Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle

Ronald D. Armstrong
Wade D. Cook
Mabel T. Kung
Lawrence M. Seiford

Priority ranking and minimal disagreement : a weak ordering model

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 16, nº 4 (1982), p. 309-318.

<http://www.numdam.org/item?id=RO_1982__16_4_309_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 16, n° 4, novembre 1982, p. 309 à 318)

PRIORITY RANKING AND MINIMAL DISAGREEMENT: A WEAK ORDERING MODEL (*) (**)

by Ronald D. ARMSTRONG (1), Wade D. COOK (2), Mabel T. KUNG (3), and Lawrence M. SEIFORD (1)

Abstract. — *In an earlier paper by Blin a model is presented for determining a consensus among a set of ordinal rankings. His model is designed specifically to derive that linear preference ordering which exhibits the minimal amount of disagreement relative to the set of voter rankings. In this paper we extend Blin's model to include the set of all weak orderings. Since this more general formulation cannot be solved via the standard linear assignment model, two algorithms are presented for determining the optimal weak ordering. Computational results are provided.*

Keywords: Ranking; consensus; algorithms; transportation; branch and bound.

Résumé. — *Blin a publié un modèle pour déterminer un consensus parmi un ensemble de rangements (rankings). Son modèle a été spécifiquement construit pour obtenir le rangement des préférences qui minimise le désaccord relativement à l'ensemble des rangements soumis par les votants. Nous étendons ici le modèle de Blin de façon à inclure l'ensemble de tous les rangements faibles. Puisque cette formulation plus générale ne peut se réduire à un problème canonique d'affectation linéaire, deux algorithmes sont présentés pour déterminer le rangement faible optimal. Nous donnons des résultats numériques.*

Keywords: Ranking; consensus; algorithms; transportation; branch and bound.

1. INTRODUCTION

In an earlier paper Blin [2] presented a model for aggregating individual/voter preferences expressed as ordinal rankings. He has shown that by combining voter responses into an agreement matrix an optimal ranking can be determined by solving a simple linear assignment problem. While voters are permitted to supply weak orderings (ties allowed), the model can be used only to derive the best strict *linear* ordering. In cases where the true optimum is not a linear ordering the assignment approach is deficient.

In this paper we extend the results of [2] to the general case where the optimal (consensus) ranking is to be selected from the space of all *weak*

(*) Received September 1981.
(**) Supported under NSERC Grants A8966 and A5177.
(1) Department of General Business, University of Texas, Austin, Texas, U.S.A., 78712.
(2) Faculty of Administrative Studies, York University, Toronto, Ontario, Canada, M3J 2R6
(3) Business School, University of California, Fullerton, California, U.S.A. 92686.

orderings. A generalized assignment model is developed, and two algorithms for solving the model are presented. Computational results are included.

2. A BINARY SOCIAL CHOICE FUNCTION FOR WEAK ORDERINGS

Consider a social choice decision problem in which each of m voters supplies a weak ordering of a set of n objects. Each ranking vector A^l, $l=1, 2, \ldots, m$ takes the form $A^l=(a_1^l, \ldots, a_n^l)$ where a_i^l is the rank assigned to the ith object by the lth voter. For example, with $n=3$ $A=(2.5, 1, 2.5)$ indicates that objects 1 and 3 are tied for second and third place (rank of 2.5), and object 2 is ranked in first place (rank of 1).

Let us represent each vector A^l by its associated binary matrix $P^l=(p_{ij}^l)$ where:

$$p_{ij}^l=\begin{cases} 1 & \text{if object } i \text{ has rank } j; \\ i=1, 2, \ldots, m, & j=1, 1.5, 2, \ldots, n, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix representation of $A=(2.5, 1, 2.5)$, for example, is:

	1	1.5	2	2.5	3
1	0	0	0	1	0
$P=$ 2	1	0	0	0	0
3	0	0	0	1	0

REMARK : If attention is restricted to linear orderings, each matrix P is an $n \times n$ permutation matrix (*see* Blin [2]).

In order to obtain a consensus ranking many criteria are possible. Following the approach of Blin, a logical and widely accepted definition of an optimal ranking is that which solves the following l^1 norm problem:

Model 1: Minimum Distance Model (l^1 norm):

$$\min_{Q=(q_{ij})} \sum_l \sum_i \sum_j |p_{ij}^l - q_{ij}|,$$

where the minimum is taken overall nx $(2n-1)$ matrices Q corresponding to weak orderings (Blin considers only linear orderings).

