First-order dominance: stronger characterization and a bivariate checking algorithm

Troels Martin Range, Lars Peter Østerdal

Published in

Mathematical Programming

2017

https://doi.org/10.1007/s10107-017-1213-9

Noname manuscript No.
(will be inserted by the editor)

First order dominance: stronger characterization and a bivariate checking algorithm

Troels Martin Range · Lars Peter Østerdal

July 30, 2017

Abstract How to determine if a distribution is superior to - i.e. first order dominates - another is a fundamental problem with many applications in economics, finance, probability theory and statistics. Nevertheless, little is known about how to efficiently check first order dominance for finite multivariate distributions. Utilizing that this problem can be formulated as a transportation problem having a special structure we provide a stronger characterization of multivariate first order dominance and develop a linear time complexity checking algorithm for the bivariate case. We illustrate the use of the checking algorithm when numerically assessing first order dominance between continuous bivariate distributions.

Keywords Multivariate first order dominance · usual stochastic order · characterization · network problem · checking algorithm

Mathematics Subject Classification (2000) 60E15 · 90C08 · 91B82

Troels Martin Range (Corresponding author)
Hospital of South West Jutland and Institute of Regional Health Research: Centre of South West Jutland, University of Southern Denmark, Finsensgade 35, DK-6700 Esbjerg, Denmark
E-mail: Troels.Martin.Range@rsyd.dk, Tel.: +45 2012 8140, and Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Alfred Getz veg 3, NO-7491 Trondheim, Norway.

Lars Peter Østerdal
Department of Economics, Copenhagen Business School, Porcelænshaven 16A, 1. floor, DK-2000 Frederiksberg, E-mail: lpo.eco@cbs.dk,Tel.: +45 3815 5641

1 Introduction

The theory of stochastic dominance is a methodological cornerstone in economics, finance, probability theory, and statistics, among other fields. In welfare economics, for instance, dominance concepts are used for partial rankings of population distributions according to better social welfare or less inequality (e.g. Atkinson and Bourguignon [3], Gravel and Moyes [13]), in decision theory and finance, stochastic orderings are used for evaluating risky assets (e.g. Sriboonchita et al. [39]), and in statistics, appropriate order constraints can support inference (e.g. Silvapulle and Sen [38]). For comprehensive reviews of stochastic dominance theory we refer to Marshall and Olkin [24], Müller and Stoyan [29], and Shaked and Shanthikumar [37].

The canonical stochastic dominance concept is that of *first order dominance*, also known simply as *dominance* or the *usual (stochastic) order* (e.g. Grant et al. [12], Levhari et al. [22], and Lehmann [21]).[1] Intuitively, first order dominance means that one (dominant) distribution is superior to, i.e. gives unambiguously higher outcomes than, the other (dominated) distribution. For two multivariate finite probability mass functions, f and g, f first order dominates g if and only if one of the following three (equivalent) conditions hold: (a) it is possible to obtain g from f by moving probability mass from better to worse outcomes, (b) the cumulative probability mass at f is smaller than or equal to that at g for every lower comprehensive subset of outcomes[2], and (c) the expected utility of f is at least as high as that of g for any non-decreasing utility function.[3] Thus, first order dominance is an ordinal concept that does not rely on assumptions about the relative importance

[1] Less restrictive dominance criteria for better distributions have been defined by imposing stronger restrictions on the set of admissible utility functions. See, e.g., Levy and Paroush [23], Harder and Russell [15], Huang et al. [16], Atkinson and Bourguignon [3], Mosler [27], Russell and Seo [33], Scarsini [35], and Meyer and Strulovici [25].

[2] A lower comprehensive subset holds the property that if an outcome is in the subset, then all smaller outcomes are also included in that subset.

[3] The equivalence between (b) and (c) was proven by Lehmann [21] and Levhari et al. [22]. The equivalence between (a) and (b) can be obtained as a corollary of a theorem by Strassen [40] (see e.g. Kamae et al. [18]) or can be established through an application of the *max-flow min-cut theorem* for flow networks (see e.g. Preston [32]). Østerdal [41] gives a direct proof for equivalence between (a) and (b) in the finite case.

of dimensions or the complementarity/substitutability relationships between dimensions (Arndt et al. [1]).[4]

For one-dimensional distributions, much research into the nature of first order dominance has been conducted (for example, the bibliography by Bawa [4] contains more than 400 references), and the theory is by now well developed and the applications many. Surprisingly, perhaps, little is known about how to efficiently check first order dominance in two or more dimensions. The naive way to check first order dominance is to use definition (b) directly. Here one needs to check an inequality for each lower comprehensive subset of outcomes. However, the number of inequalities to be tested grows dramatically in the total number of outcomes, so it is not an efficient method.[5] Mosler and Scarsini [28] and Dyckerhoff and Mosler [10] describe a method based on linear programming for checking first order dominance in the general multivariate finite case, based on definition (a) above. Empirical implementations of a method along these lines appear in Arndt et al. [1] and Arndt et al. [2]. An alternative approach would be to make use of a network flow formulation of the problem, as outlined in Preston [32] or Hansel and Troallic [14], and then check for dominance via computation of the maximum flow. We are not aware of any implementations of such a method for checking multivariate first order dominance. Focusing on the bivariate case, we will argue below that none of these existing methods are efficient.

It is not difficult to see that the problem of checking first order dominance for finite multidimensional distributions can be formulated as a transportation problem. In this paper, we argue that this transportation problem has a special structure and allows a formulation which makes it possible to identify an upper bound on the number of diminishing transfers necessary to reach one (dominated) distribution from another (dominant) distribution under first order dominance. In this way we provide a stronger characterization of first order dominance. Furthermore, we utilize this insight to obtain a linear time complexity algorithm for the bivariate case. To our knowledge, it is faster than

[4] In the multivariate context the first order dominance concept has been used with other meanings than the one given here. In particular, Atkinson and Bourguignon [3] and others have used the term "first order dominance" to denote a less restrictive and easier to check dominance concept for better distributions (also known as an orthant stochastic order cf. e.g. Dyckerhoff and Mosler [10]) suitable under a substitutability relationship between the dimensions.

[5] Dykstra and Robertson [11] shows that for the bivariate case, with n_1 elements in the first dimension and n_2 elements in the second dimension, the number of possible lower comprehensive sets increases by $\binom{n_1+n_2}{n_1}$. In effect the number increases exponentially when both dimensions grow. Sampson and Whitaker [34] extends this to the multivariate case.

any other algorithm to determine bivariate first order dominance which has been identified in the literature. As a stepping stone to arrive at the $O(n)$ algorithm we establish an intuitive and constructive $O(n^2)$ algorithm for which we prove that a certain predefined sequence of diminishing transfers will always exist if one distribution first order dominates another. We observe that it is not necessary to obtain the explicit diminishing transfers between outcomes but rather diminishing transfers between certain sets of outcomes to prove first order dominance. By exploiting this observation we derive the $O(n)$ algorithm.

Østerdal [41] provides a constructive proof of the relation (a) and (b) and, as a consequence of the proof, introduces an algorithm for identifying whether or not one distribution first order dominates another distribution. The method given relies on having an initial set of diminishing transfers and then searches for potential cycles in which non-diminishing transfers are present and can be exchanged with diminishing cycles. While it is argued that the method works, the time complexity is not considered. Indeed the search for cycles constitutes an $O(n^2)$ algorithm in the verbal description of the method, and this search is executed for each potential outcome yielding an $O(n^3)$ algorithm. The algorithms we present in this paper for the bivariate case is significantly faster. Furthermore, Østerdal [41] proves the existence of a finite sequence of diminishing transfers without considering the number of such transfers necessary for proving first order dominance. In contrast, we use the transportation formulation to provide an upper bound on the number of probability mass transfers necessary to show first order dominance. Hence we strengthen the characterization of the equivalence (a) and (b).

Efficient methods for checking multivariate first order dominance are important. In applied work, for example, observations of discrete distributions with thousands or even millions of levels along each dimension can be available, in which case efficiency of the test procedure becomes critical. When comparing continuous bivariate distributions, an approximate first order dominance check can be performed by checking first order dominance between corresponding discretized distributions. The more fine-grained this discretization is, the better approximation. Clearly, more efficient methods allows for more fine-grained discretizations, and thus more accurate dominance checks.[6]

[6] Bootstrapping procedures have also been used to analyze the robustness of first order dominance relations with respect to sample variation in the original data set (e.g. [1], [2], [17]). These use random sampling with replacement requiring many first order dominance comparisons for any pair of original distributions.

The paper is organized as follows. First, in section 2 the necessary notation and basic definitions are introduced. Next, in section 3 we relate the problem of checking (multivariate) first order dominance to the classical transportation problem. From this insight we provide a new characterization of first order dominance for the general multivariate case. Then we turn to the bivariate case in section 4, where we provide the two algorithms for checking dominance. We construct an example, in section 5, on how to apply the $O(n)$ algorithm to the numerical assessment of first order dominance between continuous bivariate distributions. Section 6 concludes. All proofs are in the appendix.

2 Notation and definitions

An outcome is a vector $\mathbf{x} = (x_1, \ldots, x_m)$, where each attribute x_j is from an attribute set $X_j = \{1, 2, ..., n_j\}$, $j = 1, \ldots, m$, and $m \geq 2$. The outcome set is the product set $X = X_1 \times \ldots \times X_m$ and has cardinality $n = \Pi_{j=1}^{m} n_j$. If $m = 2$, then we have a bivariate case. For any two elements, $\mathbf{x}, \mathbf{y} \in X$, we define $\mathbf{y} \leq \mathbf{x}$ such that $y_j \leq x_j$ for all j, and $\mathbf{y} < \mathbf{x}$ such that $y_j \leq x_j$ for all j and $\mathbf{y} \neq \mathbf{x}$. A set $Y \subseteq X$ is called *lower comprehensive* if $\mathbf{x} \in Y$, $\mathbf{y} \in X$, and $\mathbf{y} \leq \mathbf{x}$ imply $\mathbf{y} \in Y$. An illustration of a lower comprehensive set Y in the bivariate case is given in right part of Figure 1.

A *probability mass function* is a real-valued function f on X, such that $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in X$ and $\sum_{\mathbf{x} \in X} f(\mathbf{x}) = 1$. For $\mathbf{x}, \mathbf{y} \in X$ where $\mathbf{x} \neq \mathbf{y}$ we say that a probability mass function g can be obtained from a probability mass function f by a *transfer* (of probability mass) if we have that $g(\mathbf{z}) = f(\mathbf{z})$ for all $\mathbf{z} \in X \setminus \{\mathbf{x}, \mathbf{y}\}$, $g(\mathbf{y}) = f(\mathbf{y}) + \beta$, and $g(\mathbf{x}) = f(\mathbf{x}) - \beta$ for some $\beta \in [-1, 1]$. Note that if g and f are identical except for the values in $\mathbf{x}$ and $\mathbf{y}$, then we can obtain g from f by transferring a suitable amount, β, of probability mass between $\mathbf{x}$ and $\mathbf{y}$.[7] A *diminishing* transfer (of probability mass) is a shift of probability mass from one outcome, $\mathbf{x}$, to another, $\mathbf{y}$, for $\mathbf{y} < \mathbf{x}$. A diminishing transfer for the bivariate case is shown in the left part of Figure 1.

Let f and g denote two probability mass functions. A fundamental result of stochastic dominance theory tells that the following three statements are equivalent:[8]

[7] If $f(\mathbf{y}) \leq g(\mathbf{y})$ then $f(\mathbf{x}) \geq g(\mathbf{x})$ and $\beta = g(\mathbf{y}) - f(\mathbf{y}) = f(\mathbf{x}) - g(\mathbf{x})$. Thus the transfer we consider is to increase $f(\mathbf{y})$ by β and decrease $f(\mathbf{x})$ by the same amount. The resulting probability mass function will then be identical with g.

[8] See footnote 3 for references to literature establishing these equivalences.

