JOURNAL OF ALGORITHMS **16**, 218–233 (1994)

A Polynomial Time Algorithm for Unidimensional Unfolding Representations*

JEAN-PAUL DOIGNON

Université Libre de Bruxelles, c.p. 216, Bd du Triomphe, B-1050 Bruxelles, Belgium

AND

JEAN-CLAUDE FALMAGNE

School of Social Sciences, University of California, Irvine, California 92717

Received July 23, 1991; accepted May 4, 1993

Two conditions on a collection of simple orders—unimodality and straightness—are necessary but not jointly sufficient for unidimensional unfolding representations. From the analysis of these conditions, a polynomial time algorithm is derived for the testing of unidimensionality and for the construction of a representation when one exists.

Suppose that the objects forming a finite set X are ranked according to personal preference by the subjects from a finite population A. Without loss of generality, it is assumed that the (simple) orders $\leq_a$ thus produced by the subjects a from A are all distinct. Then $\mathscr{M} = (X, \{\leq_a \mid a \in A\})$ constitutes a (*multipreference*) *system*. We always set $m = |A|$ and $n = |X|$, where $m \geq 1$ and $n \geq 2$.

The unfolding model of Coombs [5] (see also Falmagne and Doignon [6]) offers an explanation of such data on the basis of relative proximity in some embedding "psychological" space. Formally, a metric space (E, d), and two mappings $f: A \to E$ and $g: X \to E$ are sought, such that for all

*This work is supported by Grant BNS 8911-339 to Jean-Claude Falmagne at the University of California, Irvine. Jean-Paul Doignon acknowledges invitations to the Irvine Research Unit in Mathematical Behavioral Sciences during August 1990 and August 1991. The authors are grateful to Jean-Pierre Barthélemy and Bernard Monjardet for having brought their attention to important references.

0196-6774/94 $6.00

$a \in A$ and $i, j \in X$:

$$i \leq_a j \quad \text{iff} \quad d(f(a), g(i)) \leq d(f(a), g(j)).$$

In this situation, the pair (f, g) of mappings is a *representation* of the system $\mathscr{M}$ in the metric space (E, d).

Any system $\mathscr{M}$ can be represented in some Euclidean space $\mathbb{E}^r$ of dimension r; this is easily seen by taking r sufficiently large (that is, $r = n - 1$). Several computer programs are available for the tentative construction of Euclidean representations in lower dimensions (for a recent survey, see, e.g., Borg and Lingoes [4]). However, it seems that the following fundamental question remains unanswered ($|p - q|$ denotes the Euclidean distance between points p, q).

Question U. Given a natural number r, find necessary and sufficient conditions on a system $\mathscr{M} = (X, \{\leq_a | a \in A\})$ for the existence of two mappings $f: A \to \mathbb{E}^r$ and $g: X \to \mathbb{E}^r$ such that for all $a \in A$ and $i, j \in X$,

$$i \leq_a j \quad \text{iff} \quad |f(a) - g(i)| \leq |f(a) - g(j)|.$$

When such an equivalence holds, the system $\mathscr{M}$ satisfies *r-unfolding*, or equivalently, is an *r-unfolding system*. The pair (f, g) is an *r-representation* of $\mathscr{M}$.

Our paper is devoted to the case $r = 1$, that is, to *one-unfolding systems*, or *unidimensional* (*unfolding*) *systems*. Let us collect here a few obvious facts, without proof.

LEMMA 1. *Any unidimensional system $\mathscr{M}$ has a one-representation (f, g) in $\mathbb{R}$ such that, for all $a \in A$ and all distinct $i, j, k, l \in X$:*

(i) $2f(a) \neq g(i) + g(j)$;

(ii) $g(i) + g(j) \neq g(k) + g(l)$;

(iii) $2g(k) \neq g(i) + g(j)$;

(iv) *$f(a)$ and $g(i)$ are rational numbers.*

Moreover, we always have $m \leq \binom{n}{2} + 1$. If $m < \binom{n}{2} + 1$, the set A can be enlarged to a set A', with $|A'| = \binom{n}{2} + 1$, so as to form a unidimensional system $(X, \{\leq'_a | a \in A'\})$ extending $(X, \{\leq_a | a \in A\})$ in the sense that $\leq'_a = \leq_a$ for any $a \in A$.

The value $\binom{n}{2} + 1$ results from the existence of exactly $\binom{n}{2}$ midpoints of pairs of represented objects. It motivates the following definition, which is adequate only in the case of unidimensional unfolding.

DEFINITION 2. A unidimensional system $\mathscr{M}$ with $m = \binom{n}{2} + 1$ is said to be *full*.

We investigate two necessary conditions for unidimensionality, that we call "unimodality" and "straightness." As shown by Coombs [5, p. 91], their conjunction does not, in general, imply unidimensionality. From the analysis of these conditions, however, we derive an algorithm for testing unidimensionality and constructing a representation when one exists. In the full case, our paper merely restates Coombs' results in an algorithmic framework. In the general case, we obtain new results.

