linear programming and pricing problems were solved using CPLEX 12.4, called from a custom built program. Average computation times in Table 1 show large variations between different sets of instances. Showing that instances are not rationalizable, as is done for the first class, can be done very quickly. On the other hand, proving that observed data

Instance	# Instances	Av. Comp. Time (sec)	# vertices
R	10	6.2	77
5	5	1249	1409
10	5	63	400
20	5	32	248

Table 1: Summary of computational results

is rationalizable takes longer. Of interest are the large differences in computation time for seemingly similar instances. These differences can be explained by the location of the data point within the LOP. If generated using a larger number of vertices, the resulting point is likely to be closer to the centre of the LOP, thus describable as a convex combination by far more combinations of vertices. The instances generated by a smaller number of preference orderings are closer to the convex hull and thus need more specific sets of vertices. This can be seen in that far more vertices need to be generated and used in the LP formulation to prove inclusion in the LOP.

We again note that results as described in Table 1 are very preliminary. Speed-ups are possible for many parts of the algorithm. Especially the pricing problem is a target of further research, a large literature exists on algorithms for the linear ordering problem [12]. Faster exact algorithms can be used to drive down the time needed to find new vertices. Additionally, it is not necessary to find the optimal solution to the pricing problem, only a solution which corresponds to a violated dual constraint. This allows usage of heuristics to quickly find new vertices without utilizing less performant exact algorithms.

3 Single-Peaked Preferences

In this section, we will look at single-peaked preferences, which are of considerable interest to the social choice community. They provide a very natural restriction to preferences if alternatives can be ordered on a line according to some property. The structure imposed by these preferences also have important theoretical consequences, for example, single-peaked preferences avoid Condorcet's paradox [2]. Related to the rationalizability question in this paper, Bartholdi and Trick [1] provide a polynomial time test of consistency of given (full) preference orderings with single-peaked preferences, which was improved upon by Escoffier et al. [6]

Formally, we distinguish single-peaked preferences from the general linear orders as follows. Each object $i \in N$ is given a position along an axis, $pos(i)$, with $pos(i) \in \{1, 2, \ldots, n\}$ and if $i \neq j$, $pos(i) \neq pos(j)$. Consider some ordering $\succ_m$ with a most preferred item a, i.e., $a \succ_m i$ for each $i \in N \setminus a$. For $\succ_m$ to be a single-peaked preference, it must hold that for each pair of objects $i, j \in N$, if $pos(i) < pos(j) < pos(a)$ or $pos(i) > pos(j) > pos(a)$, then it must be the case that $j \succ_m i$. The set of single-peaked, preference orderings given these positions is denoted by $O^{sp}(pos)$; notice that, for reasons of convenience, we will write O^{sp} for short. Notice also that O^{sp} contains an exponential number of orderings. The rationalizability question is now as follows.

Definition 2. *Observed data can be rationalized by a model of stochastic strict single peaked linear ordering preferences if and only if there exist positions $pos(i), \forall i \in N$ and numbers*

$x_m \geq 0, \forall \succ_m \in O^{sp}$ *for which:*

$$\sum_{m:\succ_m \in O_{ij}^{sp}} x_m = p_{ij}, \qquad \forall i,j \in N \tag{12}$$

We will now claim that the existence of a solution to this system of equalities (with an exponential number of variables), is equivalent to the p_{ij} values satisfying a certain condition. We will show this is the case by providing a constructive algorithm, which shows that if the condition on p_{ij} is met, values for x_m which satisfy (1) exist and vice versa. Finally, we will provide a proof that given the values of p_{ij}, the condition can be checked in polynomial time.

Theorem 1. *Observed data can be rationalized by a model of stochastic strict single-peaked linear ordering preferences if and only if there exist $pos(i)$ for all objects i in N, such that for each triple of pairwise distinct objects i, j, k in N*

$$\text{If } pos(i) > pos(j) > pos(k) \text{ or } pos(i) < pos(j) < pos(k) \text{ then, } p_{ij} \leq p_{ik} \tag{13}$$

The proof for the above theorem will be split in several parts. We will first describe an algorithm. This algorithm utilizes variables $\tilde{p}_{ij}$, which are initially equal to p_{ij}. The main part of the algorithms is a loop that outputs single-peaked linear orders. For this loop we will prove three properties, which all depend on the condition (13) being satisfied for $\tilde{p}_{ij}$ for all $i, j \in N$. First, that the loop can always run to completion, i.e. it finds a strict single-peaked linear order $\succ_m$. Second, that this $\succ_m$ can be given a weight x_m, and for all $i, j \in N$ for which $i \succ_m j$, we have $x_m \leq \tilde{p}_{ij}$ and that there exist some $i, j \in N$ for which $i \succ_m j$ and $x_m = \tilde{p}_{ij}$. Finally, that throughout the algorithm, the values of $\tilde{p}_{ij}$ satisfy condition (13). Given these three conditions, we will be able to prove that the algorithm provides values x_m such that they satisfy the system (12).