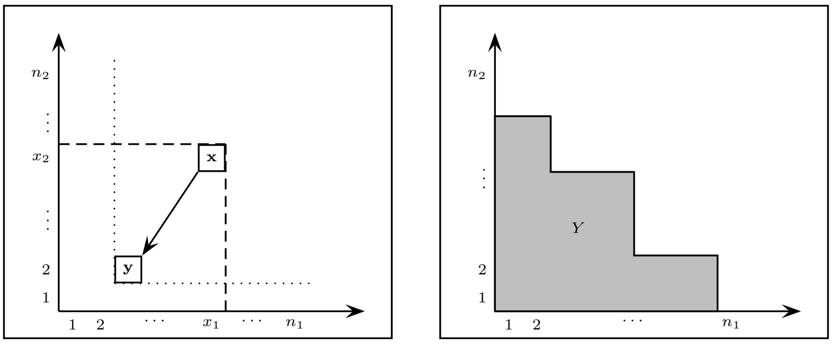

Fig. 1 Left: The lower set $L(\mathbf{x})$ is all elements below the dashed lines, the upper set $U(\mathbf{y})$ is all elements above the dotted lines, and the arrow is a diminishing transfer from $\mathbf{x}$ to $\mathbf{y}$. Right: A comprehensive set Y.

(A) g can be obtained from f by a finite number of diminishing transfers.
(B) $\sum_{\mathbf{x}\in Y} g(\mathbf{x}) \geq \sum_{\mathbf{x}\in Y} f(\mathbf{x})$ for any lower comprehensive set $Y \subseteq X$.
(C) $\sum_{\mathbf{x}\in X} u(\mathbf{x})f(\mathbf{x}) \geq \sum_{\mathbf{x}\in X} u(\mathbf{x})g(\mathbf{x})$ for every non-decreasing function u.[9]

We say that f *first order dominates* g if one of these three (equivalent) properties hold.

Example 1 Let $X_1 = \{1, 2, 3\}$ and $X_2 = \{1, 2, 3\}$ and f, g, h be bivariate probability mass functions on $X = X_1 \times X_2$ as given in Figure 2. We can

f	1	2	3
3	0.15	0.20	0.15
2	0.05	0.10	0.12
1	0.10	0.10	0.03

g	1	2	3
3	0.13	0.07	0.14
2	0.08	0.25	0.07
1	0.15	0.07	0.04

h	1	2	3
3	0.13	0.07	0.18
2	0.08	0.18	0.09
1	0.22	0.02	0.03

Fig. 2 Probability mass functions f, g, and h for Example 1.

observe that f first order dominates g as we can obtain g from f by the set of diminishing probability transfers given in Table 1. In this table $z_{\mathbf{pr}}$ is the amount of probability mass transferred from $\mathbf{p}$ to $\mathbf{r}$, while $c_{\mathbf{p}}$ and $\rho_{\mathbf{r}}$ are variables defined in section 3.2 and 4.1 used later in the paper by Algorithm 1. On the other hand, we can argue that f does not first order dominate h, as the lower comprehensive set $Y = X \setminus \{(3, 3)\}$ has

$$\sum_{\mathbf{x}\in Y} h(\mathbf{x}) = 0.82 < 0.85 = \sum_{\mathbf{x}\in Y} f(\mathbf{x})$$

[9] A real-valued function u on X is *non-decreasing* if $\mathbf{x}, \mathbf{y} \in X$ and $\mathbf{y} \leq \mathbf{x}$ implies $u(\mathbf{y}) \leq u(\mathbf{x})$.

	from ($\mathbf{p}$)	to ($\mathbf{r}$)	amount $z_{\mathbf{pr}}$	$c_{\mathbf{p}}$	$\rho_{\mathbf{r}}$
1	$(1,3)$	$(1,2)$	0.02	0	0.01
2	$(2,3)$	$(1,2)$	0.01	0.12	0
3	$(2,3)$	$(2,2)$	0.12	0	0.03
4	$(2,1)$	$(1,1)$	0.03	0	0.02
5	$(3,3)$	$(2,2)$	0.01	0	0.02
6	$(3,2)$	$(2,2)$	0.02	0.03	0
7	$(3,2)$	$(1,1)$	0.02	0.01	0
8	$(3,2)$	$(3,1)$	0.01	0	0

Table 1 Sequence of diminishing transfers showing that f first order dominates g in Example 1.

and we therefore have a lower comprehensive set violating property (B).

We add the following notation to ease the description of first order dominance. Given an outcome $\mathbf{x} \in X$, let $L(\mathbf{x}) = \{\mathbf{y} \in X | \mathbf{y} \leq \mathbf{x}\}$ be the lower orthant set relative to $\mathbf{x}$ and $U(\mathbf{x}) = \{\mathbf{y} \in X | \mathbf{x} \leq \mathbf{y}\}$ be the upper orthant set relative to $\mathbf{x}$. We will refer to these sets as the lower set and the upper set, respectively. An example of both a lower set and an upper set is given in the left part of Figure 1. For two probability mass functions, f and g, we define the real valued function $s(\mathbf{x}) := f(\mathbf{x}) - g(\mathbf{x})$ for $\mathbf{x} \in X$ along with the sets $P = \{\mathbf{x} \in X | s(\mathbf{x}) > 0\}$ and $R = \{\mathbf{x} \in X | s(\mathbf{x}) < 0\}$. For the elements of R, probability mass has to be added to f (to become equivalent to g) and, likewise, for the elements of P, probability mass has to be subtracted from f (to become equivalent to g). Thus, for f to first order dominate g, the elements of P have excess probability mass which needs to be transferred to one or more elements of R by diminishing transfers. More precisely, for each element $\mathbf{p} \in P$ we have to transfer $s(\mathbf{p})$ probability mass to a number of elements in $L(\mathbf{p}) \cap R$. In a similar vein, each element $\mathbf{r} \in R$ requires $-s(\mathbf{r})$ probability mass to be transferred from a number of elements in $U(\mathbf{r}) \cap P$, for f to become equivalent to g by diminishing transfers.

3 General characterizations

As mentioned in the introduction, for checking first order dominance it is inefficient to enumerate and check every lower comprehensive set explicitly. Instead we turn to the identification of a finite number of diminishing probability mass transfers and relate this to the transportation problem.

3.1 Relation to the transportation problem

First we need the following observation. When using diminishing transfers to obtain g from f it should be noted that we can omit all the elements $\mathbf{x} \in X$ having $s(\mathbf{x}) = 0$. The reason is that if we transfer some probability mass from $\mathbf{x}$ to an element $\mathbf{z}$ of $L(\mathbf{x}) \setminus \{\mathbf{x}\}$, then the same amount of probability mass has to be transferred to $\mathbf{x}$ from some element of $\mathbf{y} \in U(\mathbf{x}) \setminus \{\mathbf{x}\}$. By the definition of $U(\mathbf{x})$ we must have that $L(\mathbf{x}) \subset L(\mathbf{y})$, and the transfer to and from $\mathbf{x}$ can therefore be replaced by a direct transfer from $\mathbf{y}$ to $\mathbf{z}$. More generally, if a sequence of diminishing transfers uses intermediate elements of X, then, as just described, it is always possible to replace one or more of these diminishing transfers with a direct diminishing transfer. Consequently, no outcome needs to both send and receive probability mass. We let $C = \{(\mathbf{p}, \mathbf{r}) \in P \times R | \mathbf{r} \in L(\mathbf{p})\}$ be the pairs of outcomes which correspond to possible (direct) diminishing transfers.

With the definitions given in section 2 we can formulate the problem of checking first order dominance between two finite multivariate distributions as a bipartite network problem.[10] It is essentially a transportation problem where the "suppliers" from P have to transport a required amount to the "customers" from R. If it is possible to transport probability mass from $\mathbf{p} \in P$ to $\mathbf{r} \in R$, i.e. $(\mathbf{p}, \mathbf{r}) \in C$, we incur a unit cost of zero of transportation from $\mathbf{p}$ to $\mathbf{r}$, whereas if it is not possible to transport probability mass from $\mathbf{p}$ to $\mathbf{r}$, then a unit cost of one is incurred for transporting probability mass from $\mathbf{p}$ to $\mathbf{r}$. Solving the resulting transportation problem either yields a zero value objective, in which case we have identified a finite set of diminishing transfers, or a strictly positive objective showing that it is necessary to send mass from $\mathbf{p} \in P$ to a $\mathbf{r} \notin L(\mathbf{p}) \cap R$. Hence, if the objective is zero, then we have by (A) that f first order dominates g, whereas if the objective is positive, then f does not first order dominate g. If we let $b = \max\{|P|, |R|\}$

10 Preston [32] formulates this as a maximum flow problem which can be solved in $O(n^3)$ by using the method described by Orlin [31]. This can, however, be improved by setting up the network in a slightly different way. That is, instead of having an arc from $\mathbf{x}$ to all nodes in $L(\mathbf{x})$ we only add arcs from $\mathbf{x}$ to $\mathbf{y} \in L(\mathbf{x})$ having $\|\mathbf{x} - \mathbf{y}\|_2 \leq 1$. Thus, arcs only exist from $(x_1, \ldots, x_k, \ldots, x_m)$ to $(x_1, \ldots, x_k - 1, \ldots, x_m)$ for each $k = 1, \ldots, m$, in other words, all nodes of the lower set which are adjacent to $\mathbf{x}$. This yields a graph with n nodes and nm arcs. For fixed dimension m the number of arcs is then linearly bounded by the number of nodes, and we can therefore solve the corresponding max-flow problem in $O(n^2/\log n)$ by the method described by Orlin [31]. The linear programming based approaches described by Mosler and Scarsini [28] and Dyckerhoff and Mosler [10] are typically solved by means of pseudo-polynomial algorithms such as the Simplex algorithm; see Schrijver [36].

and $d = \min\{|P|, |R|\}$, and k is the number of feasible connections, then this problem can be solved in $O(b \log b(k + d \log d))$ by the method described by Kleinschmidt and Schannath [19]. Furthermore, there is a set-up cost of $\Theta(n)$ for identifying P and R from X and of $O(|P||R|)$ for identifying which arcs are feasible. Note that $|P||R| = bd$, which is part of the algorithm's complexity. Thus, using a transportation problem algorithm directly yields a time complexity of $O(n + b \log b(k + d \log d))$.

3.2 A stronger characterization

Below we present an alternative linear programming model which is based on the observation that we just need to identify a feasible transportation problem solution having objective value equal to zero. Let $z_{\mathbf{pr}} \geq 0$ be the amount of probability mass transferred from $\mathbf{p}$ to $\mathbf{r}$, where $(\mathbf{p}, \mathbf{r}) \in C$. Furthermore, let $c_{\mathbf{p}} \geq 0$ for $\mathbf{p} \in P$ and $d_{\mathbf{r}} \geq 0$ for $\mathbf{r} \in R$ be two sets of auxiliary variables. For given values of the $z_{\mathbf{pr}}$ variables, $c_{\mathbf{p}}$ measures the amount of probability mass that remains to be transferred out of $\mathbf{p} \in P$ to elements $\mathbf{r} \in L(\mathbf{p}) \cap R$ (in order to reach $s(\mathbf{p})$) where $c_{\mathbf{p}}$ attains a value of zero when a sufficient amount of probability mass is transferred out of $\mathbf{p}$. Likewise, $d_{\mathbf{r}}$ measures the excess amount (compared to $-s(\mathbf{r})$) of probability mass transferred to element $\mathbf{r} \in R$ from the elements in $U(\mathbf{r}) \cap P$ where $d_{\mathbf{r}}$ attains a value of zero when not more than the required amount of probability mass has been transferred to $\mathbf{r}$. Then the problem is to identify a feasible set of transfers such that the amount of probability mass which cannot be transferred out of $\mathbf{p} \in P$ and the amount of probability mass received beyond $-s(\mathbf{r})$ for $\mathbf{r} \in R$ is minimized. Hence, we have to solve the following linear program:

$$Z^* = \min \sum_{\mathbf{p} \in P} c_{\mathbf{p}} + \sum_{\mathbf{r} \in R} d_{\mathbf{r}} \tag{1}$$

$$\text{s.t.} \sum_{\mathbf{r} \in L(\mathbf{p}) \cap R} z_{\mathbf{pr}} + c_{\mathbf{p}} \geq s(\mathbf{p}), \quad \mathbf{p} \in P \tag{2}$$

$$\sum_{\mathbf{p} \in U(\mathbf{r}) \cap P} z_{\mathbf{pr}} - d_{\mathbf{r}} \leq -s(\mathbf{r}), \quad \mathbf{r} \in R \tag{3}$$

$$z_{\mathbf{pr}} \geq 0, \quad (\mathbf{p}, \mathbf{r}) \in C \tag{4}$$

$$c_{\mathbf{p}} \geq 0, \quad \mathbf{p} \in P \tag{5}$$

$$d_{\mathbf{r}} \geq 0, \quad \mathbf{r} \in R \tag{6}$$

The objective (1) of this problem is to minimize the untransferred probability mass from the elements of P as well as the excess probability mass transferred to elements of R. The first constraint (2) states that each element $\mathbf{p} \in P$ either has to transfer to the elements in $L(\mathbf{p}) \cap R$ or leave some of the probability mass untransferred. The second constraint (3) states that an element of $\mathbf{r}$ cannot receive more than $-s(\mathbf{r})$ probability mass from the elements in $U(\mathbf{r}) \cap P$, but if it does, then the excess probability mass is added to $d_{\mathbf{r}}$.[11] Finally, the constraints (4)-(6) just state the non-negativity of the variables.