1. UNIMODALITY

When a system $\mathscr{M}$ is represented on the Euclidean line, its set X of objects inherits two opposite orders from the ordering of points on the line. Each preference order $\leq_a$, for $a \in A$, has a special property with respect to these two orders.

DEFINITION 3. A simple order $\leq_b$ on X is *unimodal with respect* to some order $\leq_o$ on X when for all $i, j, k \in X$:

$$i \leq_o j \leq_o k \Rightarrow j \leq_b i \quad \text{or} \quad j \leq_b k.$$

A system $\mathscr{M} = (X, \{\leq_a | a \in A\})$ is *unimodal* if there exists some *reference order* $\leq_o$ on X such that each $\leq_a$ (for $a \in A$) is unimodal with respect to $\leq_o$.

PROPOSITION 4. *Any unidimensional system is unimodal.*

This is shown by taking as a reference order an order inherited from a one-representation. To discuss unimodality, we introduce some more terminology. The *open interval* $]i, k[_b$ with endpoints i, k in the order $\leq_b$ is

$$]i, k[_b = \begin{cases} \{j | i <_b j <_b k\} & \text{when } i <_b k, \\ \{j | k <_b j <_b i\} & \text{when } k <_b i, \\ \varnothing & \text{when } i = k. \end{cases}$$

The *closed interval* $[i, k]_b$ is obtained by adding i and k to $]i, k[_b$. The *beginning set* of i in $\leq_b$ is $]-\infty, i]_b = \{j \in X | j \leq_b i\}$. We denote by $\text{rk}_b i$ the rank of i in $\leq_b$ and by b_α the element of rank α in $\leq_b$ (with $1 \leq \alpha \leq n$).

A system $\mathscr{M}$ is encoded in a $m \times n$ array $\mathtt{M}$ as follows. First label arbitrarily the objects of X from 1 to n and select some ordering of the subjects of A. Then store the order $\leq_a$ of the ith subject in the ith row as the following sequence of numbers:

$$a_1 \quad a_2 \quad \ldots \quad a_n.$$

The proof of the next proposition is easy, and omitted.

PROPOSITION 5. *Given two orders* $\leq_a$ *and* $\leq_o$ *on* X, *the following three assertions are equivalent*:

(i) $\leq_a$ *is unimodal with respect to* $\leq_o$;

(ii) *each* i *from* X *is an extreme element in* $\leq_o$ *of the set* $\{j \in X | j \leq_a i\}$;

(iii) *any beginning set in* $\leq_a$ *is an interval in* $\leq_o$.

From Proposition 5, we derive an algorithm that tests whether a system $\mathscr{M}$ is unimodal with respect to a given reference order $\leq_o$. We consider the orders $\leq_a$ one at a time, and the elements $a_1, a_2, \ldots, a_n$ one after the other. By maintaining two pointers to the endpoints of the interval in $\leq_o$ formed by the set $\{a_1, a_2, \ldots, a_i\}$, we can easily test whether $\{a_1, a_2, \ldots, a_{i+1}\}$ also forms an interval in $\leq_o$. The resulting algorithm (similar to Romero [12, Section 2.4]) has execution time in $O(mn)$. (An elementary manipulation of natural numbers not larger than n, such as an assignment or comparison, counts for one step.)

When the reference order is not given, how could we test whether a system $\mathscr{M}$ is unimodal? Since a reference order never is unique (its opposite is again a reference order, and there may be other reference orders), the design of a good algorithm is less obvious. The question is also of interest in voting theory (Black [3]), because Condorcet effect of intransitive simple majority comparisons never occurs for a unimodal family of orders. Bartholdi and Trick [1] gave an algorithm, with running time in $O(mn^2)$, by referring to the consecutive one's property. In view of extensions to our unfolding problem, we design a direct algorithm with running time in $O((m + n)n)$.

Our analysis starts from Proposition 5(iii). Assume that the order $\leq_a$ is unimodal with respect to some reference order $\leq_o$. The beginning sets of $\leq_a$ having at least two objects become closed intervals in $\leq_o$; we call these intervals *boxes*. Inverting the order of all objects in a box results in another reference order for $\leq_a$. Moreover, all reference orders for $\leq_a$ are obtainable by repeating such inversions. Thus, an order on n objects is unimodal with respect to 2^{n-1} reference orders (Romero [12]). For the extension to a system $\mathscr{M}$, we need the following concept.

DEFINITION 6. Let $2 \leq \alpha \leq n$. Then α determines a *fault* in the system $\mathscr{M}$ if all beginning sets $\{a_1, a_2, \ldots, a_\alpha\}$, for $a \in A$, are equal. We also refer to this unique set as the *fault* (determined by α).

Note that n always determines a fault. In the setting of Coombs' model, a fault has a straightforward interpretation: any subject prefers any object in the fault over any object not in the fault.