Algorithm 1 Finding Single-Peaked Preferences

INPUT: p_{ij} for all $i, j \in N$ and $pos(i)$ for all $i \in N$.
Set $\tilde{p}_{ij} := p_{ij}$, $m := 1$ and create $\succ_m := \emptyset$, $M := \emptyset$ and $I := \emptyset$.
while $\tilde{p}_{ij} + \tilde{p}_{ji} > 0$ **do**
  for $|M| < n$ **do**
    Set $I := \{i : \tilde{p}_{ij} > 0, \forall j \in N \backslash M\}$
    If $I = \emptyset$, STOP.
    Set $i^* := \arg\min_{i \in I} pos(i)$.
    Add i^* to M and to the order in $\succ_m$ in last position (i.e. $\forall j \in M, j \succ_m i^*$)
  end for
  Set $x_m := \min_{i,j \in N} \tilde{p}_{ij}$ for which $i \succ_m j$.
  Set $\tilde{p}_{ij} := \tilde{p}_{ij} - x_m, \forall i, j \in N$ for which $\succ_m \in O_{ij}^{sp}$.
  Set $m := m + 1$
end while
OUTPUT: For all $i \in \{1, \ldots, m\}$ a value x_i and order $\succ_i$.

Claim 2. *If the values $\tilde{p}_{ij}$ meet condition (13), the loop will run to completion.*

Proof. If at any point there does not exist an i such that $\tilde{p}_{ij} > 0, \forall j \in N \backslash M$, the algorithm will halt. Suppose this is the case, then for $\min_i pos(i), i \in N \backslash M$, there is some j for which $pos(i) < pos(j)$ and $\tilde{p}_{ij} = 0$. Now let i' be the immediate neighbour of i ($i' = min_{i'} pos(i'), i' \in N \backslash M \cup \{i\}$). Then by condition (13), we have $\tilde{p}_{ii'} = 0$. Again by the same condition, this implies $\tilde{p}_{i'k} > 0, \forall k \in N \backslash M$ for which $pos(k) < pos(i')$. Furthermore, for i', there also exists some $j \in N \backslash M$ for which $\tilde{p}_{i'j} = 0$, this j must have $pos(j) > pos(i')$. By the same argument as for i, we can see that $\tilde{p}_{i'i''} = 0$ and so on until we reach the final object n, which has $p_{nj} > 0, \forall j \in N \backslash M$, a contradiction. In each step of the for loop there must exist an object which can be added to M and the algorithm finds a strict linear order. □

Claim 3. *x_m is such that $x_m \leq \tilde{p}_{ij}$ for all $i, j \in N$ for which $\succ_m \in O_{ij}^{sp}$ and that there exist some $i, j \in N$ for which $\succ_m \in O_{ij}^{sp}$, such that $x_m = \tilde{p}_{ij}$ and $x_m > 0$.*

Proof. This is true by construction, an object i is only added to M if $\forall j \in N \backslash M, \tilde{p}_{ij} > 0$. As $i \succ_m j$ is only the case if j was added to M after i, then all $\tilde{p}_{ij}$ over which the minimization are done are strictly positive. By nature of the minimization, there is also at least one $\tilde{p}_{ij}$ to which x_m is equal and x_m is no larger than any of the $\tilde{p}_{ij}$. □

Claim 4. *If condition (13) is met, the $\tilde{p}_{ij}$ values will satisfy condition (13) throughout the algorithm.*

