RAIRO. RECHERCHE OPÉRATIONNELLE

ALAIN GUÉNOCHE

Un algorithme pour pallier l'effet Condorcet

RAIRO. Recherche opérationnelle, tome 11, n° 1 (1977), p. 77-83

<http://www.numdam.org/item?id=RO_1977__11_1_77_0>

R.A.I.R.O. Recherche opérationnelle/Operations research
(vol. 11, n° 1, 1977, p. 77 à 83)

UN ALGORITHME POUR PALLIER L'EFFET CONDORCET (*)

par Alain GUÉNOCHE (1)

1. INTRODUCTION

Le problème de l'ordre total à distance minimale d'un tournoi est assez connu et discuté dans la littérature scientifique pour que nous présentions brièvement les faits.

N juges (ou votants) ordonnent p possibilités (ou candidats) suivant leurs préférences. De là, on essaye de construire un ordre total qui reflète au mieux les différentes opinions. Elles sont résumées dans un graphe dont les sommets sont les possibilités ($X = a, b, \ldots$) et un arc orienté qui représente une opinion relie a à b si une majorité de juges préfère le candidat a à b. Ce graphe (X, U) n'est généralement pas transitif, il peut contenir un ou plusieurs circuits. C'est ce qu'on appelle l'effet Condorcet.

En l'absence de procédure « idéale » que le théorème de K. Arrow interdit, on en est réduit à chercher l'ordre total à distance minimale du tournoi, c'est-à-dire, au sens de Slater, celui que l'on obtient en inversant le nombre minimal d'arcs du graphe orienté. L'algorithme de Remage-Thomson, amélioré par J.-C. Bermond, résout ce problème, bien que sa complexité croisse exponentiellement avec p.

Mais une dimension du problème est ainsi écartée. Les arcs qui relient les candidats sont non seulement orientés, mais valués naturellement par le nombre de voix de majorité qu'ils remportent. Ainsi, si sur 20 juges, 15 préfèrent a à b et 12 b à c, l'arc (a, b) vaut 5 et l'arc (b, c) vaut 2. Il est moins contraignant que l'opinion collective contredise l'avis (b, c) en classant c avant b plutôt que de classer b avant a. Pourtant, toutes choses égales d'ailleurs le nombre d'arcs inversés est le même.

C'est pourquoi nous proposons un algorithme qui détermine — par une méthode Remage-Thomson associée à une procédure « branch and bound » — le ou les ordres totaux que l'on obtient en contredisant le nombre minimal de préférences individuelles.

(*) Reçu octobre 1975, révisé juin 1976.
(1) Laboratoire d'Informatique pour les Sciences de l'Homme, Marseille.

2. NOTATIONS

Soit $\mathscr{T}_p$ l'ensemble des tournois $T = (X, U)$ à p sommets $X = \{1, 2, ..., p\}$ dont les arcs sont valués par une fonction

$$v : U \to \left\{0, 1, \ldots, \left[\frac{N}{2}\right]\right\} \quad (^2).$$

Soit n_{ij} le nombre de juges qui préfèrent i à j; $n_{ij}+n_{ji} = N$, $(i, j) \in U$ si et seulement si $n_{ij} > n_{ji}$ et

$$v(i, j) = n_{ij} - \left[\frac{N}{2}\right].$$

Soit $\mathscr{O}_p$ l'ensemble des tournois transitifs à p sommets, isomorphe à l'ensemble des permutations d'ordre p, et $0 = (X, V)$ un élément de $\mathscr{O}_p$. On définit

$$d : \quad \mathscr{T}_p \times \mathscr{O}_p \to \mathbf{N},$$

tel que

$$d(T, 0) = \sum_{\substack{(i,j) \in U \\ (i,j) \notin V}} v(i, j)$$

et

$$J : \quad \mathscr{T}_p \to \mathbf{N},$$

tel que

$$J(T) = \min_{0 \in \mathscr{O}_p} d(T, 0) = d(T, 0^*).$$

Le problème est donc de déterminer les 0^* qui réalisent le minimum de $d(T, 0)$. Dans l'exemple traité ultérieurement, on verra que l'ordre total correspondant n'est pas le même que celui obtenu en minimisant le nombre d'arcs à inverser dans un tournoi, pour le rendre transitif.