PROPOSITION 7. *Assume that the system $\mathscr{M}$ is unimodal. Then its reference order is unique up to reversing iff there is only one fault in $\mathscr{M}$. In general, there are 2^f reference orders if f is the number of faults in $\mathscr{M}$.*

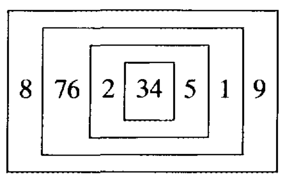
The proof of Proposition 7 will be based on Lemmas 10 and 11. The faults in a unimodal system $\mathscr{M}$ with reference order $\leq_o$ all become intervals in $\leq_o$, that we again call *boxes*. Inverting the order of objects in a box, i.e., "flipping" the box, transforms $\leq_o$ into another reference order for $\mathscr{M}$. More precisely,

Remark 8. All reference orders mentioned in Proposition 7 are easily generated from one of them by flipping some or all of the f boxes derived from the faults.

EXAMPLE 9. Suppose a unimodal system is encoded as the array

3	4	2	5	6	7	1	8	9
4	3	2	5	1	6	7	9	8
3	4	5	2	1	6	7	8	9

There are four faults. The 16 reference orders can be summarized by showing the boxes in one of them, e.g.,



(this suggests an easily implementable data structure for the set of reference orders).

LEMMA 10. *Suppose that two distinct objects i and j appear at rank n in some two of the orders of the unimodal system $\mathscr{M}$. Let us take any reference order $\leq_o$ for which $i <_o j$, and denote by S the transitive closure of the union of all $\leq_a$, $a \in A$. Then any object k such that iSk or jSk has its rank in $\leq_o$ determined by $\mathscr{M}$.*

Proof. First note that i must be the first element in $\leq_o$ (use Proposition 5(iii)). Assume that, say, $i <_c p <_b l <_a k$. Then $]-\infty, p]_c$ must become an interval in $\leq_o$ with p as an extremity and i as the other. Thus we can locate p, and, moreover, we know $[i, p]_o$ (as an unordered set).

Next, $]-\infty, l]_b$ must become an interval in $\leq_o$ that includes p and has l as one of its extremities. In both cases $p <_o l$ and $l <_o p$, we can locate l and in the same time compute $[i, l]_o$ (as an unordered set). The next step is similar: $]-\infty, k]_a$ becomes an interval that includes l; thus we can locate k.

The general situation is handled the same way; we remark that the case jSk becomes similar when $\leq_o$ is replaced with its opposite. □

LEMMA 11. *Assume that the system $\mathscr{M}$ has only one fault. If k is an object that appears with rank α in some order $\leq_a$, where $a \in A$ and $1 < \alpha < n$, and k never appears with greater rank in any of the $\leq_b$ for $b \in A$, then there is an object l having rank less than α in $\leq_a$ but greater than α in some of the orders $\leq_c$ with $c \in A$.*

We leave the proof of Lemma 11 and of Proposition 7 to the reader. Now to check a system $\mathscr{M}$ for unimodality, we can proceed as follows.

SKETCH OF THE ALGORITHM. The columns of the array `M` representing a system $\mathscr{M}$ will be visited from last to first. The reference order is being built in an array `ref` in such a way that the still non-assigned cells form an interval. Two pointers are kept to the extremities of this interval. Moreover, the ith component of an auxiliary array indexed by $1, 2, \ldots, n$ holds the value "undone," "left," or "right," thus indicating in which extremity of the unassigned interval (the label of) object i is assigned.

If the last column of `M` contains only one object x, this object is assigned to the rightmost empty cell of `ref`; the same with the next column, etc. When a column with more than one object is first detected, there are two cases. If the column contains more than two objects, then the system is not unimodal and the algorithm stops. Else the two objects are assigned respectively (and arbitrarily) to the leftmost and rightmost empty cells of `ref`.

The algorithm continues by visiting the remaining columns of `M`. When an object k is met for the first time, it is assigned to one of the two extremities of the non-assigned interval of `ref`. To decide which one, an object l as in Lemma 11 is sought. If none exists, then we have a fault at the rank of k, and we may assign k to any of the extremities. If l is found, then k must be allocated to the left extremity exactly in case l was assigned to the right.

When `ref` has been completely assigned, it remains to check that it encodes a reference order for $\mathscr{M}$; if not, unimodality fails. □

PROPOSITION 12. *The algorithm described above has running time in $O((m+n)n)$.*

Proof. During execution, all cells of `M` are visited. When a new object k is met, the beginning set it determines in one of the orders $\leq_a$ has to be scanned; note that this will happen less than n times. We have seen after Proposition 5 that unimodality with respect to a known order can be tested in $O(mn)$ steps. Thus the total number of steps is in $O(n^2 + mn)$. □

Remark 13. It is not difficult to amend the algorithm in such a way that it produces all the reference orders, using a data structure of the kind suggested in Example 9.