While Model 1 is simple in form, it is extremely difficult to handle due to the special structural requirements of the (q_{ij}). An alternative and far more tractable approach (Model 2) is based on the simple average of the P^l matrices.

Define the *disagreement* coefficient:

$$\Phi_{ij}=\sum_l (1-p_{ij}^l).$$

Model 2: Minimum Disagreement Model:

$$\min_{X=(x_{ij})} \sum_i \sum_j \Phi_{ij} x_{ij}.$$

$$\text{Subject to:} \quad \left\{ \begin{array}{ll} \sum_j x_{ij}=1, & i=1, 2, \ldots, n, \\ \sum_i x_{ij}-D_j=0, & j=1, 1.5, 2, \ldots, n, \\ x_{ij}\in\{0, 1\}, & \end{array} \right. \qquad (2.1)$$

where each variable D_j represents the number of objects having rank j. Define the consensus ranking as that whose matrix representation is $X=(x_{ij})$.

Models 1 and 2 are generalizations (to the set of weak orderings) of the two models given in [2].

We state the following two theorems. Proofs are straightforward and are, thus, omitted.

THEOREM 2.1: *In the space of weak orderings Models* 1 *and* 2 *are equivalent.*

THEOREM 2.2: *In the space of weak orderings the* l^1 *and* l^2 *distance norms are equivalent when applied to Model* 1.

Theorems 2.1 and 2.2 are analogous to lemmas 2 and 3 respectively in [2].

While Model 2 is more convenient to deal with than Model 1, it cannot be solved in a manner as straightforward as was the case for linear orderings (*see* [2]). The string of variables $D_1, D_{1.5}, \ldots, D_n$ must constitute a ranking. While it is possible to reformulate (2.1) as a large integer programming problem, direct application of any standard I. P. algorithm would prove ineffective even for moderate sized problems.

In the section to follow we present two branch and bound algorithms for solving (2.1). The first is particularly simple to understand and apply, but is limited with respect to the size of problems which it can solve. The second, more complex, approach is designed to handle large problems. Computational results pertaining to both procedures are discussed.

3. ALGORITHMS FOR THE MINIMUM DISAGREEMENT MODEL

3.1. An implicit enumeration algorithm

In this algorithm we create a solution tree consisting of partial rankings, each of which has an associated bound. At each stage in the algorithm that partial with the lowest bound to date is selected as the *parent* from which to branch and create *offspring* partials. Each offspring will have one more object ranked than did its parent. When a full weak ordering is finally obtained, whose bound is not greater than any of those for the partials, that ordering is optimal.

The lower bound on all weak orderings is $L=\sum_i \Phi_{ij_i}$ with Φ_{ij_i} being the minimum element in the ith row of the disagreement matrix. An upper bound U on all rankings can be obtained by solving (2.1) for the optimal linear ordering. In stage 1 2 $n-1$ partial rankings are created in which object 1 is ranked 1, 1.5, 2, 2.5, . . ., n. The lower bound on the partial having rank r assigned to object 1 is $L_r=L+(\Phi_{1r}-\Phi_{1j_1})$.

At any stage in which the parent has, say l objects ranked, then the *eligible* ranks which the $(l+1)$ st object can assume must be known at the time branching takes places. For this purpose an *eligibility* vector E must be carried along with each partial ranking R. $E=(e_1, e_{1.5}, \ldots, e_n)$ is created as follows.

Initially:

$$e_j=\left\{\begin{array}{cl} 2j-1 & \text{for } j\leqq(n+1)/2. \\ 2n-(2j-1) & \text{for } (n+D_1/2)<j\leqq n. \end{array}\right\} \tag{3.1}$$

For example, with $n=5$ the allowable number of ranks at positions (1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5) are (1, 2, 3, 4, 5, 4, 3, 2, 1) $=E$, respectively. Each time a rank is assigned, certain other ranks become ineligible. For example, if object #1 is assigned rank 2.5 then ranks 1.5, 2, 3, 3.5 become ineligible. Hence, the partial ranking (2.5, –, –, –, –) carries with it the *modified* E-vector (1, 0, 0, 3, 0, 0, 3, 2, 1). Note that $e_{2.5}$ now equals $3=4-1$; since one object has been ranked at 2.5, at most 3 more can possess that rank. The following rules are used to restructure E at each stage.