Proof. First, let us consider the situation $pos(i) < pos(j) < pos(k)$, which implies $\tilde{p}_{ij} \leq \tilde{p}_{ik}$. Only if an order $j \succ_m i \succ_m k$ is found, can the algorithm decrease $\tilde{p}_{ik}$ but not $\tilde{p}_{ij}$. $j \succ_m i$ implies that there exists $l \in N$, such that $\tilde{p}_{il} = 0$ and $\tilde{p}_{jl} > 0$. We will consider three separate situations. First, $pos(l) < pos(i)$, then $pos(l) \geq pos(j)$ and finally $pos(i) < pos(l) < pos(j)$. Let us consider $pos(l) < pos(i)$. Then $\tilde{p}_{li} = y$, which implies $\tilde{p}_{lj} = y$ and $\tilde{p}_{jl} = 0$ as $pos(l) < pos(i) < pos(j)$ gives $\tilde{p}_{li} \leq \tilde{p}_{lj}$. Therefore, $pos(l)$ can not smaller than $pos(i)$, as it would also prevent j from being added to the order. In the case of $pos(l) \geq pos(j)$, it is clear that because $\tilde{p}_{il} = 0$, we must also have $\tilde{p}_{ij} = 0$. Finally, if $pos(i) < pos(l) < pos(j)$, we must have $j \succ_m l$, if this were not the case i could be added to M after l but before j. By the earlier arguments in the paragraph $j \succ_m l$ while $pos(l) < pos(j)$ is only possible if there is some other object $l' \in N$, with $pos(l) < pos(l')$ and $\tilde{p}_{ll'} = 0$. $pos(j) < pos(l')$ gives $\tilde{p}_{lj} = 0$ and therefore $\tilde{p}_{ij} = 0$. If on the other hand $pos(l) < pos(l') < pos(j)$, we can repeat the same argument until we find some l'' with $pos(i) < pos(l'') < pos(j)$ and $\tilde{p}_{jl''} = y$, implying $\tilde{p}_{ji} = y$ and $\tilde{p}_{ij} = 0$. In conclusion, if $pos(i) < pos(j) < pos(k)$, we can only have $j \succ_m i \succ_m k$ if $\tilde{p}_{ij} = 0$. If this is the case, than $\tilde{p}_{ij} \leq \tilde{p}_{ik}$ is satisfied, as $\tilde{p}_{ik} \geq 0$.

The second situation is $pos(i) > pos(j) > pos(k)$, in which case we also have $\tilde{p}_{ij} \leq \tilde{p}_{ik}$. Here, only an order with $j \succ_m i \succ_m k$ can lead to the condition being violated after the algorithm. In the previous paragraph, we established that if $pos(i) < pos(j)$ and the algorithm places $j \succ_m i$, we have $\tilde{p}_{ij} = 0$. Here, $pos(k) < pos(i)$ and $i \succ_m k$, so $\tilde{p}_{ki} = 0$. As $\tilde{p}_{ji} + \tilde{p}_{ij} = \tilde{p}_{ik} + \tilde{p}_{ki}$, we have $\tilde{p}_{ji} + \tilde{p}_{ij} = \tilde{p}_{ik}$ and $\tilde{p}_{ij} \leq \tilde{p}_{ik}$.

□

We are now in a position to prove theorem 1.

Proof. We have shown, by claim 2, that given a set of values $\tilde{p}_{ij}$ which satisfy condition (13), we can find a strict single-peaked linear order. By claim 3 we have also seen that we can attach a weight to this order which is non-negative. Even stronger, we have shown that this weight is equal or less than the value $\tilde{p}_{ij}$ for some $i, j \in N$, for which $x_m \in O_{ij}^{sp}$. As the final step of the loop will decrease these $\tilde{p}_{ij}$ values, at least one of these values is set to zero in each run. After at most $O(n^2)$ iterations of the loop, each value $\tilde{p}_{ij}$ will then be zero. It can be easily checked that at this point, the values x_m form a solution to (12). As this proof requires the loop to be run multiple times, and the loop requires condition (13) to hold, claim 4 is crucial, as it shows that if the input of the loop exhibits the necessary characteristic, the output will as well.

This establishes the sufficiency of the condition for rationalizability. The necessity can easily be verified by a three object example. Suppose a, b, c, with $pos(a) < pos(b) < pos(c)$ and $p_{ab} > p_{ac}$. By definition of single-peaked linear orders, each order for which $a \succ b$ it is also the case $a \succ c$. This means $O^{sp}_{ab} \subset O^{sp}_{ac}$ and $\sum_{k:\succ_m \in O^{sp}_{ab}} x_m \leq \sum_{k:\succ_m \in O^{sp}_{ac}} x_m$. □

By the previous arguments, we have shown that the system of equalities 12 and the condition 13 are equivalent. We now claim this condition can be checked in polynomial time. In the proof, we will exclude one very specific case, in which there is some subset of objects N', such that $p_{ij} = p_{ik}$ for every i in $N \backslash N'$ and all j, k in N'. In other words, if all objects in the subset are identical when compared to objects outside of the subset. It can be verified that if such a subset is encountered, Algorithm 2 can be run for this subset to establish relative positions within the subset. These relative positions can then be used to assign positions within the complete set.