2. Straightness

For each pair $(g(i), g(j))$ of real numbers representing a pair of objects (i, j), consider its midpoint $(g(i) + g(j))/2$. Suppose that there is only one such midpoint between the representing points of two subjects a and b. Then their orders $\leq_a$ and $\leq_b$ only differ by the transposition of the two corresponding objects. Such a transposition will be called a *mutation*. Each mutation selects an unordered pair from X, which will also be referred to as the mutation.

A graph called the *permutohedron* is obtained by taking all the orders on X as vertices, and declaring two orders as *adjacent* whenever they differ by exactly one mutation. This graph happens to be the vertex–edge graph of an $(n - 1)$-polyhedron, also called permutohedron (Guilbaud and Rosenstiehl [8]; see also Feldman Högaasen [7] and Le Conte de Poly-Barbut [10]; Björner [2] has an extensive list of references). The distance between two orders on X, defined as half the cardinality of their symmetric difference, is also the distance between the corresponding two vertices on the permutohedron.

Here is another necessary condition for unidimensional unfolding that is also well known.

Proposition 14. *Any unidimensional system has orders that form a subset of the vertices in a shortest path on the permutohedron.*

Definition 15. A system will be called *straight* when its orders form a subset of the vertices in a shortest path on the permutohedron.

Thus a straight system determines two opposite, canonical *listings* of its orders. It is not difficult to design an algorithm that constructs a listing while checking the system for straightness. A listing will be encoded as an array `perm` indexed by $1, 2, \ldots, n$, with $\texttt{perm}_i$ referencing to the order $\leq_a$ that comes in the ith position. Here we simply say that $\texttt{perm}_i$ contains a.

SKETCH OF THE ALGORITHM. To improve algorithm performance, we will record in another field of $\mathtt{perm}_i$ an integer number between $-n(n-1)/2$ and $n(n-1)/2$ in such a way that the distance between two orders equals the absolute difference of the two associated numbers.

We start by choosing an order $\leq_a$, and initialize $\mathtt{perm}_1$ with a and 0. Then we take a second order, say $\leq_b$, and initialize $\mathtt{perm}_2$ with b and the distance between $\leq_a$ and $\leq_b$.

The remaining orders are considered one at a time. When the $(i+1)$th order $\leq_d$ is considered, its distances to the orders referenced by $\mathtt{perm}_1$ and $\mathtt{perm}_2$ are computed. We can determine from these distances where $\leq_d$ has to be inserted in the listing and determine its associated number. Straightness of the actual listing of $i+1$ orders needs to be checked only by looking at $\leq_d$ and either its two neighbors (on both sides of $\leq_d$ in the listing), or to the two extremities in the listing (if $\leq_d$ was added at one extremity of the listing). □

PROPOSITION 16. *The above algorithm terminates in at most $O(m(m + n \log n))$ steps.*

Proof. When considering the $(i+1)$th order, we need to compute at most four distances (refer to lemma below), and to perform $O(i)$ operations on the array $\mathtt{perm}$. □

LEMMA 17. *The distance between two orders $\leq_a$ and $\leq_b$ can be computed in at most $O(n \log n)$ steps.*

Proof. First construct the order $\leq_c$ with $c_{a_i} = b_i$. The distance between $\leq_a$ and $\leq_b$ equals the distance between $\leq_c$ and the natural order on $\{1, 2, \ldots, n\}$. The latter distance can be computed by applying the list merge sorting algorithm to $\leq_c$ (see Knuth [9, Exercise 5.2.4-21]). □

3. NON-SUFFICIENCY OF UNIMODALITY AND STRAIGHTNESS

Coombs [5] showed that a full, straight, and unimodal system does not necessarily satisfy unidimensional unfolding. His counterexample is based on the following idea. As soon as the ordering of the objects on the Euclidean line is known, the search for a representation reduces to the resolution of a system of linear inequalities. Such an ordering must be a reference order that comes at one extremity of the listing of the orders; assume here it is the natural order on $\{1, 2, \ldots, n\}$, at the end of the listing. Introduce real unknowns x_i for $i = 1, 2, \ldots, n$, and consider the system of inequalities (or equivalently the system of reverse inequalities),

$$x_1 < x_2 < \cdots < x_n,$$
$$x_i + x_j < x_k + x_l,$$

where an inequality of the last type appears whenever the mutation ij occurs before kl in the listing of the orders (we write ij instead of $\{i, j\}$). A one-representation (f, g) exists iff this system of linear inequalities admits a solution. Indeed, if this system has a solution, set $g(i) = x_i$, and then, for $a \in A$, pick $f(a)$ in such a way that

$$2f(a) > g(i) + g(j) \qquad (\text{resp. } 2f(a) < g(i) + g(j))$$

if the mutation ij, where $i <_o j$, occurs (resp. does not occur) between $\leq_a$ and $\leq_o$.