Rule 1: An *even* number of objects must be assigned at each non-integer rank position (e. g. the number of objects ranked at position 2.5 must be 0,2 or 4); an *odd* number of objects must be assigned at any integer position (e. g. the number of objects ranked at position 3 must be 0, 1, 3, or 5).

Rule 2: If an object is being assigned an integer rank r (e. g. $r=2$), and if no other object currently possesses that rank then positions $r-.5$ and $r+.5$ are both set to zero in the E-vector and e_r is decreased by 1. If r is a non-integer rank then positions $r-1$, $r-.5$, $r+.5$ and $r+1$ are all set to zero in E.

Rule 3: If the object being assigned to a position r is not the first in that position we proceed as follows: If r is integer and an odd number of objects (1, 3, 5...) has already been assigned (i. e. this is the 2nd or 4th or 6th... object being assigned) then the *two* closest nonzero rank positions less than r *and* the *two* greater than r are set to zero; e_r is decreased by 1. If an even number currently have rank r then only e_r is decreased by 1.

If r is non-integer, and an odd number has been previously assigned then only e_r is decreased (by 1 unit). If an even number has been assigned then e_r is decreased by 1 and the *two* nearest nonzero ranks below and the two above are set to zero.

The following is a statement of the essential steps of the algorithm:

Step 1: Compute an upper bound U by solving the linear assignment problem associated with (2.1) (i. e. using $j=1, 2, \ldots, n$ only). Compute the lower bound $L=\sum_i \Phi_{ij_i}$. Generate the $2n-1$ partial rankings of object 1 with ranks 1, 1.5, 2, . . ., and the associated (sorted) bounds

$$L_r = L + (\Phi_{1r} - \Phi_{1j_1}).$$

Store the partial rankings in $2n-1$ vectors $R_1, \ldots, R_{2n-1}$ each of dimension n. Create, using rules 1, 2, 3, the $2n-1$ eligibility vectors $E_1, \ldots, E_{2n-1}$.

Go to Step 2.

Step 2: At stage l select the partial ranking with the lowest bound (designated R_1 due to sorting). For each rank position j for which $e_j \neq 0$ create an offspring from R_1 (assume R_1 has l objects), by assigning to object $l+1$ the rank associated with that rank position. Update the lower bound on R_1, and update E_1 for each of the new $(l+1)$-vectors using rules 1, 2, 3. If any L_r exceeds

$U-1$, discard the associated partial ranking. Renumber all L, R, E combinations such that $L_1 \leqq L_2 \leqq \ldots$

Go to step 3.

Step 3: If R_1 has all objects ranked, then R_1 is optimal. Otherwise go to step 2.

Example: Suppose 10 voters rank 5 objects. Let the rankings l be:

	Rankings for Voter									
	1	2	3	4	5	6	7	8	9	10
Object:										
1	1	1.5	4	3	1	2.5	5	3	5	1
2	2	1.5	5	3	3	2.5	4	3	4	3
3	3	5	2.5	3	4	2.5	3	3	2	2
4	4	4	1	5	2	2.5	2	3	2	4
5	5	3	2.5	1	5	5	1	3	2	5

The corresponding disagreement matrix is given by:

	Rank Position								
	1	1.5	2	2.5	3	3.5	4	4.5	5
Object:									
1	$\underline{7}$	9	10	9	8	10	9	10	8
2	10	9	9	9	$\underline{6}$	10	8	10	9
3	10	10	8	8	$\underline{6}$	10	9	10	9
4	9	10	$\underline{7}$	9	9	10	7	10	9
5	8	10	9	9	8	10	10	10	$\underline{6}$

The best complete ranking is $[1, 3, 3, 3, 5] \Rightarrow U=34$. Hence, at each stage to follow we will keep partial rankings only if the corresponding $L_r \leqq 33$.

Step 1: $L=32$ (sum of numbers underlined in above table). Initial E-vector: $E=[1, 2, 3, 4, 5, 4, 3, 2, 1]$ e. g., this says that 2 objects can be assigned at rank position 1.5.

The $2n-1=9$ partial rankings are (ranking of object # 1 only).