For the problem (1)-(6), a vector $(\mathbf{z}, \mathbf{c}, \mathbf{d})$, where $\mathbf{z} = (z_{\mathbf{pr}})_{(\mathbf{p},\mathbf{r}) \in C}$, $\mathbf{c} = (c_{\mathbf{p}})_{\mathbf{p} \in P}$, and $\mathbf{d} = (d_{\mathbf{r}})_{\mathbf{r} \in R}$ is denoted a solution. A solution is said to be feasible if it satisfies all the constraints (2)-(6). Furthermore, a solution is said to be optimal if it is feasible and minimizes objective (1). If a solution is optimal, then it is denoted $(\mathbf{z}^*, \mathbf{c}^*, \mathbf{d}^*)$. It can be shown that a feasible solution to (1)-(6) with $\sum_{\mathbf{p} \in P} c_{\mathbf{p}} + \sum_{\mathbf{r} \in R} d_{\mathbf{r}} = 0$ has the characteristic that the inequality constraints (2) and (3) have to be binding.

Our application of formulation (1)-(6) is twofold. First, we use it to strengthen the characterization of first order dominance. More precisely we can use the relation to the transportation problem to obtain an upper bound on the number of diminishing transfers necessary to show finite (multivariate) first order dominance. We will argue below that this upper bound corresponds to $|P|+|R|-1$. Second, in section 4 we use this formulation as well as the strengthened characterization to construct efficient algorithms for checking first order dominance in the bivariate case.

Clearly, if the optimal solution of problem (1)-(6) is zero, i.e. $Z^* = 0$, then we have a feasible finite series of diminishing transfers – showing that f first order dominates g – whereas a positive objective value corresponds to the case where no feasible series of diminishing transfers exists. In the latter case we conclude that f does not first order dominate g.

Lemma 1 *f first order dominates g if and only if a vector $\mathbf{z} \in \mathbb{R}^{|C|}$ exists with $\mathbf{z} \geq \mathbf{0}$ and*

$$\sum_{\mathbf{r} \in L(\mathbf{p}) \cap R} z_{\mathbf{pr}} = f(\mathbf{p}) - g(\mathbf{p}), \quad \forall \mathbf{p} \in P \tag{7}$$

$$\sum_{\mathbf{p} \in U(\mathbf{r}) \cap P} z_{\mathbf{pr}} = g(\mathbf{r}) - f(\mathbf{r}), \quad \forall \mathbf{r} \in R \tag{8}$$

[11] To be consistent we use the convention that a sum of no elements is zero, i.e. if $L(\mathbf{p}) \cap R = \emptyset$, then $\sum_{\mathbf{r} \in L(\mathbf{p}) \cap R} z_{\mathbf{pr}} = 0$, and if $U(\mathbf{r}) \cap P = \emptyset$, then $\sum_{\mathbf{p} \in U(\mathbf{r}) \cap P} z_{\mathbf{pr}} = 0$.

It can be observed that the equations (7) and (8) in conjunction with $\mathbf{z} \geq \mathbf{0}$ describes a bounded polyhedron. Any vertex in this bounded polyhedron has at most $|P|+|R|-1$ non-zero elements, as (7) and (8) are linearly dependent. This observation gives rise to the following result:

Theorem 1 *f first order dominates g if and only if g can be obtained from f by at most $|P|+|R|-1$ diminishing probability mass transfers.*

Theorem 1 gives a sharper characterization of first order dominance than the part (A) of the equivalence. While we thereby can bound the needed number of transfers to show first order dominance it should be noted that alternative finite sequences of diminishing transfers often exists which use more than $|P|+|R|-1$ diminishing transfers. However, the algorithms we provide in this paper will search for sets of diminishing transfers which are no larger than $|P|+|R|-1$. This is done by gradually – in each iteration – satisfying the amount transferred out of elements of P or the amount transferred into the elements of R.

For the development of algorithms in the following section another important property of the linear programming model (1)-(6) should be noted. That is (1)-(6) may have alternative solutions yielding the same objective value. We make the following observation for a feasible solution.

Lemma 2 (Alternative solutions) *Let $(\overline{\mathbf{z}}, \overline{\mathbf{c}}, \overline{\mathbf{d}})$ be a feasible solution for (1)-(6). Let $\mathbf{x}, \mathbf{y} \in P$ and $\mathbf{v}, \mathbf{w} \in L(\mathbf{x}) \cap L(\mathbf{y}) \cap R$ and put*

$$\beta = \min\left\{\overline{z}_{\mathbf{xv}}, \overline{z}_{\mathbf{yw}}\right\}. \tag{9}$$

If $\beta > 0$, then we can construct an alternative solution $(\mathbf{z}', \mathbf{c}', \mathbf{d}')$ having $\mathbf{c}' = \overline{\mathbf{c}}$, $\mathbf{d}' = \overline{\mathbf{d}}$, and all elements of $\mathbf{z}'$ equal to the corresponding elements of $\overline{\mathbf{z}}$ except for

$$z'_{\mathbf{xv}} = \overline{z}_{\mathbf{xv}} - \beta \tag{10}$$

$$z'_{\mathbf{yw}} = \overline{z}_{\mathbf{yw}} - \beta \tag{11}$$

$$z'_{\mathbf{xw}} = \overline{z}_{\mathbf{xw}} + \beta \tag{12}$$

$$z'_{\mathbf{yv}} = \overline{z}_{\mathbf{yv}} + \beta \tag{13}$$

with the same objective value.

The observation from Lemma 2 is illustrated for the bivariate case in Figure 3. The full arrows are the transfers which are decreased, whereas the dashed

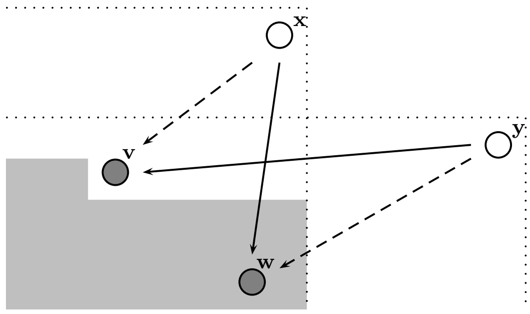

Fig. 3 Swap of probability mass transfer.

arrows are the transfers which are increased. Lemma 2 will be used to show that given any solution to (1)-(6) with objective value zero we can construct an alternative solution following a specific pattern. As a consequence, it will only be necessary to search for this pattern when checking first order dominance in the bivariate case.

4 Two algorithms for the bivariate case

In this section, first we introduce a constructive $O(n^2)$ algorithm for identifying a finite sequence of diminishing transfers. Then we observe that it is actually not necessary to construct the finite sequence of diminishing transfers directly to show that such a sequence exists, and we use this observation as a basis for an $O(n)$ algorithm determining whether or not f first order dominates g.

For both algorithms the elements of X are traversed in a specific order. Then we introduce the direct algorithm, and finally we introduce the indirect algorithm for checking first order dominance.

We use two specific complete orderings of the elements of X. These orderings are used when iterating through the elements of X. We say that an element $\mathbf{x} = (x_1, x_2) \in X$ has a lower $(1,2)$-order than $\mathbf{y} = (y_1, y_2) \in X$ if $x_1 < y_1$ or if $x_1 = y_1$ and $x_2 > y_2$. In case $\mathbf{x}$ has a lower $(1,2)$-order than $\mathbf{y}$ then we write $o_{12}(\mathbf{x}) < o_{12}(\mathbf{y})$ and we say that $\mathbf{y}$ has higher $(1,2)$-order than $\mathbf{x}$. The left part of Figure 4 shows the elements with lower $(1,2)$-order than $\mathbf{x}$ as white and the elements with higher $(1,2)$-order as gray. Analogously, we say that an element $\mathbf{x} = (x_1, x_2) \in X$ has a lower $(2,1)$-order than $\mathbf{y} = (y_1, y_2) \in X$ if $x_2 < y_2$ or if $x_2 = y_2$ and $x_1 > y_1$. In case $\mathbf{x}$ has a lower $(2,1)$-order than $\mathbf{y}$ then we write $o_{21}(\mathbf{x}) < o_{21}(\mathbf{y})$ and we say that $\mathbf{y}$ has higher $(2,1)$-order than **x**. The right part of Figure 4 illustrates $(2,1)$-ordering, where the white elements have lower $(2,1)$-order than **x** and the gray elements have higher $(2,1)$-ordering than **x**.

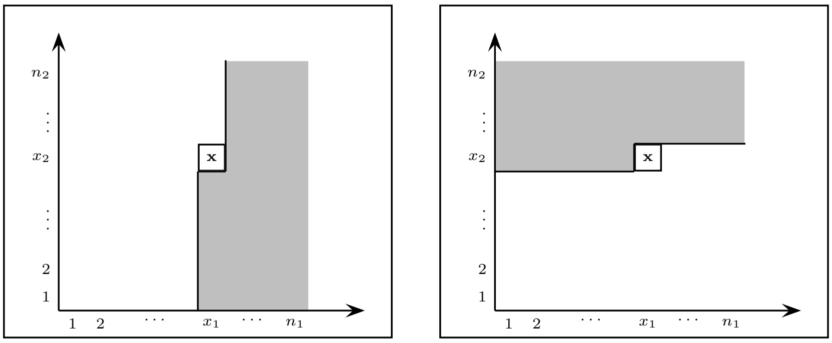

Fig. 4 Left: All gray elements have higher $(1,2)$-order than **x**. Right: All gray elements have higher $(2,1)$-order than **x**

Suppose that we have a subset of elements $Y \subseteq X$ and given that we have an ordering (a,b) either equal to $(1,2)$ or equal to $(2,1)$ then we say that an element $x \in Y$ has minimal (a,b)-order with respect to Y if no other element $y \in Y$ exists such that $o_{ab}(y) < o_{ab}(x)$. Likewise we say that $x \in Y$ has maximal (a,b)-order with respect to Y if no other element $y \in Y$ exists such that $o_{ab}(y) > o_{ab}(x)$.

4.1 A direct approach

In the direct algorithm we explicitly identify the set of diminishing transfers by using the orderings described above. The resulting algorithm can be solved in $O(n^2)$ in the worst case. The algorithm identifies a set of diminishing transfers which may yield a solution value Z^* of zero, and we prove (in Lemma 3) that if a set of diminishing transfers yielding $Z^* = 0$ exists for problem (1)-(6), then the algorithm will identify such a set.

The idea is that we start with a feasible – but not necessarily optimal – solution to problem (1)-(6) and by carefully selecting the diminishing transfers which should have a positive value we gradually decrease the objective value until it is either zero or we can show that it is not possible to obtain a value of zero.