To take advantage of this idea in the non-full case, we need to characterize among all reference orders for a unidimensional system those that can be used as the ordering of objects in at least one representation. This characterization is the goal of the next two sections.

We mention that the technique of Scott [13] can be applied to the system of inequalities mentioned above in order to characterize unidimensional systems. However, as shown by Michell [11], the theoretical conditions derived on the system seem to remain intricate.

4. Straight Reference Orders

Given a system, a first step in the search for a representation could be the construction of a reference order for unimodality that does not destroy straightness.

Definition 18. An order $\leq_o$ on X is a *straight reference order* for a (straight) system $\mathscr{M} = (X, \{\leq_a \mid a \in A\})$ if this system is unimodal with respect to $\leq_o$, and, moreover, the system augmented with $\leq_o$ and the opposite of $\leq_o$ is straight.

Given a straight system, the following lemma shows that a straight reference order always must be added, or come at the beginning or at the end of any listing (and, of course, its opposite at the other extremity). A *convex set* in a graph is a set C of vertices such that C contains all the vertices on all shortest paths connecting two vertices of C. The *convex closure* of a set A of vertices is the smallest convex set containing A.

LEMMA 19. *Let* $\leq_o$ *be a reference order for a unimodal system* $\mathscr{M}$. *Then* $\leq_o$ *belongs to the convex closure (on the permutohedron) of* $\{\leq_a |a \in A\}$ *only if* $\leq_o \in \{\leq_a |a \in A\}$ *and, in that case,* $\leq_o$ *does not belong to the convex closure of the other orders.*

Proof. Assume w.l.o.g. that $\leq_o$ is the natural order. All orders $\leq_a$ unimodal with respect to $\leq_o$ have $2 <_a 1$ except if $\leq_a = \leq_o$. Thus all orders $\leq_b$ in the convex closure of the orders distinct from $\leq_o$ also have $2 <_b 1$. □

What can we say about existence and uniqueness of a straight reference order for a given, straight, and unimodal system $\mathscr{M}$? When counting straight reference orders, we will only consider those that are, or can be, added at the end of the listing (assuming, of course, this listing to be fixed). In the sequel, such a straight reference order will be called *final*.

DEFINITION 20. Let $2 \leq \alpha \leq n$. Then α determines a *strong fault* in $\mathscr{M}$ if it determines a fault and, moreover, all orders $\leq_a$, for $a \in A$, have the same element in rank α.

The interpretation of a strong fault in Coombs' setting is obvious: there is an object that is related to any other in the same way by all subjects without being the most preferred object.

PROPOSITION 21. *Assume that the system* $\mathscr{M}$ *is unimodal and straight with* $m \geq 2$ *and that it admits a final straight reference order* $\leq_o$ *on* X *(which needs not be one of the* $\leq_a$*'s). Then this final straight reference order is unique iff there is no strong fault in* $\mathscr{M}$. *More precisely, there are* 2^s *final straight reference orders if* s *is the number of strong faults in* $\mathscr{M}$.

Proof. If α determines a strong fault with object i at rank α, a final straight reference order $\leq_o$ can be transformed into another one by moving i to the other side of the interval in $\leq_o$ formed by all the objects having rank less than α in the system.

We now assume that there is more than one final straight reference order. If only one object appears in the last position of all the $\leq_a$'s, then n determines a strong fault. If two objects are seen in those last positions, any straight reference order has a well-defined position at one extremity of the listing: any two final straight reference orders $\leq_o$ and $\leq_{o'}$ must end with the same object. We also know from Remark 8 that $\leq_o$ is transformed into $\leq_{o'}$ through a sequence of steps of the form:

—select a box B_1 in $\leq_o$ and flip it;

—select a box $B_2 \neq B_1$ inside B_1, and flip it;

...

—select a box $B_k \neq B_{k-1}$ inside B_{k-1}, and flip it, where each of these boxes corresponds to a fault in the system. Take an element i that is located at one extremity of box B_1 and does not belong to box B_2. For $y \in B_1 \setminus \{i\}$, we obtain ($y <_o i$ and $i <_{o'} y$) or ($y <_{o'} i$ and $i <_o y$). In order to have straightness of all $\leq_a$'s plus $\leq_o$ and $\leq_{o'}$, there can be no mutation iy with $y \in B_1 \setminus \{i\}$ between two of the $\leq_a$'s (because otherwise one of $\leq_o$ or $\leq_{o'}$, which are both final, cannot be straight). Hence the relative position of i and y is the same in all $\leq_a$'s.