$(1, -) : L_1 = L + (7-7) = 32$, $E=[0, 0, 1, 2, 3, 4, 3, 2, 1]$;

$(1.5, -) : L_{1.5} = 34$, discard since $L > 33$;

$(2, -) : L_2 = 35$, discard;

$(2.5, -) : L_{2.5} = 34$, discard;

$(3, -) : L_3 = 33$, $E=[1, 2, 1, 0, 4, 0, 1, 2, 1]$;

$(3.5, -) : L_{3.5} = 35$, discard;

$(4, -) : L_4 = 34$, discard;
$(4.5, -) : L_{4.5} = 35$, discard;
$(5, -) : L_5 = 33$, $E = [1, 2, 3, 4, 3, 2, 1, 0, 0]$.
We therefore retain only partials $(1, -)$, $(3, -)$, $(5, -)$.

Step 2: Selecting the lowest bound of 32 on $(1, -)$ we compute new bounds.
$(1, 2, -) : L_{1, 2} = 35$, discard;
$(1, 2.5, -) : L_{1, 2.5} = 35$, discard;
$(1, 3, -) : L_{1, 3} = 32$, $E = [0, 0, 1, 0, 2, 0, 1, 2, 1]$;
$(1, 3.5, -) : L_{1, 3.5} = 36$, discard;
$(1, 4, -) : L_{1, 4} = 34$, discard;
$(1, 4.5, -) : L_{1, 4.5} = 36$, *discard;*
$(1, 5, -) : L_{1, 5} = 35$, discard.

We retain only $(1, 3, -)$, $(3, -)$, $(5, -)$.

Branching from $(1, 3, -)$ we get only $(1, 3, 3-)$ with a bound less than 34.

$$L_{1, 3, 3} = 32, \qquad E = [0, 0, 0, 0, 1, 0, 0, 0, 1].$$

Branching from $(1, 3, 3, -)$ all bounds exceed 33.

We retain only $(3, -)$ and $(5, -)$.

Branching from $(3, -)$, only $(3, 3, -)$ has a bound less than 34.

$$L_{3, 3} = 33, \qquad E = [1, 0, 0, 0, 3, 0, 0, 0, 1].$$

Branching from $(3, 3, -)$, only $(3, 3, 3, -)$ has a bound less than 34.

$$L_{3, 3, 3} = 33 \qquad E = [1, 0, 0, 0, 2, 0, 0, 0, 1].$$

Branching from $(3, 3, 3, -)$, all bounds exceed 33.

We retain only $(5, -)$.

Branching from $(5, -)$, only $(5, 3, -)$ has a bound less than 34.

$$L_{5, 3} = 33, \qquad E = [1, 2, 1, 0, 2, 0, 1, 0, 0].$$

Branching from $(5, 3, -)$, only $(5, 3, 3, -)$ has a bound less than 34.

$$L_{5, 3, 3} = 33, \qquad E = [1, 0, 0, 0, 1, 0, 0, 0, 0].$$

All partials generated from $(5, 3, 3, -)$ have bounds greater than 33.

Hence, the optimum is the linear ordering $(1, 3, 3, 3, 5)$.

Since most voter responses were strict linear orderings in this particular example, so also was the consensus. In general, however, when a relatively high percentage of the voters provide tied preferences, such a result will not occur. There is, of course, no direct means of determining this in advance.

While the algorithm described above is simple to apply, experience, based on several test problems, indicates that the number of iterations increases drastically with the number of objects. For problems involving more than 10 objects, storage requirements become excessive. To deal with the case where larger numbers of objects are present, an alternate solution procedure has been designed, computerized and tested. We outline the essentials of this procedure below.

3.2. A transportation-based algorithm

This method begins by creating a relaxed version of problem (2.1) which can be solved efficiently. If the optimal solution to this initial *candidate* problem is feasible for (2.1) then that solution is optimal for (2.1). If not, two new problems are created by further restricting (2.1) such that:

(*a*) the union of the feasible regions of the two problems is identical to that of (2.1);

(*b*) the intersection of the feasible regions for the two new problems is empty, and;

(*c*) a relaxation of either new problem is easily solved. One of the two candidate problems is chosen at each stage, and its relaxation is solved. This strategy is continued until the entire feasible region for (2.1) has been implicitly explored.

The initial candidate problem is:

$$\min_{X=(x_{ij})} \sum_i \sum_j \Phi_{ij} x_{ij},$$

subject to:

$$\left.\begin{aligned} \sum_j x_{ij} &= 1, \qquad i=1,2,\ldots,n,\\ \sum_i x_{ij}+s_j &= e_j, \qquad j=1,1.5,2,2.5,\ldots,n. \end{aligned}\right\} \tag{3.2}$$

$$0 \leqq x_{ij} \leqq 1, \quad 0 \leqq s_j \leqq e_j, \qquad i=1,2,\ldots,n, \qquad j=1,\,1.5,\,2,\,2.5,\ldots,n.$$

Problem (3.2) is a relaxation of (2.1). Additionally, it is a capacitated transportation problem for which efficient solution procedures exist (*see* Armstrong *et al.* [1]). In fact, an optimal integer solution will be obtained using an extreme point algorithm.