Notons :

— T_i (resp. $T_{i,j}$ ...) le graphe déduit de T en supprimant le sommet i (resp. $i, j, \ldots$) et les arcs qui en partent ou y arrivent.

— $f_T(i) = \sum_{\substack{j \in X \\ (j,i) \in U}} v(i, j)$ (somme des valeurs des arcs dont i est sommet terminal).

On a alors la propriété évidente :

$$J(T) = \min_{i \in X} [J(T_i) + f_T(i)]$$

et la valeur de i qui réalise ce minimum est l'élément initial d'un ordre total 0^* qui réalise

$$J(T) = d(T, 0^*).$$

(²) $[N/2]$ (resp. $[N/2]$) représente la partie entière par excès (resp. par défaut) du quotient.

On en déduit

$$J(T) \geqq \min_{i \in X} f_T(i) \quad (1)$$

ce qui permet de construire la procédure « branch and bound » en prenant comme borne inférieure $J'(T)$ du critère J, la plus petite valeur de f_T sur l'ensemble des sommets non classés.

3. ALGORITHME

L'algorithme proposé repose sur la relation (1) et sur les considérations suivantes.

Étant donné un tournoi T, un ordre total associé 0* à distance minimale — au sens précédent — commence par l'une des p possibilités, (la i-ième par exemple), pour laquelle $J(T)$ est certainement supérieure ou égale à $f_T(i)$.

On retiendra la possibilité i qui rend minimal f_T, pour examiner le tournoi T_i et déterminer une borne inférieure $J'(T_{i,j})$ de $J(T)$, plus voisine de $J(T)$ que $f_T(i)$ en choisissant un candidat j tel que

$$f_{T_i}(j) \leqq f_{T_i}(k), \qquad \forall k \neq i.$$

On a alors

$$J(T) \geqq f_T(i) + f_{T_i}(j) = J'(T_{i,j}).$$

L'ordre total 0 que l'on crée ainsi commence par $(i, j, \ldots)$.

On construit donc pas à pas une arborescence, dont le sommet initial est fictif, ses successeurs sont les ordres partiels qui commencent par chacun des candidats. La valeur d'un sommet est la borne inférieur de J (T).

$$J'(T_{i,j,\ldots}) = \sum_{k \,:\, \text{sommets déjà classés}} f_{T_{k-1}}(k).$$

A chaque étape de l'algorithme on détermine le sommet terminal de l'arborescence dont la valeur est minimale. Soit, par exemple $J'(T_{i,j})$ sa valeur. Pour chacun des candidats non classés à ce sommet on prolonge l'arborescence d'un sommet terminal dont on calcule la valeur.

k non classé :

$$J'(T_{i,j,k}) = J'(T_{i,j}) + f_{T_{i,j}}(k),$$

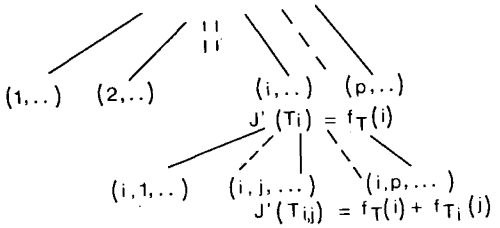
Si à une étape q candidats sont classés, on crée $p-q$ sommets terminaux auxquels correspondent des ordres totaux contenant un candidat supplémentaire.

Si tous les candidats sont classés on s'arrête, sinon on passe à l'étape suivante.

Il est clair que lorsque tous les candidats sont classés, l'ordre total 0* est constitué et

$$J(T) = J'(T_{i,j,\ldots,l}).$$

p candidats dans l'ordre 0*



En opérant sur les ordres partiels correspondant aux sommets dont la valeur est minimale, on est assuré de ne faire les calculs que sur ceux qui sont susceptibles de fournir une solution.