If $\{i\} = B_1 \setminus B_2$, we see immediately that i must be the largest element in the beginning set B_1 for each order $\leq_a$; thus $\mathrm{rk}_a(i)$ determines a strong fault. If $\{i\} \subset B_1 \setminus B_2$, then $B_1 \setminus B_2$ contains only one other element i^* extremal in B_1 for $\leq_o$ (and for $\leq_{o'}$). The same arguments can be used for i^* as for i: the relative position of i^* and y taken in $B_1 \setminus \{i^*\}$ is the same in all $\leq_a$'s. Hence, the largest element in the beginning set B_1 for any $\leq_a$ must be always i or always i^*. Thus we have a strong fault at $\mathrm{rk}_a(i)$ or at $\mathrm{rk}_a(i^*)$. This establishes the uniqueness result. A recurrence on s completes the proof. □

Remark 22. The proof shows that the generation of all final straight reference orders from one of them is done in a simple way: objects located at strong faults can be moved to only one other, well defined, position.

The following result will have an important consequence on the design of an algorithm for checking whether a system admits a final straight reference order (see Remark 24 below).

THEOREM 23. *Any unimodal and straight system $\mathscr{M}$ admits a straight reference order.*

Proof. (1) The thesis holds for $m = 2$ in case there is only one fault. By Proposition 21, the reference order is then unique up to reversing. Let $A = \{a, b\}$, $u = a_n$, and $v = b_n$. Thus $u \neq v$ since $n > 1$. Taking the reference order $\leq_o$ with $u <_o v$, we prove that $\leq_a$, $\leq_b$, $\leq_o$ is a listing. If it were not, some mutation would appear between $\leq_a$ and $\leq_b$, and also between $\leq_b$ and $\leq_o$. Let us pick one such mutation ij with $[i, j]_o$ as large as possible. Exchanging notations i and j if necessary, we will have $i <_a j$, $j <_b i$, and $i <_o j$. Note that $l <_o i$ implies by unimodality that $i <_b l$ and, similarly, that $j <_o p$ implies $j <_a p$. Then $l <_o i$ implies both $l <_o j$ and $j <_b l$; by the choice of i and j, there results $j <_a l$. Similarly, $j <_o p$ implies both $i <_o p$ and $i <_a p$; thus also $i <_b p$.

From the above, $]-\infty, j]_a = [i, j]_o =]-\infty, i]_b$ and there is a fault determined by $\mathrm{rk}_a(j) = \mathrm{rk}_b(i)$. By our starting assumption, $j = u$ and $i = v$, in contradiction with $i <_o j$ and $u <_o v$.

(2) The thesis holds for $m = 2$. Restrict the system to the objects in the first fault (that is, the fault determined by the least possible rank). By

the first part of the proof, there exists a straight reference order $\leq_o$ for this restricted system. Now collapse the fault in one artificial object, and construct a straight reference order $\leq_1$ for the system consisting of this artificial object and all objects in the second fault but not in the first (with obvious orders). Restore inside $\leq_1$ the original objects according to $\leq_o$. The resulting order is a straight reference order for the system restricted to the second fault. A similar construction can be applied to each successive fault.

(3) The thesis holds in the general case. Assume that the system $\mathcal{M}$ is unimodal and straight, with $\leq_a$ the first order in a listing, and $\leq_e$ the last one. Consider any reference order for $\{\leq_a, \leq_e\}$ such that $\leq_a$, $\leq_e$, $\leq_o$ is a listing (we just proved the existence of $\leq_o$). Of course, $\{\leq_a | a \in A\} \cup \{\leq_o\}$ is automatically straight. Let us show that any order $\leq_c$, for $c \in A$, is unimodal with respect to $\leq_o$. If not, we have $i, j, k \in X$ such that

$$i <_o j <_o k$$

and j after i and k in $\leq_c$. This leads to two cases:

(1) $i <_c k <_c j$. Then we infer that $i <_e j$ from straightness, and that $j <_e k$ from unimodality. Straightness gives $k <_a j$ and then unimodality gives $j <_a i$. Hence the last element of the set $\{i, j, k\}$ in respectively $\leq_a$, $\leq_c$, $\leq_e$ is i, j, k. This is impossible because the system restricted to $\{i, j, k\}$ must remain unimodal.

(2) $k <_c i <_c j$. The arguments are similar. □

Remark 24. With the notations of part (3) of preceding proof, to construct a straight reference order for a straight and unimodal system, one needs only to look at the "extreme" orders $\leq_a$ and $\leq_e$: any final reference order for $\{\leq_a, \leq_e\}$ is also a final reference order for the whole system.

Here is how to check whether a given system $\mathcal{M}$ admits a final reference order.

SKETCH OF THE ALGORITHM.

Phase 1. Find, if possible, a listing of the orders (using the algorithm of Section 2). If none exists, straightness is violated and there is no final reference order.

Phase 2. Denoting by $\leq_a$ the first and by $\leq_e$ the last orders in the listing, try to construct a final reference order $\leq_o$ for $\{\leq_a, \leq_e\}$, with $\leq_{o'}$ its opposite (apply a variant of the algorithm sketched in Section 1, taking also into account the desired straightness of $\{\leq_{o'}, \leq_a, \leq_e, \leq_o\}$). In case of failure, the given problem has no solution.