We manipulate the variables of the formulation (1)-(6) in such a way that we keep $d_{\mathbf{r}} = 0$ for all $\mathbf{r} \in R$ while keeping constraint (2) binding. Hence, we have that $c_{\mathbf{p}} = s(\mathbf{p}) - \sum_{\mathbf{r} \in L(\mathbf{p}) \cap R} z_{\mathbf{pr}}$ for all $\mathbf{p} \in P$, and we put $\rho_{\mathbf{r}} = -s(\mathbf{r}) - \sum_{\mathbf{p} \in U(\mathbf{r}) \cap P} z_{\mathbf{pr}}$. Clearly, $\rho_{\mathbf{r}}$ corresponds to the slack variable of constraint (3) and has to be non-negative. Implicitly, we initialize $z_{\mathbf{pr}} = 0$ for all pairs $(\mathbf{p}, \mathbf{r}) \in P \times R$.[12] Thus, increasing $z_{\mathbf{pr}}$ will decrease both $c_{\mathbf{p}}$ and $\rho_{\mathbf{r}}$, and the aim is to identify a sequence of increases of $z_{\mathbf{pr}}$ such that both $c_{\mathbf{p}}$ and $\rho_{\mathbf{r}}$ become zero for all $\mathbf{p}$ and $\mathbf{r}$. If the values of $c_{\mathbf{p}}$ are zero, and because we maintain the invariant of (2) being binding, the solution found will satisfy (7). Likewise if $\rho_{\mathbf{r}}$ is zero and, because we maintain $d_{\mathbf{r}}$ at zero, we have that (8) is satisfied. Hence the solution found satisfies Lemma 1 and, consequently, we have shown that f first order dominates g.

Algorithm 1 Direct transfer algorithm

Step 0: Initialize $c_{\mathbf{p}} = s(\mathbf{p})$ for all $\mathbf{p} \in P$ and $\rho_{\mathbf{r}} = -s(\mathbf{r})$ for all $\mathbf{r} \in R$.

Step 1: Let $P^+ = \{\mathbf{p}' \in P | c_{\mathbf{p}'} > 0\}$. Select $\mathbf{p} \in P^+$ with minimal $(1,2)$-order, $o_{12}(\mathbf{p})$, with respect to P^+. If $P^+ = \emptyset$, then terminate, continue otherwise.

Step 2: Let $R^+ = \{\mathbf{r}' \in R | \rho_{\mathbf{r}'} > 0\}$. Select $\mathbf{r} \in L(\mathbf{p}) \cap R^+$ with maximal $(2,1)$-order, $o_{21}(\mathbf{r})$, with respect to R^+. If $L(\mathbf{p}) \cap R^+ = \emptyset$, then terminate else continue.

Step 3: Update

$$\begin{aligned} z_{\mathbf{pr}} &= \min\{c_{\mathbf{p}}, \rho_{\mathbf{r}}\} \\ c_{\mathbf{p}} &= c_{\mathbf{p}} - z_{\mathbf{pr}} \\ \rho_{\mathbf{r}} &= \rho_{\mathbf{r}} - z_{\mathbf{pr}} \end{aligned} \tag{14}$$

Go to step 1.

The direct approach is given in Algorithm 1. It initializes the variables and then repeats three steps, which results in either a solution to the problem (1)-(6) with a value zero (terminating in step 1) or a positive objective value (terminating in step 2).

An example of the progression in the three steps is given in the left part of Figure 5. The element with lowest $(1,2)$-order will be in the upper left corner, whereas the element with the highest $(1,2)$-order will be in the lower right corner of X. The figure shows the sequence of selections of elements of P and R. Elements of P are white nodes, whereas elements of R are black nodes. The first element of P encountered is $\mathbf{p}^1$, and it transfers probability mass to $\mathbf{r}^6$, $\mathbf{r}^5$, $\mathbf{r}^4$, and $\mathbf{r}^3$. While $\mathbf{r}^6$, $\mathbf{r}^5$, and $\mathbf{r}^4$ are fully saturated, $\mathbf{r}^3$ only received a fraction of $-s(\mathbf{r}^3)$, and it can therefore receive more later in the algorithm.

[12] It is not necessary to initialize $z_{\mathbf{pr}}$ explicitly as we just need to keep track of which pairs $(\mathbf{p}, \mathbf{r})$ have increased values of $z_{\mathbf{pr}}$.

The sequence of diminishing transfers of probability mass away from $\mathbf{p}^1$ is illustrated by the full black arrows in the figure. The same is then done for $\mathbf{p}^2$ and $\mathbf{p}^3$, where the gray arrows represent the sequence of diminishing transfers from $\mathbf{p}^2$ and the dashed arrows show the sequence of diminishing transfers from $\mathbf{p}^3$.

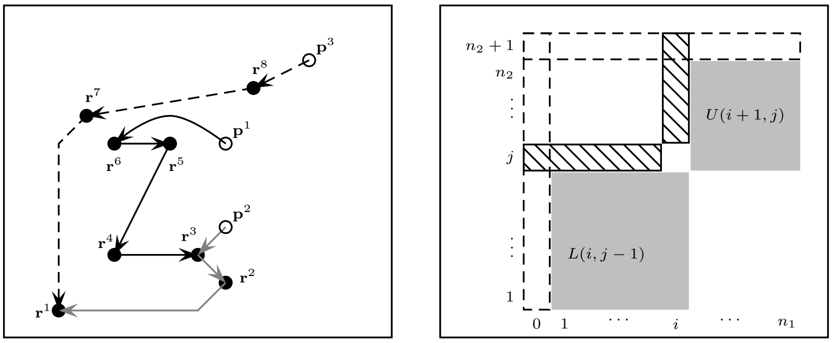

Fig. 5 Left: The process of the $O(n^2)$ algorithm. Right: The basic sets for the $O(n)$ algorithm.

While it is clear that f first order dominates g when identifying a solution satisfying Lemma 1, it is not clear that f does not first order dominate g if Algorithm 1 fails to identify a solution satisfying Lemma 1. We can, however, show that if a set of diminishing transfers satisfying Lemma 1 exists, then the sequence used in Algorithm 1 will also identify a solution satisfying Lemma 1. This is summarized in Lemma 3 below.

Lemma 3 *If a solution* $\mathbf{z} \geq 0$ *exists such that* (7) *and* (8) *are both satisfied then the sequence of diminishing transfers* $\mathbf{z}' \geq 0$ *identified by Algorithm 1 will also satisfy* (7) *and* (8).

Lemma 3 uses Lemma 2 with $\overline{\mathbf{c}} = \overline{\mathbf{d}} = 0$ to transform any solution satisfying (7) and (8) into the solution obtained by Algorithm 1. Consequently the solution found by Algorithm 1 is always an alternative solution to any solution satisfying (7) and (8) and if a solution found by Algorithm 1 fails to satisfy (7) and (8) then no other solution will satisfy (7) and (8). Now we can state the correctness and worst case complexity of Algorithm 1:

Theorem 2 *Algorithm 1 terminates in at most* $O(n^2)$ *iterations and either terminates with a finite sequence of diminishing transfers or shows that no such sequence exists.*

A numerical example of Algorithm 1 is given in Table 1 for the two probability mass functions f and g in figure 2. The sequence of the algorithm corresponds to the sequence of the entries in the table and the values of $z_{\mathbf{pr}}$, $c_{\mathbf{p}}$, and $\rho_{\mathbf{r}}$ corresponds to the values after updating in Step 3.

Algorithm 1 is an intuitive approach for checking first order dominance. However, we can improve on the worst case complexity of $O(n^2)$ to a worst case complexity of $O(n)$, by observing that we do not always need to identify the individual diminishing transfers but only diminishing transfers between certain sets. We will observe that we can use the same sequence as in Algorithm 1 but without doing the explicit transfers.

4.2 An indirect approach

We now present a more efficient algorithm for checking first order dominance. Like Algorithm 1 it is based on iterating through the elements of X in increasing $(1,2)$-order, but it only records how much probability mass is transferred between specific aggregated subsets without specifying the transfers directly.

To ease the notation, we expand the sets X_1 and X_2 to $\overline{X}_1 = X_1 \cup \{0\}$ and $\overline{X}_2 = X_2 \cup \{n_2+1\}$. Now we use the simpler notation $\mathbf{x} = (i,j)$ for $i \in \overline{X}_1$ and $j \in \overline{X}_2$. Furthermore, we let $f_{ij} = f(\mathbf{x})$, $g_{ij} = g(\mathbf{x})$, and $s_{ij} = s(\mathbf{x})$. We let $f_{0j} = g_{0j} = s_{0j} = 0$ for all $j \in \overline{X}_2$, and $f_{i,n_2+1} = g_{i,n_2+1} = s_{i,n_2+1} = 0$ for all $i \in \overline{X}_1$. We say that elements $(i,\cdot)$ are the column of i, and the elements $(\cdot,j)$ are the row of j. See the right part of Figure 5 for an illustration of the set-up, where the dashed boxes correspond to the artificial elements added, and the hatched boxes correspond to the row and column elements in row j and column i having a lower $(2,1)$-order and higher $(1,2)$-order, respectively, compared to the element (i,j).

Algorithm 2 is an indirect method for determining whether or not f first order dominates g without constructing a finite number of diminishing transfers. That is, in contrast to Algorithm 1, it does not construct a vector $\mathbf{z} \geq \mathbf{0}$ satisfying Lemma 1, explicitly. Instead Algorithm 2 identifies how much probability mass has to be transferred from $S_{ij} = \{(h,k) \in X | h = i, k \geq j\}$ to elements of $T_{ij} = \{(h,k) \in X | h \leq i, k = j\}$. In the following we will describe the elements of Algorithm 2.

The algorithm iterates through X in increasing $(1,2)$-order i.e. it processes the columns in increasing order and within the columns the rows are processed in decreasing order. Consequently, at the time of processing column i

Algorithm 2 Indirect transfer algorithm

Step 0: Let $f_{0j} = g_{0j} = e_{0j} = 0$ for all $j \in \overline{X}_2$ and $f_{i,n_2+1} = g_{i,n_2+1} = u_{i,n_2+1} = 0$ for all $i \in \overline{X}_1$.
Let $i = 1$, $j = n_2$.
Step 1: Calculate

$$\begin{aligned} s_{ij} &= f_{ij} - g_{ij} \\ t_{ij} &= u_{i,j+1} - e_{i-1,j} + s_{ij} \\ u_{ij} &= \max\{0, t_{ij}\} \\ e_{ij} &= \max\{0, -t_{ij}\} \end{aligned}$$

Step 2: Choose one of the following
- If $j > 1$, then put $j = j - 1$ and go to step 1.
- If $j = 1$, $i < n_1$, and $u_{i1} = 0$, then put $i = i + 1$, $j = n_2$ and goto step 1.
- If $j = 1$ and $u_{i1} > 0$, then return FALSE.
- If $j = 1$, $i = n_1$, and $u_{i1} = 0$, then return TRUE.

all columns $h < i$ have already been processed. The reason for processing the rows within the columns in decreasing order is to transfer as much as possible to the top rows as early as possible such that it does not block transfers in later iterations. When processing row j within column i the algorithm has already transferred as much as possible from elements in column i with larger row indices than j to rows having larger row indices than row j. The amount which cannot be transferred when iterating through X in $(1, 2)$-order is denoted u_{ij}.

At the time of reaching element (i, j) the amount $u_{i,j+1}$ still remains to be transferred out of some elements (i, h) with $h \geq j + 1$ to some elements of $L(i, j)$. Furthermore, if $s_{ij} > 0$ then this in addition has to be transferred to $L(i, j)$. Again we are interested in transferring as much as possible to rows with as large row indices as possible. In this case it is the row j i.e. transferring as much of $u_{i,j+1} + s_{ij}$ to elements in the row j with column indices $k < i$. The elements (k, j) with $k < i$ has already been processed and we do not block additional transfers by doing this. We let $e_{i-1,j}$ be the amount which can be transferred to these row elements. Hence, if $s_{ij} > 0$ then we transfer $\min\{u_{i,j+1} + s_{ij}; e_{i-1,j}\}$ from the elements S_{ij} to elements of $T_{i-1,j}$. In the case of $s_{ij} < 0$ then by an analogous argument we transfer $\min\{u_{i,j+1}; e_{i-1,j} - s_{ij}\}$ from $S_{i,j+1}$ to T_{ij}. We put $t_{ij} = u_{i,j+1} - e_{i-1,j} + s_{ij}$ and show, in the proof of Lemma 4, that $u_{ij} = \min\{0; t_{ij}\}$ and $e_{ij} = \min\{0; -t_{ij}\}$.