4. EXEMPLE

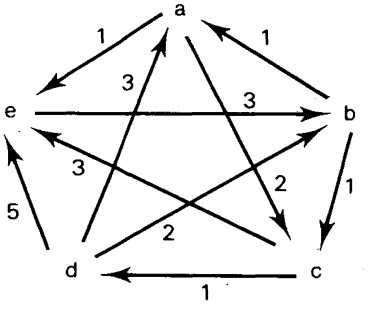
Soit par exemple $N = 13$ et $p = 5$ les 13 opinions suivantes où les candidats sont rangés suivant l'ordre des préférences décroissantes :

(b, c, d, e, a), (d, e, b, a, c), (d, a, c, e, b), (c, d, e, b, a),
(a, b, c, d, e), (d, e, b, c, a), (b, d, a, c, e), (c, d, a, e, b),
(e, b, a, c, d), (b, c, d, a, e), (a, c, d, e, b), (d, a, c, e, b),
(e, a, d, c, b)

Pour toute paire de possibilités on obtient un avis (correspondant au sens de l'arc) et sa valeur.

Paires	Avis	Valeurs	Paires	Avis	Valeurs
$\{a, b\}$	(b, a)	1	$\{b, d\}$	(d, b)	2
$\{a, c\}$	(a, c)	2	$\{b, e\}$	(e, b)	3
$\{a, d\}$	(d, a)	3	$\{c, d\}$	(c, d)	1
$\{a, e\}$	(a, e)	1	$\{c, e\}$	(c, e)	3
$\{b, c\}$	(b, c)	1	$\{d, e\}$	(d, e)	5

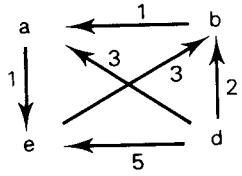
Ce tableau permet de construire le tournoi T suivant :



Les détails des calculs nécessaires figurent intégralement ci-dessous, ainsi que l'arborescence effective.

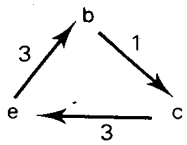
$$f_T(a) = 4, \quad f_T(b) = 5, \quad f_T(c) = 3, \quad f_T(d) = 1, \quad f_T(e) = 9.$$

Première étape : T_d : $J'(T_d) = 1$.



$$f_{T_d}(a) = 1, \quad f_{T_d}(b) = 3, \quad f_{T_d}(c) = 3, \quad f_{T_d}(e) = 4.$$

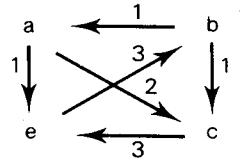
Deuxième étape : $T_{d,a}$: $J'(T_{d,a}) = 2$.



$$f_{T_{d,a}}(b) = 3, \quad f_{T_{d,a}}(c) = 1, \quad f_{T_{d,a}}(e) = 3.$$

On a 2 sommets terminaux de valeur 3, celui correspondant à l'ordre commençant par $(c, \ldots\ldots)$ et celui commençant par $(d, a, c, \ldots)$. Pour le premier :

Troisième étape : T_c : $J'(T_c) = 3$.



$$f_{T_c}(a) = 4, \quad f_{T_c}(b) = 5, \quad f_{T_c}(d) = 0, \quad f_{T_c}(e) = 6.$$

Pour le second, qui fournit nécessairement une solution puisqu'il ne reste que 2 candidats.

Quatrième étape : $T_{d,a,c}$: b $\xleftarrow[3]{}$ e $J'(T_{d,a,c}) = 3$.

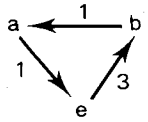
$$f_{T_{d,a,c}}(b) = 3, \quad f_{T_{d,a,c}}(e) = 0.$$

On obtient un ordre à distance 3 de T :

$$0^* = (d, a, c, e, b).$$

Si l'on cherche tous les ordres à distance 3, il faut encore examiner :