Phase 3. For $c \in A \setminus \{a, e\}$, check that $\leq_c$ is unimodal with respect to $\leq_o$. If this is not always true, there is no final reference order for the given system. Otherwise, $\leq_o$ is a solution. □

PROPOSITION 25. *This algorithm has running time in* $O(m^2 + n^2 + mn \log n)$.

Proof. Phase 1 performs in at most $O(m(m + n \log n))$ steps (Proposition 16), Phase 2 in at most $O(n^2)$ steps (Proposition 12), Phase 3 at most $O(mn)$ steps (see after Proposition 5). □

5. UNIDIMENSIONAL UNFOLDING

How do we decide whether a given system $\mathscr{M}$ satisfies unidimensional unfolding? We cannot simply state that it suffices to build a final straight reference order, and, in case of success to work out the resulting system of inequalities (as in Section 3). There could be "good" (i.e., leading to a representation) and "bad" (i.e., not leading to any representation) orders among the 2^s final straight reference orders, where s is again the number of strong faults in the system. Fortunately, the following result shows that any final straight reference order will work.

THEOREM 26. *Let* $\mathscr{M} = (X, \{\leq_a | a \in A\})$ *be a straight and unimodal system, and let* $\leq_o$ *be any final straight reference order for* $\mathscr{M}$. *Then* $\mathscr{M}$ *admits a one-representation (in* $\mathbb{R}$*) iff it admits a one-representation* (f, g) *that also satisfies for all* $x, y \in X$:

$$g(x) < g(y) \qquad \text{iff} \qquad x <_o y.$$

Proof. Suppose that $\mathscr{M}$ admits a one-representation (f', g'). We then set, for $x, y \in X$,

$$x \leq_{o'} y \qquad \text{iff} \qquad g'(x) \leq g'(y).$$

Then $\leq_{o'}$ is a straight reference order for $\mathscr{M}$, and either it is final or its opposite is. We may assume that $\leq_{o'}$ is final. We indicate how to transform the representation (f', g') into a representation (f, g) satisfying the conditions of the theorem.

By our analysis of final straight reference orders, we know that $\leq_{o'}$ is transformed into $\leq_o$ by a sequence of steps of the following type. Select a strong fault α, with thus $2 \leq \alpha \leq n$, and, a being any subject, move the object a_α from one end of the interval formed in the actual reference order by $a_1, a_2, \dots, a_\alpha$, to the other end of that interval (see Remark 22); note that after each such step we again have a final straight reference order. We only treat the case in which $\leq_{o'}$ is transformed in only one step into $\leq_o$ and indicate how to modify (f', g') under that assumption.

Let i^* and i be the extremities of the interval formed in $\leq_{o'}$ by $a_1, a_2, \ldots, a_\alpha$, with $i^* <_{o'} i$. We handle the case in which i^* is not the initial element of $\leq_{o'}$ and i is not the final element, leaving the other cases to the reader. Note that a_α equals i or i^*; we give details in the first eventuality only (in fact, taking the reverse listing of the orders $\leq_a$ exchanges the two eventualities). Set

$$k = \max_{\leq_{o'}}\{x \in X | x <_{o'} i^*\},$$

$$j = \max_{\leq_{o'}}\{x \in X | x <_{o'} i\},$$

$$l = \min_{\leq_{o'}}\{x \in X | i <_{o'} x\}.$$

The new final straight reference order $\leq_o$ is thus obtained from $\leq_{o'}$ by moving i to a position between k and i^*. If $x \in [i^*, i]_{o'}$ and $y \in X \setminus [i^*, i]_{o'}$, then we have $x <_a y$ for each a in A. This fact will be used repeatedly. For instance, $i <_a k$, together with $g'(k) < g'(i)$, implies the left inequality below, while the right inequality comes from $i^* <_a i$, together with $g(i^*) < g(i)$:

$$\frac{g'(k) + g'(i)}{2} < f'(a) < \frac{g'(i^*) + g'(i)}{2}.$$

Hence, if at the same time we subtract a constant positive value t from all $g'(x)$ with $x <_{o'} i^*$ and add t to all $g'(y)$ with $i <_{o'} y$, we still have a representation. We can choose t in such a way that the resulting representation, again denoted as (f', g'), satisfies for any two subjects a, e from A,

$$2f'(e) - 2f'(a) + g'(j) < g'(l),$$

that is

$$2f'(e) - g'(l) < 2f'(a) - g'(j). \tag{1}$$

Since $j <_a k$ with $g'(k) < g'(j)$, we have

$$\frac{g'(k) + g'(j)}{2} < f'(a),$$

that is

$$g'(k) < 2f'(a) - g'(j). \tag{2}$$

Similarly, since $i^* <_e l$ with $g'(i^*) < g'(l)$, we have

$$f'(e) < \frac{g'(i) + g'(l)}{2},$$

that is

$$2f'(e) - g'(l) < g'(i^*). \tag{3}$$

Moreover, from $k <_{o'} i^*$ and $g'(k) < g'(i^*)$, we derive

$$g'(k) < g'(i^*). \tag{4}$$

Because of Eqs. (1)–(4), there is a real number r such that, for all subjects a and e in A,