Let (X^*, S^*) denote the optimal solution to (3.2), and define $D_j^* = e_j - s_j^*$ for each j. Recall that D_j^* represents the number of objects ranked at position j. It must now be verified whether or not the set of D_j^* constitute a ranking. This is accomplished by applying rules 2 and 3 given in the previous subsection (as a matter of convention the $\{e_k\}$ set to zero by a given D_j^* will be those for which $2j - D_j^* \leqq k \leqq 2j + D_j^* - 1$). Now, provided each $e_{k/2}$ is set to zero exactly once we have a ranking. If at any stage there is a duplication (overlap), then that D_j^* which caused the overlap is chosen as a candidate to be further restricted. Rule 3 implies that $e_1 = e_{1.5} = e_2 = 0$ (since $D_{1.5} = 2$) and $e_{1.5} = e_2 = e_{2.5} = e_3 = e_{3.5} = e_4 = 0$ (since $D_j^* = 3$). Since e_2 (also $e_{1.5}$) is set to zero twice, we do not have a ranking. Hence, either $D_{1.5}^*$ or D_3^* can be selected as a candidate to be restricted.

The restriction $D_{j_0} \geqq D_{j_0}^*$ means that the *upper* bound on s_{j_0} is $e_{j_0} - D_j^*$; that is, at least $D_{j_0}^*$ objects are assigned rank j_0. This restriction does not directly eliminate the optimal solution to the parent relaxed candidate problem. However, it follows from rules 1 and 2 that the upper bound on certain D_j adjacent to D_{j_0} can be decreased without eliminating a possible ranking. The addition of these bound restrictions does, however, eliminate the optimal solution to the parent relaxed candidate problem.

The restriction $D_{j_0} \leqq D_j^* - 1$ places a lower bound of $e_{j_0} - D_{j_0}^* + 1$ on s_{j_0}. That is, at most $D_{j_0}^* - 1$ objects are assigned rank j_0 and the current solution becomes infeasible.

When two candidate problems are created, the one with the restriction $D_{j_0} \geqq D_{j_0}^*$ is always chosen to be the current candidate to be examined. A candidate problem can be fathomed as the result of obtaining a ranking, finding no feasible solution or obtaining a linear programming objective value greater than the objective value associated with a ranking. After a candidate problem is fathomed, a new current candidate problem is chosen with a last-in-first-out decision rule. The dual algorithm of Armstrong *et al.* [1] is used to solve all transportation problems with the final solution of the last inspected candidate problem providing an initial dual feasible solution for the current relaxed candidate problem.

Computational Results

The algorithm outlined in this subsection was tested using a FORTRAN computer code on a CYBER 170/750. The results of the computational study are shown in table A. Each test problem used fifty randomly generated weak orderings. Times reported are CPU seconds required to verify the optimal ranking.

TABLE A

Computational Results

# Objects	10	15	20	25	30	35
CPU time.	.8	2.7	10.6	18.5	52.3	84.2
# Transportation problems inspectec.	87	126	282	295	386	422

4. CONCLUSIONS

This paper has presented two branch and bound algorithms for determining the consensus (optimal weak ordering) among a set of voter rankings. The transportation-based procedure is capable of solving relatively large problems, and in addition permits a number of variations of the standard consensus problem (weights on individual voters, upper limits on the number of objects to be ranked in certain positions, etc.). This algorithm has been applied to other distance functions as well, and has proven to be an efficient computational tool in a wide variety of situations.

REFERENCES

1. R. D. ARMSTRONG, D. D. KLINGMAN and D. H. WHITMAN, *Implementation and Analysis of a Variant of the Dual Method for the Capacitated Transshipment Problem*, The European Operations Research Journal, Vol. 4, No. 6, 1980, pp. 403-430.
2. J.-M. BLIN, *A Linear Assignment Formulation of the Multiattribute Decision Problem*, Revue Française d'Automatique, Informatique et Recherche Opérationnelle, Vol. 10, No. 6, 1976, pp. 21-32.