Example 2 (Example 1 continued) Figure 6 shows the results of applying Algorithm 2 to verify that f first order dominates g. Likewise we have in Figure 7 the results of applying Algorithm 2 to show that f does not first order dominate h. This can be seen by $u_{31} = 0.03 > 0$.

s	1	2	3
3	0.02	0.13	0.01
2	-0.03	-0.15	0.05
1	-0.05	0.03	-0.01

u	1	2	3
3	0.02	0.13	0.01
2	0.00	0.00	0.03
1	0.00	0.00	0.00

e	1	2	3
3	0.00	0.00	0.00
2	0.01	0.03	0.00
1	0.05	0.02	0.00

Fig. 6 Algorithm 2 testing whether or not f first order dominates g of Example 1.

s	1	2	3
3	0.02	0.13	-0.03
2	-0.03	-0.08	0.03
1	-0.12	0.08	0.00

u	1	2	3
3	0.02	0.13	0.00
2	0.00	0.04	0.03
1	0.00	0.00	**0.03**

e	1	2	3
3	0.00	0.00	0.03
2	0.01	0.00	0.00
1	0.12	0.00	0.00

Fig. 7 Algorithm 2 testing whether or not f first order dominates h of Example 1.

It is worth noting that the algorithm only determine how much is transferred between *sets* of elements. This is without determining the transfers between specific elements, and the solution found by the algorithm therefore represents a continuum of possible transfers rather than a single possible solution.

Lemma 4 describes the case where the algorithm returns that f first order dominates g, whereas Lemma 5 describes the case where the algorithm returns that f does not first order dominate g. Finally, Theorem 3 states the correctness and time complexity of the indirect algorithm.

Lemma 4 *If Algorithm 2 terminates with* $u_{n_1 1} = 0$*, then a finite sequence of diminishing transfers exists such that* g *can be obtained from* f*.*

Algorithm 2 iterates through the elements in increasing $(1, 2)$-order such that it starts in the top left corner and finishes in the bottom right corner. Hence, when it reaches element $(n_1, 1)$ then the algorithm has reached the bottom right corner. If it is reached without having any untransferred probability mass, then we have succeeded in transferring all excess probability mass out of the elements of P, which is what Lemma 4 states.

Lemma 5 *If Algorithm 2 terminates with* $u_{i1} > 0$ *a lower comprehensive set* $Y \subseteq X$ *exists such that* $\sum_{\mathbf{x} \in Y} g(\mathbf{x}) < \sum_{\mathbf{x} \in Y} f(\mathbf{x})$*.*

If for some i we reach the bottom and have not transferred all the required probability mass out of the elements of $L(i, n_2)$ then u_{i1} will be non-zero. Lemma 5 argues that if this happens, then we can always identify a lower comprehensive set which violates (B) in the fundamental equivalences of first order dominance.

As Algorithm 2 either terminates in the situation of Lemma 4 or in the situation of Lemma 5 we can directly obtain Theorem 3.

Theorem 3 *Algorithm 2 terminates in $O(n)$ iterations either stating that f first order dominates g or that f does not first order dominate g.*

It is possible to achieve the sequence of diminishing transfers without increasing the worst case time complexity by augmenting Algorithm 2. The proof of Lemma 4 uses insertion of elements into lists. These lists and the corresponding insertions could be added to an augmented version of Algorithm 2. Insertions and deletions of elements from the lists are only performed in the end of the lists, and we can therefore do this in $O(1)$ time complexity. Throughout the algorithm at most $|P|+|R|$ elements is inserted into the lists, as any element is inserted into one of the lists only once. Hence, the number of insertions and number of deletions are therefore bounded by $|P|+|R|$ and this augmentation will have a complexity of $O(n)$.

Furthermore, it is also possible to derive a violating lower comprehensive set without increasing the worst case time complexity by applying the constructive method in the proof of Lemma 5. The method uses no more than $\max\{|P|, |R|\}$ iterations of complexity $O(1)$, and Algorithm 2 can therefore be augmented with this construction and still have a worst case time complexity of $O(n)$.

5 Application to numerical assessment of first order dominance between continuous bivariate distributions

To give an indication of the speed and application of the indirect algorithm we have constructed a small computational example where we discretize two continuous bivariate distributions and check whether or not one first order dominates the other.

We illustrate the importance of fine-grained discretization when numerically establishing whether or not one bivariate probability mass function dominates another. We use two truncated normalized bivariate normal distributions as an example. They are depicted in figure 8. These distributions have been constructed as follows; the distribution f has mean $(500, 500)$ whereas g has mean $(450, 450)$, and they have covariance matrices of

$$Cov_f = \begin{bmatrix} 15000 & 8000 \\ 8000 & 10000 \end{bmatrix} \quad Cov_g = \begin{bmatrix} 9000 & 5000 \\ 5000 & 8000 \end{bmatrix}.$$

Using $X_1 = \{1, \ldots, 1024\}$ and $X_2 = \{1, \ldots, 1024\}$ we construct a discretization of each distribution by numerical integration of each discrete entry corre-

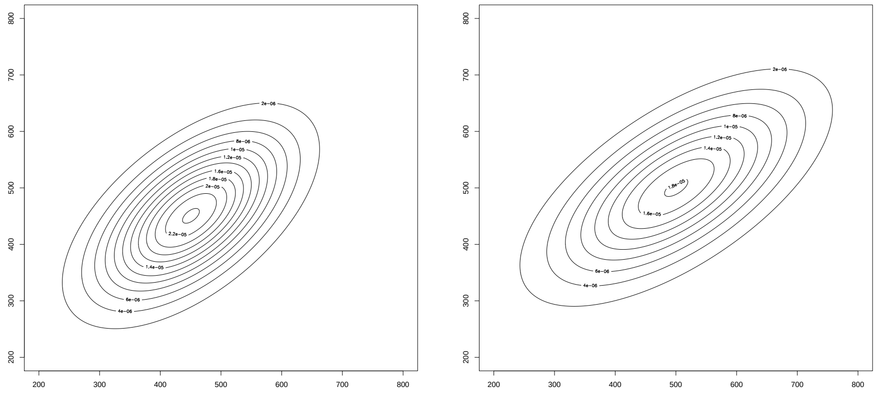

Fig. 8 Truncated normalized bivariate normal distributions, where g is given to the left and f is given to the right.

sponding to the unit box (x_1, x_2) to (x_1+1, x_2+1).[13] We truncate the value if it is less than $10e^{-7}$. To obtain a valid distribution we normalize the outcomes by the sum of all entries. The level curves are illustrated in Figure 8.

To test the effect of the granularity of the discretization we have intervals of length $1024/2^k$ where $k = 1, \ldots, 10$ yielding 2^{2k} outcomes. Hence for a unit increase in k the number of outcomes is quadrupled. Thus with a linear time complexity algorithm we would anticipate a quadrupled time consumption.

We have implemented Algorithm 2 in C++ and compiled it using the MinGW TDM64 5.1.0 compiler. The experiment was conducted on a single core of a computer using Microsoft Windows 7 equipped with an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and 32Gb RAM. To measure the run time (especially for the tests where k is small) we have repeated the algorithm 1,000,000 times.

The results of the experiment are given in Table 2. The first column gives the value of k determining the interval length, as described above, and the interval length is given in the next column, I. The resulting number of outcomes, $|X|$, is given in the next column. The following columns $|P|$, $|R|$, and $|C|$ indicates the size of the sets of outcomes having mass to send, mass to receive, and the number of feasible send-receive pairs for diminishing transfers. Finally, the column Time(s) indicates how many seconds it took to execute

[13] We have used the R package mvtnorm to obtain the probability mass in each unit box. This package uses numerical integration.

k	I	$\|X\|$	$\|P\|$	$\|R\|$	$\|C\|$	Time(s)	f dominates g
1	512	4	3	1	3	0.016	true
2	256	16	4	4	16	0.047	true
3	128	64	12	9	95	0.140	true
4	64	256	40	23	755	0.562	true
5	32	1,024	120	83	8,007	0.750	false
6	16	4,096	435	313	110,577	2.702	false
7	8	16,384	1,644	1,214	1,618,773	9.802	false
8	4	65,536	6,389	4,768	24,717,389	45.129	false
9	2	262,144	25,154	18,926	386,544,433	168.675	false
10	1	1,048,576	99,859	75,408	6,117,591,336	636.902	false

Table 2 Test of aggregated normalized bivariate truncated normal distribution

Algorithm 2 1,000,000 times. Finally, the result of the first order dominance test is shown in last column.

It is interesting to observe that different levels of aggregation yields different conclusions on whether or not f first order dominates g. For the coarse-grained levels ($k = 1, \ldots, 4$) f first order dominates g, but at the more fine-grained levels ($k = 5, \ldots, 10$) the conclusion is the opposite. This demonstrates that for numerical evaluation of continuous bivariate distributions it is necessary to have sufficiently high level of discretization.

The sizes $|P|$ and $|R|$ gives the number of constraints of problem (1)-(6) and $|C|$ corresponds to the number of $z_{\mathbf{pr}}$ variables in the problem. Especially the number of variables will be prohibitive if one has to solve the linear program directly for more fine-grained discretizations.

The time it takes to check first order dominance is – as anticipated – roughly quadrupled when making unit increases in k. However, there is one exception from $k = 4$ to $k = 5$ which reflects the fact that for $k = 5$ the algorithm terminates as soon as it observes that f does not first order dominate g.

The computational time for a single instance of the finely grained discretization where $k = 10$ is on average 0.000637 seconds i.e. a fraction of a millisecond. We consider this to be fast.

If we extrapolate based on the above observation and consider the case $k = 4$ – where we found dominance – and multiply the average time per instance $0.562/10^6$ by 2^{12} (corresponding to the increase in number of outcomes) then we would have on average 0.002302 seconds per instance i.e. a little more than two milliseconds. Indeed, if we could further subdivide each of the 1024 intervals into 16 intervals and still expect a running time less than a second i.e. we would have a discretization 16384×16384 – corresponding to $k = 14$ – and expect to use approximately $0.562 \cdot 2^{2(k-4)}/10^6 = 0.5892$ seconds.

6 Final remarks

In this paper we have obtained a strengthening of first order dominance for the general multivariate case. Furthermore, we have described two algorithms for checking first order dominance in the bivariate case, one of which has linear time worst case complexity and is easy to implement.[14] It should be noted that when taking the setup times into account then it is not possible to obtain an algorithm which has sub linear worst case run time complexity. Hence the indirect algorithm presented in this paper is fast.

For numerical evaluations of *continuous* bivariate distributions the continuous distribution can be discretized into a finite set of outcomes, as illustrated in Section 5. By checking first order dominance of these constructed finite bivariate distributions we can obtain an approximate check on whether or not one first order dominates the other. Clearly, the more fine-grained this discretization is the better approximation. Thus it is important for empirical applications to have efficient methods for checking very fine-grained discretizations of continuous bivariate distributions.

The algorithms provided can easily be extended to the case of non-rectangular finite subsets of $\mathbb{R}^2$. This is done by extending the set of outcomes to the smallest rectangular envelopment of the given set, and then put $f(x) = g(x) = 0$ for any element not contained in the original set.

It remains an open question whether or not it is possible to identify equally efficient – linear time worst case complexity – algorithms in the general multivariate case. That is, algorithms which are more efficient than setting up the corresponding transportation problem or max-flow problem and solving it as such.