$$2f'(e) - g'(l) < r < 2f'(a) - g'(j),$$
$$g'(k) < r < g'(i^*).$$

If we now take $f = f'$ and $g = g'$, except for $g(i) = r$, we readily see that (f, g) is again a one-representation. Indeed, we only need to check that the comparisons of object i with the other objects are faithfully represented (because subjects and objects distinct from i keep the same representing points). For $x \in X \setminus \{i\}$, we have for each subject b either $\mathrm{rk}_b(x) < \alpha$ or $\mathrm{rk}_b(x) > \alpha$. In the first case, $g'(x) \leq g'(j) < g'(i)$; since $g(i) < 2f(b) - g(j)$, we derive $f(b) > (g(i) + g(x))/2$, which is the required condition because $g(i) < g(x)$. In the second case, we may have $g'(x) \leq g'(k)$, or $g'(l) \leq g'(x)$.

The first alternative gives

$$\frac{g(x) + g(i)}{2} \leq \frac{g(k) + g(i)}{2} < \frac{g(k) + g(i^*)}{2} < f(b)$$

(the last inequality is because $i^* <_b k$), and thus $(g(x) + g(i))/2 < f(b)$, which is the required condition because here $g(x) < g(i)$.

The second alternative gives

$$\frac{g(x) + g(i)}{2} \geq \frac{g(l) + g(i)}{2} > f(b)$$

(the last inequality is by the choice of $g(i)$), and thus $f(b) < (g(x) + g(i))/2$, which is the required condition because here $g(i) < g(x)$.

We finally note that, for any two objects x, y, we clearly have $g(x) < g(y)$ iff $x <_o y$. □

An algorithm for checking a system for unidimensionality now results from our analysis. First, find (if possible) a listing and a final straight reference order by applying the algorithm from Section 4. Then determine a solution (if any) to the system of linear inequalities described in Section 3. As the coefficients take only the values 1, 0, and -1, this can be done in polynomial time (thanks to a result of Tardos [14] on combinatorial programs). Let $F(p, q)$ be the number of steps required to check the consistency and produce a solution to a system of p strict linear inequalities in q variables.

THEOREM 27. *There is a polynomial time algorithm for checking whether a system $\mathcal{M}$ satisfies unidimensional unfolding. In the positive case, the algorithm produces a one-representation of $\mathcal{M}$. Execution is done in at most $O(m^2 + n^2 + mn \log n + F(n^2, n))$ steps.*

REFERENCES

1. J. BARTHOLDI AND M. A. TRICK, Stable matchings with preferences derived from a psychological model, *Oper. Res. Lett.* **5** (1986), 165–189.
2. A. BJÖRNER, Orderings of Coxeter groups, *Contemp. Math.* **34** (1984), 175–195.
3. D. BLACK, "The Theory of Committees and Elections," Univ. Press, Cambridge, UK, 1958.
4. I. BORG AND J. LINGOES, "Multidimensional Similarity Structure Analysis," Springer-Verlag, New York, 1987.
5. C. H. COOMBS, "A Theory of Data," Wiley, New York, 1964.
6. J.-CL. FALMAGNE AND J.-P. DOIGNON, Bisector spaces: Geometry for triadic data, *in* "Mathematical Psychology: Current Developments" (J.-P. Doignon and J.-Cl. Falmagne, Eds.), pp. 89–105, Springer-Verlag, New York, 1991.
7. J. FELDMAN HÖGAASEN, Ordres partiels et permutoèdre, *Math. Sci. Humaines* **28** (1969), 27–38.
8. G. TH. GUILBAUD & P. ROSENSTIEHL, Analyse algébrique d'un scrutin, *Math. Sci. Humaines* **4** (1963), 9–33.
9. D. E. KNUTH, "The Art of Computing Programming. (Vol. 3. Sorting and Searching)," 2nd printing, Addison-Wesley, Menlo Park, CA, 1973.
10. CL. LE CONTE DE POLY-BARBUT, Le diagramme du treillis permutoèdre est intersection des diagrammes de deux produits directs d'ordres totaux, *Math. Inform. Sci. Humaines* **112** (1990), 49–53.
11. J. MICHELL, "An Introduction to the Logic of Psychological Measurement," Erlbaum, Hillsdale, NJ, 1990.
12. D. ROMERO, "Variations sur l'effet Condorcet," unpublished doctoral dissertation, Université Scientifique et Médicale de Grenoble, 1978.
13. D. SCOTT, Measurement models and linear inequalities, *J. Math. Psychol.* **1** (1964), 233–247.
14. É. TARDOS, A strongly polynomial algorithm to solve combinatorial linear programs, *Oper. Res.* **34** (1986), 250–256.