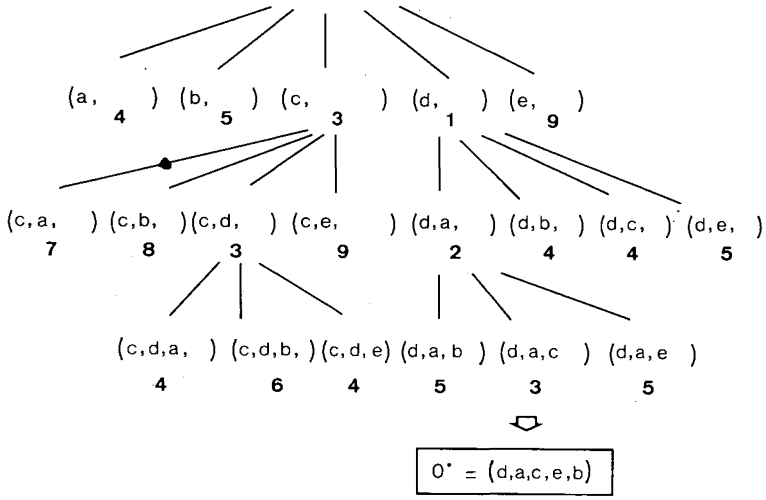
Cinquième étape : $T_{c,d}$: $J'(T_{c,d}) = 3$.



$$f_{T_{c,d}}(a) = 1, \qquad f_{T_{c,d}}(b) = 3, \qquad f_{T_{c,d}}(e) = 1.$$

Il n'en existe aucun autre.

L'arborescence construite est la suivante :



On remarquera que l'ordre 0* contredit 3 avis (c, d), (b, a), (b, c) faiblement majoritaires tandis que l'ordre $\tilde{0} = (d, b, a, c, e)$ qui minimise le nombre de contradictions, ne va à l'encontre que de 2 avis (c, d), (e, b), mais le second est fortement majoritaire et $d(T, \tilde{0}) = 4$.

5. EXPÉRIMENTATION

Cet algorithme a été programmé en FORTRAN (150 instructions environ) pour un calculateur CII 10070, dans le but de tester ses performances. On peut en obtenir un listing sur demande.

Si l'on veut évaluer le temps de calcul nécessaire à la résolution d'un tournoi et la place mémoire que nécessite ce calcul, en fonction du nombre de candidats en présence, il faut opérer de façon statistique, et calculer pour des tournois aléatoires les moyennes de ces 2 paramètres.

Les tournois aléatoires sont obtenus de la façon suivante. Pour chaque juge, on tire p nombres au hasard entre 0 et 1 (s'il y a p candidats) et la permutation qui les ordonne suivant l'ordre croissant est prise comme l'opinion de ce votant. Le tournoi au hasard est la somme de ces avis.

L'encombrement mémoire EM (en nombres de mots-machine) est fonction de la taille NM de l'arborescence développée et du nombre de candidats :

$$EM = 3 * NM + 2\,p^2 + p.$$

On concevra que les 2 derniers termes sont rapidement faibles devant le premier, lorsque p croît.

Ce temps de calcul est fonction de la transitivité du tournoi et du nombre de juges N, nous avons calculé pour N variant entre 50 et 500, par multiples de 50, un temps moyen en secondes de résolution du problème — temps qui ne tient pas compte de la détermination des opinions aléatoires —, pour des valeurs de p variant de 6 à 14, et la valeur de NM, en nombre de mots-machine.

p	6	7	8	9	10	11	12	14
Temps	0,13	0,2	0,3	0,3	0,5	3,4	6,1	40
NM	25	40	80	60	100	550	1 010	2 000

Pour les valeurs de p supérieures à 15 on ne peut calculer raisonnablement des temps de calculs car on dépasse trop souvent la taille maximale de l'arborescence (9 000 mots) ce qui arrête les opérations en cours d'exécution. Mais certains tournois — présentant des effets Condorcet — peuvent être résolus dans des temps acceptables.

BIBLIOGRAPHIE

1. K. J. ARROW, *Social Choice and Individual Value*, New York, Wiley, 1963.
2. C. BERGE, *Graphes et hypergraphes*, Dunod, Paris, 1970.
3. J.-C. BERMOND, *Ordres à distance minimale d'un tournoi et graphes partiels sans circuits maximaux*, Math. Sc. Hum., n° 37, 1972, p. 5-25.
4. G. Th. GUILBAUD et P. ROSENTHIEL, *Analyse algébrique d'un scrutin*, Ordres totaux finis, Gauthier-Villars, Paris, 1971, p. 71-100.
5. E. JACQUET-LAGRÈZE, *Le problème de l'agrégation des préférences : une classe de procédures à seuil*, Math. Sc. Hum., n° 43, 1973, p. 29-37.