One might speculate that the approach suggested in this paper can be used in stochastic optimization problems having stochastic dominance constraints (see for instance Noyan and Ruszczyński [30] and Dentcheva and Ruszczyński [8]). In particular optimization problems containing constraints on multivariate first order dominance may use the suggested approach. For example Noyan

[14] An implementation of Algorithm 2 in C++ is available from the authors. For practitioners it also easy to set up a spreadsheet applying Algorithm 2.

and Ruszczyński [30] suggests using an integer formulation for a problem having one-dimensional first order dominance constraints. We could augment such an approach to multivariate first order dominance using constraints stating that if first order dominance exists then a binary indicator should be equal to one. It would be possible to add our formulation (1)-(6) as a part of the problem, and then use a big-M constraint to set the variable. Another approach would be to apply Benders' combinatorial cuts, see Codato and Fischetti [5], where a feasibility problem (in our case problem (1)-(6) or Algorithm 2 if the problem is bivariate) is solved and if it is infeasible, then a cut separating the solution is added in order to remove the current solution. It may indeed also augment the two stage stochastic program suggested by Drapkin and Schultz [9].

A relation between first order dominance and chance constraints exists, where a target distribution has to be satisfied. Dentcheva [6] gives an example where a multiple number of chance constraints have to be satisfied yielding a target distribution. This is equivalent to first order dominance and it can be augmented into a multivariate setting. Viewing the chance constraints in a first order dominance context may improve the methods of e.g. Dentcheva et al. [7] and Kogan and Lejeune [20] where cuts can be separated using the Benders decomposition as hinted above. This is, however, beyond the scope of this paper and we have left it for future research.

Acknowledgements

We are grateful to Bettina Klinz, the two anonymous referees, and the associate editor for valuable comments and suggestions. Furthermore, we would like to thank participants of the 12th International Conference on Computational Management Science, 2015, in Prague as well as the participants of the Conference on Economic Design 2015, Istanbul, July 2015, for helpful discussions.

A Proofs

In this appendix the proofs of the lemmas and theorems of the paper are given. In addition two auxiliary lemmas – Lemma A1 and Lemma A2 – are given. The proofs are arranged in the order their respective lemmas and theorems appear in the paper with the exception of Lemma A1 and Lemma A2 which are placed just prior to their use.

Lemma A1 *Let* $(\mathbf{z}, \mathbf{c}, \mathbf{d})$ *be a feasible solution to problem* (1)-(6). *If* $\sum_{\mathbf{p}\in P} c_{\mathbf{p}} + \sum_{\mathbf{r}\in R} d_{\mathbf{r}} = 0$*, then*

1. $c_{\mathbf{p}} = 0$ *for all* $\mathbf{p} \in P$,
2. $d_{\mathbf{r}} = 0$ *for all* $\mathbf{r} \in R$,
3. *constraints* (2) *and* (3) *will be binding.*

Proof of Lemma A1. Let $(\mathbf{z}, \mathbf{c}, \mathbf{d})$ be a feasible solution to problem (1)-(6) with $\sum_{\mathbf{p}\in P} c_{\mathbf{p}} + \sum_{\mathbf{r}\in R} d_{\mathbf{r}} = 0$. As $\sum_{\mathbf{p}\in P} c_{\mathbf{p}} + \sum_{\mathbf{r}\in R} d_{\mathbf{r}} = 0$, and both $\mathbf{c} \geq \mathbf{0}$ and $\mathbf{d} \geq \mathbf{0}$, we must have that $c_{\mathbf{p}} = 0$ for all $\mathbf{p} \in P$ and $d_{\mathbf{r}} = 0$ for all $\mathbf{r} \in R$ showing parts 1 and 2 of the lemma. Part 3 of the lemma can be realized as follows: For each pair $(\mathbf{p}, \mathbf{r}) \in C$ the variable $z_{\mathbf{pr}}$ is present in exactly one of the constraints (2), and the corresponding coefficient is equal to one. The same holds for constraint (3). Thus, we have the relation

$$\sum_{\mathbf{p}\in P} \sum_{\mathbf{r}\in L(\mathbf{p})\cap R} z_{\mathbf{pr}} = \sum_{\mathbf{r}\in R} \sum_{\mathbf{p}\in U(\mathbf{r})\cap P} z_{\mathbf{pr}} \tag{15}$$

Hence, summarizing constraint (2) yields the following

$$\begin{aligned} \sum_{\mathbf{p}\in P} (s(\mathbf{p}) - c_{\mathbf{p}}) &\leq \sum_{\mathbf{p}\in P} \sum_{\mathbf{r}\in L(\mathbf{p})\cap R} z_{\mathbf{pr}} \\ &= \sum_{\mathbf{r}\in R} \sum_{\mathbf{p}\in U(\mathbf{r})\cap P} z_{\mathbf{pr}} \\ &\leq \sum_{\mathbf{r}\in R} (-s(\mathbf{r}) + d_{\mathbf{r}}) . \end{aligned}$$

Note that, as $\sum_{\mathbf{x}\in X} s(\mathbf{x}) = \sum_{\mathbf{x}\in X} f(\mathbf{x}) - \sum_{\mathbf{x}\in X} g(\mathbf{x}) = 0$, we have that $\sum_{\mathbf{p}\in P} s(\mathbf{p}) = \sum_{\mathbf{r}\in R} -s(\mathbf{r})$. Consequently, the above has to hold with equality when $c_{\mathbf{p}} = 0$ and $d_{\mathbf{r}} = 0$. When $c_{\mathbf{p}} = 0$ we have that $\sum_{\mathbf{p}\in P} s(\mathbf{p}) = \sum_{\mathbf{p}\in P} \sum_{\mathbf{r}\in L(\mathbf{p})\cap R} z_{\mathbf{pr}}$. Now suppose that for an element $\mathbf{p}' \in P$ we have that $s(\mathbf{p}') < \sum_{\mathbf{r}\in L(\mathbf{p}')\cap R} z_{\mathbf{p}'\mathbf{r}}$. Then some other $\mathbf{p}'' \in P$ must exist having $s(\mathbf{p}'') > \sum_{\mathbf{r}\in L(\mathbf{p}'')\cap R} z_{\mathbf{p}''\mathbf{r}}$, i.e. requiring that $c_{\mathbf{p}''} > 0$, which forces the objective to be positive. Hence, for the objective to have value zero we therefore must have $s(\mathbf{p}) = \sum_{\mathbf{r}\in L(\mathbf{p})\cap R} z_{\mathbf{pr}}$ for all $\mathbf{p} \in P$. Therefore, constraint (2) must be binding for an optimal solution to have value zero. An analogous argument can be made for (3), and consequently both constraints (2) and (3) have to be binding. ■

Proof of Lemma 1. If f first order dominates g, then a finite sequence of diminishing transfers exists. Consequently, a feasible set of transfers between elements of C exists yielding a solution value of zero for problem (1)-(6). Then directly by Lemma A1 we have that (7) and (8) hold. On the other hand, if $\mathbf{z} \geq \mathbf{0}$ exists such that (7) and (8) hold, then it is a feasible solution for problem (1)-(6) with $\mathbf{c} = \mathbf{0}$ and $\mathbf{d} = \mathbf{0}$. The values of $z_{\mathbf{pr}}$ then constitute a finite sequence of diminishing transfers. ■

Proof of Theorem 1. Clearly if we can obtain f from g by at most $|P| + |R|$ diminishing transfers then we have a finite number of diminishing transfers and, consequently, we have that f first order dominates g.

Now assume that f first order dominates g. From Lemma 1 we have that the set of feasible diminishing transfers is

$$\mathbf{Z} = \left\{ \mathbf{z} \in \mathbb{R}^{|C|} \;\middle|\; \begin{array}{rl} \sum\limits_{\mathbf{r} \in L(\mathbf{p}) \cap R} z_{\mathbf{pr}} = & f(\mathbf{p}) - g(\mathbf{p}),\ \forall \mathbf{p} \in P, \\ \sum\limits_{\mathbf{p} \in U(\mathbf{r}) \cap P} z_{\mathbf{pr}} = & g(\mathbf{r}) - f(\mathbf{r}),\ \ \forall \mathbf{r} \in R \\ \mathbf{z} \geq \mathbf{0} & \end{array} \right\} \tag{16}$$

and as f first order dominates g we have that $\mathbf{Z} \neq \emptyset$. The set of feasible diminishing probability transfers $\mathbf{Z}$ is a convex polytope, as it is the intersection of finitely many close half spaces. Furthermore, $\mathbf{Z}$ is bounded because variables can only have non-negative values and each variable is included in at least one constraint having only positive coefficients and positive right-hand side. If the constraint only has this variable, then the variable is fixed. If the constraints has more than one variable, then if we increase one variable it is necessary to decrease another variable, and this can only be continued until the other variable becomes zero. Hence, $\mathbf{Z}$ is a bounded convex polytope.

Any element of $\mathbf{Z}$ corresponds to a finite number of diminishing transfers. In particular, as $\mathbf{Z}$ is a bounded polytope we know, see e.g. Minoux [26], that each extreme point corresponds to at least one basis dividing the variables into basic variables, which can attain positive variables, and non-basic variables, which are fixed at their lower bound of zero. The number of basic variables is equal to the number of constraints i.e. at most $|P| + |R|$ variables can be positive. Thereby, at least one element of $\mathbf{Z}$ uses at most $|P| + |R|$ positive variable values corresponding to at most $|P| + |R|$ diminishing transfers.

Finally, due to (15) we have that the constraints are linear dependent and, as a consequence, the basic solutions will be degenerate and have at most $|P|+|R|-1$ positive variable values. Therefore at most $|P| + |R| - 1$ diminishing transfers are necessary. ■

Proof of Lemma 2. If the two solutions have $\mathbf{c}' = \overline{\mathbf{c}}$ and $\mathbf{d}' = \overline{\mathbf{d}}$, then they have the same objective value. Thus, we have to show that altering $\overline{\mathbf{z}}$ to $\mathbf{z}'$ maintains $\mathbf{c}' = \overline{\mathbf{c}}$ and $\mathbf{d}' = \overline{\mathbf{d}}$. For both $\mathbf{x}$ and $\mathbf{y}$ we have added and subtracted β in constraint (2), thus not changing the values $c_{\mathbf{x}}$ and $c_{\mathbf{y}}$. Furthermore, we have for both $\mathbf{v}$ and $\mathbf{w}$ added and subtracted β in the constraint (3) thereby not changing $d_{\mathbf{v}}$ or $d_{\mathbf{w}}$ either. Consequently, as solution $\overline{\mathbf{z}}$ was feasible so will $\mathbf{z}'$ be, and they will have the same objective value. ■

Lemma A2 *Let $X = X_1 \times X_2$. Given four elements $\mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{w} \in X$ having $\mathbf{w} \in L(\mathbf{x})$ and $\mathbf{v} \in L(\mathbf{y})$, then if $x_1 \leq y_1$ and $v_2 \geq w_2$, then $\mathbf{w} \in L(\mathbf{y})$.*

Proof of Lemma A2. This can be realized as follows: As $\mathbf{w} \in L(\mathbf{x})$ then $w_1 \leq x_1$, and by assumption we have that $x_1 \leq y_1$. Furthermore, we have that $\mathbf{v} \in L(\mathbf{y})$, then $v_2 \leq y_2$, and by assumption we have that $w_2 \leq v_2$. Hence, $\mathbf{w} \in L(\mathbf{y})$. ■

Proof of Lemma 3. We show that if a $\mathbf{z} \geq \mathbf{0}$ exists satisfying (7) and (8) of Lemma 1, then it can always be transformed into the solution found by Algorithm 1 without violating any of the constraints (7) and (8). Consequently, if the Algorithm 1 terminates without identifying a feasible sequence of diminishing transfers, then no such sequence exists.

Note that Lemma 2 provides the possibility of shifting between alternative solutions, all satisfying the sets of equations (7) and (8). Thus we will use this lemma for the transformation.