Theorem 2. *Algorithm 2 can be used to check whether there exists positions for which Condition 13 holds in polynomial time.*

Proof. Suppose that for a given object i, positions exist, such that $pos(i) = 1$ and for which the condition (13) is satisfied. We claim that Algorithm 2 returns these positions. First, given that $pos(i) = 1$, $p_{ij} < p_{ik}$ implies $pos(j) < pos(k)$, by construction, the algorithm chooses the only option that satisfies this property, because of line 6. In case of ties and multiple positions satisfying the property for i, j, k, lines 7-14 ensures that the positions are chosen so as to be the only option that satisfies the conditions of some other triple of objects. In this way, every object that is given a position is placed in the only feasible position. Thus, if there exists an ordering that satisfies condition 13 with $pos(i) = 1$, it will be found. As the algorithm runs through every object as a starting object, it is ensured that an ordering is found with some starting object, it it exists. It can easily be checked that this algorithm runs in polynomial time. □

4 Conclusion

In this paper, we have described ways of testing a stochastic theory of choice behaviour. A broadly applicable algorithm was presented which, given proper adjustments, can be used for tests with different kinds of data or preference classes. Computational results show that this algorithm can be used to test datasets of sizes which can not be tackled using current techniques. We also show that computationally, much work can still be done given the large literature of heuristics and exact methods for solving linear ordering problems which can be employed in our column generation approach. Furthermore we have provide a polynomial time test for a special case of preferences which is widely used in social choice theory by exploiting the structure of this special class.

Algorithm 2 Ordering Algorithm

Set $i = 1$.
for $i \leq n$ **do**
    Set $R := N$, $C := L := \emptyset$, $a := 1$.
    Set $pos(i) := 1$ and $R := R \backslash i, L := L \cup i$, $a = a + 1$.
    for $R \neq \emptyset$ **do**
        $\forall j \in R$ for which $\min_{k \in R} p_{ik} = p_{ij}$, set $C := C \cup j, R := R \backslash j$.
        for $|C| > 1$ **do**
            if $\exists j, k \in C$ and $l \in L$, for which $p_{lj} < p_{lk}$ **then**
                Set $C := C \backslash k$ and $R := L \cup k$.
            end if
            if $\exists j, k \in C$ and $l \in R$, for which $p_{lj} > p_{lk}$ **then**
                Set $C := C \backslash k$ and $R := L \cup k$.
            end if
        end for
        For $j \in C$, set $pos(j) := a$, $L := L \cup j$, $C := C \backslash j$, $a := a + 1$.
    end for
    Given the resulting positions, check condition 13, if satisfied, STOP, output YES-INSTANCE.
    Set i = i + 1.
end for
Output NO-INSTANCE

References

[1] J. Bartholdi III and M. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165–169, 1986.

[2] D. Black. On the rationale of group decision-making. *The Journal of Political Economy*, 56(1):23, 1948.

[3] I. Charon and O. Hudry. A survey on the linear ordering problem for weighted or unweighted tournaments. *4OR*, 5(1):5–60, 2007.

[4] V. Chvátal. *Linear programming.* WH Freeman and Company, New York, 1983.

[5] B. Debord. Caractérisation des matrices des préférences nettes et méthodes d'agrégation associées. *Mathématiques et Sciences humaines*, 25(97):5–17, 1987.

[6] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *ECAI*, volume 8, pages 366–370, 2008.

[7] S. Fiorini. Determining the automorphism group of the linear ordering polytope. *Discrete applied mathematics*, 112(1):121–128, 2001.

[8] P. Fishburn and J. Falmagne. Binary choice probabilities and rankings. *Economics Letters*, 31(2):113–117, 1989.

[9] M.R. Garey and D.S. Johnson. *Computers and intractability*, volume 174. Freeman San Francisco, CA, 1979.

[10] G. Loomes and R. Sugden. Incorporating a stochastic element into decision theories. *European Economic Review*, 39(3):641–648, 1995.

[11] J. Marschak. Binary-choice constraints and random utility indicators. In *Proceedings of a Symposium on Mathematical Methods in the Social Sciences*, 1960.

[12] R. Martí and G. Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, volume 175 of *Applied Mathematical Sciences*. Springer-Verlag Berlin Heidelberg, 2011.

[13] David C McGarvey. A theorem on the construction of voting paradoxes. *Econometrica: Journal of the Econometric Society*, 21:608–610, 1953.

[14] M. Regenwetter, J. Dana, and C.P. Davis-Stober. Testing transitivity of preferences on two-alternative forced choice data. *Frontiers in psychology*, 1(148):1–15, 2010. doi: 10.3389/fpsyg.2010.00148.

[15] M. Regenwetter, J. Dana, and C.P. Davis-Stober. Transitivity of preferences. *Psychological Review*, 118(1):42, 2011.

[16] R. Suck. Geometric and combinatorial properties of the polytope of binary choice probabilities. *Mathematical Social Sciences*, 23(1):81–102, 1992.

Bart Smeulders
Faculty of Economics and Business - ORSTAT
KU Leuven
Leuven, Belgium
Email: `bart.smeulders@kuleuven.be`

Michel Regenwetter
Psychology Department - Quantitative Division
University of Illinois
Urbana-Champaign, USA
Email: `regenwet@illinois.edu`

Frits C.R. Spieksma
Faculty of Economics and Business - ORSTAT
KU Leuven
Leuven, Belgium
Email: `frits.spieksma@kuleuven.be`