Due to the update in step 3 the algorithm will never transfer more probability mass away from $\mathbf{p} \in P$ than $s(\mathbf{p})$ and, likewise, it will never transfer more probability mass to $\mathbf{r} \in R$ than $-s(\mathbf{r})$. Furthermore, due to the selection of pairs $(\mathbf{p}, \mathbf{r})$ in step 1 and 2, we guarantee that $\sum_{\mathbf{r} \in L(\mathbf{p}) \cap R} z_{\mathbf{pr}} \leq s(\mathbf{p})$ for all $\mathbf{p} \in P$ and $\sum_{\mathbf{p} \in U(\mathbf{r}) \cap P} z_{\mathbf{pr}} \leq -s(\mathbf{r})$ for all $\mathbf{r} \in R$ at any point in Algorithm 1.

Let $\mathbf{z}^0 \geq \mathbf{0}$ satisfy equations (7) and (8) and suppose that it is different from the transfers performed by Algorithm 1. Now selecting elements $\mathbf{p} \in P$ in increasing $(1,2)$-order and corresponding elements $\mathbf{r} \in R \cap L(p)$ in decreasing $(2,1)$-order, according to step 1 and step 2, respectively. Then, as $\mathbf{z}^0$ is different from the transfers made by Algorithm 1, a first pair of elements $(\mathbf{p}^0, \mathbf{r}^0)$ exists such that $z^0_{(\mathbf{p}^0,\mathbf{r}^0)}$ is different from what is transferred by the algorithm. Recall that Algorithm 1 is greedy and therefore transfers as much probability mass as possible as soon as possible. Hence, $z^0_{(\mathbf{p}^0,\mathbf{r}^0)}$ is strictly less than the amount transferred by the algorithm. Otherwise the algorithm would have transferred more from $\mathbf{p}^0$ to $\mathbf{r}^0$. As the algorithm can transfer more than $z^0_{(\mathbf{p}^0,\mathbf{r}^0)}$ then we know that $z^0_{(\mathbf{p}^0,\mathbf{r}^0)} < s(\mathbf{p}^0)$ and $z^0_{(\mathbf{p}^0,\mathbf{r}^0)} < -s(\mathbf{r}^0)$. But as $\mathbf{z}^0$ satisfy both equations (7) and (8) we must have that at least one $\mathbf{p}'$ with $o_{12}(\mathbf{p}') > o_{12}(\mathbf{p}^0)$ must exist transferring $z^0_{\mathbf{p}'\mathbf{r}^0} > 0$ from $\mathbf{p}'$ to $\mathbf{r}^0$. In the following we let $\mathbf{p}' \neq \mathbf{p}^0$ be the minimal $(1,2)$-order element of P, where $z^0_{\mathbf{p}'\mathbf{r}^0} > 0$. Similarly, an $\mathbf{r}'$ with $o_{21}(\mathbf{r}') < o_{21}(\mathbf{r}^0)$ must exist transferring $z^0_{\mathbf{p}^0\mathbf{r}'} > 0$ from $\mathbf{p}^0$ to $\mathbf{r}'$. We let $\mathbf{r}' \neq \mathbf{r}^0$ be the maximal $(2,1)$-order element having $z^0_{\mathbf{p}^0\mathbf{r}'} > 0$.

Now observe that, as $o_{12}(\mathbf{p}^0) < o_{12}(\mathbf{p}')$ we have that $p^0_1 \leq p'_1$. Furthermore, as $o_{21}(\mathbf{r}^0) > o_{21}(\mathbf{r}')$ we have that $r^0_2 \geq r'_2$. Therefore by Lemma A2 we have that $\mathbf{r}^0, \mathbf{r}' \in L(\mathbf{p}^0) \cap L(\mathbf{p}') \cap R$. By Lemma 2 we can put

$$\beta = \min\{z^0_{\mathbf{p}'\mathbf{r}^0}, z^0_{\mathbf{p}^0\mathbf{r}'}\} \tag{17}$$

and construct the new vector $\mathbf{z}^1 \geq \mathbf{0}$ having all elements equal to $\mathbf{z}^0$ except for

$$\begin{aligned} z^1_{\mathbf{p}'\mathbf{r}^0} &= z^0_{\mathbf{p}'\mathbf{r}^0} & -\beta \\ z^1_{\mathbf{p}^0\mathbf{r}'} &= z^0_{\mathbf{p}^0\mathbf{r}'} & -\beta \\ z^0_{\mathbf{p}^0\mathbf{r}^0} &= z^1_{\mathbf{p}^0\mathbf{r}^0} & +\beta \\ z^0_{\mathbf{p}'\mathbf{r}'} &= z^1_{\mathbf{p}'\mathbf{r}'} & +\beta \end{aligned}$$

which satisfy (7) and (8). Consequently, $\mathbf{z}^1$ still constitutes a finite number of diminishing transfers.

By putting $\mathbf{z}^0 = \mathbf{z}^1$ and repeating the argument above we will gradually get the transfers corresponding to those identified by Algorithm 1. Thus, if $\mathbf{z} \geq \mathbf{0}$ exists satisfying (7) and (8), then it is possible to transform the vector $\mathbf{z}$ into the vector obtained by Algorithm 1 without violating (7) and (8).

■

Proof of Theorem 2. If Algorithm 1 terminates in step 1, then a sequence of diminishing transfers has been found satisfying all of the constraints (2) and (3). This has been done such that all $c_{\mathbf{p}}$ values have been decreased to zero while keeping the $d_{\mathbf{r}}$ values at zero. Hence, a finite sequence of diminishing transfers exists.

We need to show that if Algorithm 1 terminates in step 2, then no feasible sequence of diminishing transfers exists. From Lemma 3 we know that if a feasible set of diminishing transfers exists then Algorithm 1 will obtain a feasible set of diminishing transfers. Thus, if f first order dominates g then it is always possible to obtain a finite number of diminishing transfers by Algorithm 1, and therefore if Algorithm 1 fails to identify such a finite number of diminishing transfers, then no such finite number of diminishing transfers exists.

Algorithm 1 has a time complexity of $O(n^2)$, as for each element $\mathbf{p} \in P$ we have to search through the elements of $L(\mathbf{p})$ to identify a suitable element $\mathbf{r} \in L(\mathbf{p}) \cap R$. ■

Proof of Lemma 4. We prove this by showing that the sequence of diminishing transfers obtained by Algorithm 1 can be obtained by Algorithm 2 as well. Algorithm 2 traverses the elements in increasing $(1, 2)$-order and therefore encounters elements of P in the same order as Algorithm 1. We need to keep track of the actual transfers. For this we use two sets of ordered lists which are updated during step 1 for element (i, j) in Algorithm 2.

For $i \in X_1$ we let $\Theta_i = (\mathbf{p}^1, \dots, \mathbf{p}^k)$ be the elements of $\{\mathbf{p} \in S_{ij} \cap P | c_{\mathbf{p}} > 0\}$ ordered such that $o_{12}(\mathbf{p}^a) < o_{12}(\mathbf{p}^{a+1})$ for $a = 1, \dots, k-1$. Hence, the $(1, 2)$-minimal element of Θ_i is the first element. All elements of $(x_1, x_2) \in \Theta_i$ have lower $(1, 2)$-order than any element of $(y_1, y_2) \in \Theta_{i+1}$ because $x_1 = i < i + 1 = y_1$.

For $j \in X_2$ we let $\Delta_j = \{\mathbf{r}^1, \dots, \mathbf{r}^h\}$ be the elements of $\{\mathbf{r} \in T_{ij} \cap R | \rho_{\mathbf{r}} > 0\}$ ordered such that $o_{21}(\mathbf{r}^a) > o_{21}(\mathbf{r}^{a+1})$ for $a = 1, \dots, h-1$. Hence the first element of Δ_j is the $(2, 1)$-maximal element of Δ_j. Any element of $(x_1, x_2) \in \Delta_j$ has lower $(2, 1)$-order than any element of $(y_1, y_2) \in \Delta_{j+1}$ because $x_2 = j < j + 1 = y_2$. Hence decreasing $j + 1$ to j yields lower $(2, 1)$-order elements.

Step 1 of algorithm is now augmented to encompass the update of the two lists. First, if $s_{ij} > 0$ then (i, j) is added to the end of Θ_i, as $(i, j) \in P$ with $c_{(i,j)} > 0$ and it has higher $(1, 2)$-order than the other elements of Θ_i. On the other hand if $s_{ij} < 0$ the (i, j) is added to the end of Δ_j as it has lower $(2, 1)$-order than the other elements of Δ_j and $(i, j) \in R$ with $\rho_{(i,j)} > 0$. For simplicity we let τ_{ij} be the amount transferred from S_{ij} to T_{ij} and we initialize $\tau_{ij} = 0$. Then the following is repeated until either Θ_i is empty or Δ_j is empty:

1. Let $\mathbf{p}$ be the first element of Θ_i and $\mathbf{r}$ be the first element of Δ_j.
2. Put
$$\begin{aligned} z_{\mathbf{pr}} &= \min\{c_{\mathbf{p}}, \rho_{\mathbf{r}}\} \\ c_{\mathbf{p}} &= c_{\mathbf{p}} - z_{\mathbf{pr}} \\ \rho_{\mathbf{r}} &= \rho_{\mathbf{r}} - z_{\mathbf{pr}} \\ \tau_{ij} &= \tau_{ij} + z_{\mathbf{pr}} \end{aligned}$$
3. If $c_{\mathbf{p}} = 0$ then remove $\mathbf{p}$ from Θ_i and if $\rho_{\mathbf{r}} = 0$ then remove $\mathbf{r}$ from Δ_j.

The sequence of which the pairs $(\mathbf{p}, \mathbf{r})$ are selected is equivalent to the sequence in Algorithm 1. This is due to selecting $\mathbf{p}$ as the first of the elements in Θ_i and thereby selecting these in increasing $(1, 2)$-order. Furthermore, selecting $\mathbf{r}$ as the first element of Δ_j corresponds to selecting the $(2, 1)$-maximal element within $L(i, j) \cap R$. We also update the diminishing transfer values, $z_{\mathbf{pr}}$, in the exact same sequence as for Algorithm 1.

Finally we have to address the correspondence between the amount transferred and the values u_{ij} and e_{ij}. We have two cases. The first case is when $s_{ij} \geq 0$. Then the transferred amount $\tau_{ij} = \min\{u_{i,j+1} + s_{ij}; e_{i-1,j}\}$. If $\tau_{ij} = e_{i-1,j}$ then $e_{ij} = 0$ and the corresponding list Δ_j is empty, and $u_{ij} = u_{i,j+1} + s_{ij} - e_{i-1,j}$. On the other hand if $\tau_{ij} = u_{i,j+1} + s_{ij}$ then $u_{ij} = 0$ and the corresponding list Θ_i is empty, and $e_{ij} = e_{i-1,j} - u_{i,j+1} - s_{ij}$. In the second case we have that $s_{ij} < 0$, and the amount transferred is $\tau_{ij} = \min\{u_{i,j+1}; e_{i-1,j} - s_{ij}\}$. If $\tau_{ij} = u_{i,j+1}$ then $u_{ij} = 0$, with Θ_i being empty, and $e_{ij} = e_{i-1,j} - s_{ij} - u_{u,j+1}$. If $\tau_{ij} = e_{i-1,j} - s_{ij}$ then $e_{ij} = 0$ and $u_{ij} = u_{i,j+1} - e_{i-1,j} + s_{ij}$. Each of the two cases corresponds to putting

$$\begin{aligned} u_{ij} &= \max\{0; u_{i,j+1} - e_{i-1,j} + s_{ij}\} = \max\{0; t_{ij}\} \\ e_{ij} &= \max\{0; e_{i-1,j} - u_{i,j+1} - s_{ij}\} = \max\{0; -t_{ij}\} \end{aligned}$$

which is exactly the values calculated in step 1 of Algorithm 2. Thus τ_{ij} provides the connection between $z_{\mathbf{pr}}$ and the two values u_{ij} and e_{ij}.

Note that when $u_{ij} = 0$ then the list Θ_i is empty i.e. all elements of $\mathbf{p} \in S_{ij} \cap P$ have sent $s(\mathbf{p})$ probability mass to elements of $L(\mathbf{p}) \cap R$. Consequently, if Algorithm 2 terminates with $u_{n_1,1} = 0$, then all lists Θ_i, for $i \in X_1$, are empty, corresponding to the case where all elements $\mathbf{p} \in P$ have sent $s(\mathbf{p})$ probability mass to elements of $L(\mathbf{p}) \cap R$. Therefore a finite sequence of diminishing transfers exists showing that g can be obtained from f. ■

Proof of Lemma 5. We can explicitly identify a lower comprehensive set which violates (B) if the value of $u_{i1} > 0$ for some $i \in X_1$. First, note that

$$u_{ij} - e_{ij} = \max\{0, t_{ij}\} - \max\{0, -t_{ij}\} = t_{ij} = u_{i,j+1} - e_{i-1,j} + s_{ij}$$

and suppose that we are given a lower comprehensive set Υ. Then we have

$$\sum_{(i,j)\in\Upsilon} (u_{ij} - e_{ij}) = \sum_{(i,j)\in\Upsilon} (u_{i,j+1} - e_{i-1,j}) + \sum_{(i,j)\in\Upsilon} s_{ij} \tag{18}$$

Now let

$$\begin{aligned} H &= \{ \quad i \in X_1 & &| \quad (i,1) \in \Upsilon\} \\ I &= \{ \quad j \in X_2 & &| \quad (1,j) \in \Upsilon\} \\ J &= \{ \quad (i,j) \in \Upsilon & &| \quad (i+1,j) \notin \Upsilon \quad \} \\ K &= \{ \quad (i,j) \notin \Upsilon & &| \quad (i,j-1) \in \Upsilon \quad \} \end{aligned}$$

The sets J and K are illustrated in Figure 9, where Υ is the gray area (including both shades of gray). The union of the hatched boxes corresponds to J, whereas the union of the dark gray boxes is the set K. Furthermore, the dashed box is the elements $(i,1) \in \Upsilon$ with $i \in H$, and the dotted box is the elements $(1,j)$ with $j \in I$. We can then rearrange (18) as

$$\sum_{(i,j)\in\Upsilon} s_{ij} = \sum_{j\in I} e_{0j} - \sum_{(i,j)\in J} e_{ij} + \sum_{h\in H} u_{h1} - \sum_{(i,j)\in K} u_{ij} \tag{19}$$

where $\sum_{j\in I} e_{0j} = 0$ by the definition of e_{0j}. Showing that the lower comprehensive set Υ violates condition (B) corresponds to showing that $\sum_{(i,j)\in\Upsilon} s_{ij} > 0$, which is equivalent to showing that $\sum_{h\in H} u_{h1} > \sum_{(i,j)\in J} e_{ij} + \sum_{(i,j)\in K} u_{ij}$.

Suppose that the Algorithm 2 terminates with $u_{i1} > 0$. Then we know that $u_{h1} = 0$ for $h = 1, \ldots, i-1$ and therefore $\sum_{h\in H} u_{h1} = u_{i1}$. Hence, if we can construct the lower comprehensive set Y such that $\sum_{(h,j)\in J} e_{hj} + \sum_{(h,j)\in K} u_{hj} = 0$ then we have the violation we are seeking. We construct Y implicitly by constructing J and K explicitly. Each time an element (h,j) is added to J, then all elements (a,j) with $a \leq h$ are added to Y. Start with $(h,j) = (i,1)$. Because $u_{i1} > 0$, we have that $e_{i1} = 0$. Therefore add $(i,1)$ to J. Repeat the following until $h = 0$. If $u_{h,j+1} = 0$, add $(h,j+1)$ to K and put $h = h-1$, otherwise $u_{h,j+1} > 0$ and consequently $e_{h,j+1} = 0$ and therefore add $(h,j+1)$ to J and put $j = j+1$. When terminating we have only added elements to J having $e_{hj} = 0$ and elements to K

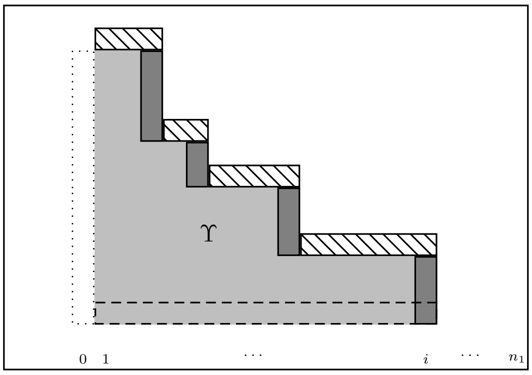

Fig. 9 A violated lower comprehensive set.

having elements $u_{hj} = 0$, thus having $\sum_{(i,j)\in J} e_{ij} + \sum_{(i,j)\in K} u_{ij} = 0 < u_{i1}$ showing that (B) is violated by Υ. ■

Proof of Theorem 3. If the algorithm terminates with $u_{n_1 1} = 0$, then we have by Lemma 4 and property (A) that f first order dominates g. On the other hand, if the algorithm terminates with $u_{i1} > 0$ for some $i \in X_1$, then by Lemma 5 a violated lower comprehensive set exists. Consequently, by property (B) f does not first order dominate g. Finally, as each element of X is traversed maximally once and the number of operations for each element is constant, the algorithm terminates in $O(n)$ iterations. ■

References

1. Arndt, C., Distante, R., Hussain, M. A., Østerdal, L. P., Huong, P. L., and Ibraimo, M. (2012). Ordinal welfare comparisons with multiple discrete indicators: A first order dominance approach and application to child poverty. *World Development*, 40:2290 – 2301.
2. Arndt, C., Hussain, M., Salvucci, V., Tarp, F., and Østerdal, L. P. (2013). Advancing small area estimation. WIDER Working Paper, No. 2013/053.
3. Atkinson, A. and Bourguignon, F. (1982). The comparison of multi-dimensioned distributions of economic status. *Review of Economics Studies*, 49:183–201.
4. Bawa, V. S. (1982). Stochastic dominance: A research bibliography. *Management Science*, 28(6):698–712.
5. Codato, G. and Fischetti, M. (2006). Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766.
6. Dentcheva, D. (2006). *Optimization Models with Probabilistic Constraints*, pages 49–97. Springer London, London.
7. Dentcheva, D., Prékopa, A., and Ruszczyński, A. (2000). Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89(1):55–77.
8. Dentcheva, D. and Ruszczyński, A. (2009). Optimization with multivariate stochastic dominance constraints. *Mathematical Programming*, 117(1):111–127.

9. Drapkin, D. and Schultz, R. (2010). An algorithm for stochastic programs with first-order dominance constraints induced by linear recourse. *Discrete Applied Mathematics*, 158(4):291 – 297. 6th Cologne/Twente Workshop on Graphs and Combinatorial Optimization (CTW 2007).
10. Dyckerhoff, R. and Mosler, K. (1997). Orthant orderings of discrete random vectors. *Journal of Statistical Planning and Inference*, 62:193–205.
11. Dykstra, R. L. and Robertson, T. (1982). An algorithm for isotonic regression for two or more independent variables. *The Annals of Statistics*, 10(3):pp. 708–716.
12. Grant, S., Kajii, A., and Polak, B. (1992). Many good choice axioms: when can many-good lotteries be treated as money lotteries? *Journal of Economic Theory*, 56:313–337.
13. Gravel, N. and Moyes, P. (2012). Ethically robust comparisons of bidimensional distributions with an ordinal attribute. *Journal of Economic Theory*, 147:1384–1426.
14. Hansel, G. and Troallic, J. (1978). Measures marginales et théorème de Ford-Fulkerson. *Zeitschrift für Wahrsheinlichkeitstheorie und verwandte Gebiete*, 43:245–251.
15. Harder, J. and Russell, W. (1974). Stochastic dominance in choice under uncertainty. In Balch, M., D.L., M., and Wu, S., editors, *Essays on Economic Behavior under Uncertainty*. North-Holland, Amsterdam.
16. Huang, C., Kira, D., and Vertinsky, I. (1978). Stochastic dominance rules for multi-attribute utility functions. *Review of Economic Studies*, 45:611–615.
17. Hussain, M. A., Jørgensen, M. M., and Østerdal, L. (2014). Refining population health comparisons: A multidimensional first order dominance approach. COHERE Working Paper, University of Southern Denmark.
18. Kamae, T., Krengel, U., and O'Brien, G. (1977). Stochastic inequalities on partially ordered spaces. *The Annals of Probability*, 5:899–912.
19. Kleinschmidt, P. and Schannath, H. (1995). A strongly polynomial algorithm for the transportation problem. *Mathematical Programming*, 68:1–13.
20. Kogan, A. and Lejeune, M. A. (2014). Threshold boolean form for joint probabilistic constraints with random technology matrix. *Mathematical Programming*, 147(1):391–427.
21. Lehmann, E. (1955). Ordered families of distributions. *The Annals of Mathematical Statistics*, 26:399–419.
22. Levhari, D., Paroush, J., and Peleg, B. (1975). Efficiency analysis for multivariate distributions. *Review of Economic Studies*, 42:87–91.
23. Levy, H. and Paroush, J. (1974). Toward multivariate efficiency criteria. *Journal of Economic Theory*, 7:129–142.
24. Marshall, A. and Olkin, I. (1979). *Inequalitites: Theory of Majorization and Its Applications*, volume 143 of *Mathematics in Science and Engineering*. New York: Academic Press.
25. Meyer, M. and Strulovici, B. (2015). Beyond correlation: Measuring interdependence through complementarities. Unpublished working paper.
26. Minoux, M. (1986). *Mathematical programming: theory and algorithms*. Wiley-Interscience series in discrete mathematics and optimization. Wiley.
27. Mosler, K. (1984). Stochastic dominance decision rules when the attributes are utility independent. *Management Science*, 30:1311–1322.
28. Mosler, K. and Scarsini, M. (1991). Some theory of stochastic dominance. In Mosler, K. and Scarsini, M., editors, *Stochastic Orders and Decision under Risk*. Institute of Mathematical Statistics, Heyward, California.

29. Müller, A. and Stoyan, D. (2002). *Comparison Methods for Stochastic Models and Risks*. John Wiley and Sons.

30. Noyan, N. and Ruszczyński, A. (2008). Valid inequalities and restrictions for stochastic programming problems with first order stochastic dominance constraints. *Mathematical Programming*, 114(2):249–275.

31. Orlin, J. B. (2013). Max flows in o(nm) time, or better. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing*, STOC '13, pages 765–774, New York, NY, USA. ACM.

32. Preston, C. (1974). A generalization of the FKG inequalities. *Communications in Mathematical Physics*.

33. Russell, W. and Seo, T. (1978). Ordering uncertain prospects: The multivariate utility functions case. *Review of Economic Studies*, 45:605–610.

34. Sampson, A. R. and Whitaker, L. R. (1988). Positive dependence, upper sets, and multidimensional partitions. *Mathematics of Operations Research*, 13(2):pp. 254–264.

35. Scarsini, M. (1988). Dominance conditions for multivariate utility functions. *Management Science*, 34:454–460.

36. Schrijver, A. (1987). *Theory of Linear and Integer Programming*. Discrete Mathematics and Optimization. Wiley-Interscience.

37. Shaked, M. and Shanthikumar, J. (2007). *Stochastic Orders*. Springer.

38. Silvapulle, M. J. and Sen, P. K. (2011). *Constrained statistical inference: Order, inequality, and shape constraints.*, volume 912. Wiley-Interscience.

39. Sriboonchita, S., Dhompongsa, S., Wong, W. K., and Nguyen, H. T. (2009). *Stochastic dominance and applications to finance, risk and economics*. Chapman & Hall/CRC.

40. Strassen, V. (1965). The existence of probability measures with given marginals. *The Annals of Mathematical Statistics*, 36:423439.

41. Østerdal, L. P. (2010). The mass transfer approach to multivariate discrete first order stochastic dominance: direct proof and implications. *Journal of Mathematical Economics*, 46:1222–1228.