

Analyse des Préférences et Tournois Pondérés

Alain Guénoche

Institut de Mathématiques de Marseille (AMU - CNRS)

*Correspondance : alain.guenoche@univ-amu.fr

DOI : 10.18713 - 170117-2-2

Soumis le 31 mai 2016 - Publié le 17 janvier 2017

Volume : 2 - Année : 2017

Titre du numéro : **thème interdisciplinaire**

Éditeurs : Rosa Figueiredo, Vincent Labatut

Résumé

Dans de nombreuses études expérimentales, on dispose de n éléments ordonnés suivant plusieurs classements (votes, notes ou critères). Nous traitons et comparons deux problèmes : (i) Etablir un classement unique (ordre total) des n items et (ii) sélectionner les k meilleurs éléments parmi n . Il s'agit, dans les deux cas, de minimiser le nombre de préférences qui vont à l'encontre de ces choix.

Mots-Clés

Préférences ; Tournoi ; Ordres médians ; Sélection de candidats

I INTRODUCTION

Les données de préférences sont induites par des ordres :

- des ordres totaux, c'est à dire des classements sur un ensemble d'individus ou d'items, tous classés, qui n'admettent pas d'ex-aequo ;
- des préordres totaux, qui admettent les ex-aequo. Ce sont donc des ordres totaux sur des classes d'équivalence, constituées des items à égalité du point de vue de l'ordre.

Ces ordres peuvent venir de votes (opinions formulées par des classements sur des individus ou des jugements d'appréciation sur des produits). Ils peuvent aussi provenir de notes attribuées aux items par des critères quantitatifs ou des épreuves comparatives. On retrouve ainsi les problèmes liés aux procédures de dépouillement de scrutins, dont l'origine remonte à Condorcet (1785) (5), aux classements hédonistes basés sur des appréciations d'experts, à la sélection de candidats pour des concours de recrutement ou au classement des élèves selon plusieurs matières.

Nous adoptons les formulations suivantes : soit X un ensemble d'items ($|X| = n$) et Π un

ensemble de m ordres ou préordres totaux, appelé *profil*¹

$$\Pi = \{O_1, O_2, \dots, O_m\}.$$

Les préférences sont les paires ordonnées d'items. On notera $x \prec_O y$ si x est avant y dans O . Un ordre est équivalent à l'ensemble de ses préférences et la distance *naturelle* entre les ordres O_i et O_j est la distance de Kendall δ :

$$\delta(O_i, O_j) = |\{(x, y) \in X^2 \text{ tels que } (x \prec_{O_i} y \wedge y \prec_{O_j} x) \vee (y \prec_{O_j} x \wedge x \prec_{O_i} y)\}|.$$

Nous abordons deux problèmes :

- (i) Etablir un ordre *consensus* le plus en accord possible avec le profil, c'est-à-dire un ordre total π sur X ;
- (ii) Sélectionner une partie de X , S de cardinal k fixé. La classe complémentaire est notée $R = X - S$, R pour rejetés.

Le principe est de minimiser le nombre de préférences des ordres du profil Π qui (i) contredisent celles de π ou (ii) celles qui vont de R vers S .

Un ordre qui minimise la somme des préférences du profil en contradiction avec cet ordre est appelé un ordre *médian*. C'est tout à fait justifié dans le cas d'un profil d'ordres totaux car, pour la distance de Kendall δ et si $T(x, y)$ est le nombre d'ordres du profil qui placent x avant y . Cet ordre consensus π est une *médiane* du profil car $\sum_{i=1, \dots, m} \delta(O_i, \pi)$ est minimum dans l'ensemble des ordres totaux sur X (1). Mais si le profil contient des préordres, ou si les pondérations des arcs ne sont pas les nombres d'ordres, un ordre à écart minimum du tournoi n'est pas assuré d'être une médiane. Néanmoins, on continuera d'employer, de manière abusive, le terme *ordre médian* pour tout *ordre à écart minimum d'un tournoi*. S'il n'est pas unique, on cherchera à énumérer tous les ordres optimaux. Pour le second problème, on construit un groupe de taille k fixée, sans chercher à ordonner les items à l'intérieur des classes.

Du point de vue méthodologique, nous allons voir que ces deux problèmes sont différents. Le premier est classique dans le cadre de la *Théorie du choix social* (vote) ou de l'*Agrégation des préférences* selon des comparaisons par paires (1). Le problème de sélection est souvent traité via un ordre total, en retenant les k éléments qui sont les mieux classés. Nous verrons que, du point de vue de la minimisation du nombre de préférences contradictoires, c'est une erreur !

Ces problèmes peuvent être posés en terme de programmation mathématique en variables entières, puisqu'il s'agit d'affecter des positions ou des rangs à n items qui introduisent des pénalités si leurs positions relatives ne respectent pas les préférences majoritaires. On voit ainsi que ces problèmes sont des cas particuliers du fameux problème d'affectation quadratique (QAP) pour lequel de nombreuses heuristiques d'approximation et des méthodes Branch & Bound ont été développées (10). L'obtention d'une borne inférieure du coût d'une affectation partielle est souvent réalisée par des méthodes de programmation mathématique (18), ce qui n'est pas le cas dans notre approche.

Les méthodes développées ci-dessous reposent sur des comparaisons par paires des éléments de X . De chaque ordre ou préordre du profil, on retient que x est meilleur que y , ou l'inverse ou encore qu'ils sont à égalité. Notons $T(x, y)$ le poids des préférences de x en faveur de y . Le

1. en référence aux profils d'opinions.

plus souvent, $T(x, y)$ est le nombre d'ordres dans Π qui préfèrent x à y ². Si $T(x, y) > T(y, x)$ la préférence de x en faveur de y est dite *majoritaire*, ce qui correspond à la terminologie des votes.

Au profil Π on associe un *tournoi* (graphe complet orienté de sommets X) dont les arcs correspondent aux préférences majoritaires. Entre les sommets x et y on met un arc (x, y) de x vers y si et seulement si $T(x, y) > T(y, x)$. Cet arc a pour poids la différence $w(x, y) = T(x, y) - T(y, x)$ et, dans la table W des poids, on pose $w(y, x) = 0$. Si $T(x, y) = T(y, x)$, on peut mettre l'arc dans n'importe quel sens, puisque $w(x, y) = w(y, x) = 0$.

Tout ordre total sur les items $O = (x_1, x_2, \dots, x_n)$ distingue deux types d'arcs ; les *arcs directs* qui sont orientés selon l'ordre O et les *arcs-retour* qui sont orientés dans le sens opposé. Soit $W(O)$ la somme des poids de ces arcs-retour qui mesure l'*écart* de l'ordre O au tournoi. Cet ordre est d'autant plus compatible avec le tournoi que $W(O)$ est faible³. Mais que faut-il compter dans W ?

- Dans le premier problème, on considère tous les arcs-retour et on cherche donc un ordre total sur X qui minimise

$$W_o(O) = \sum_{i < j} w(x_j, x_i).$$

- Dans le second cas, on cherche seulement deux classes, une bipartition orientée, avec d'un côté les sélectionnés S et de l'autre les rejetés R et on ne compte que les arcs-retour qui vont de R vers S .

$$W_p(S|R) = \sum_{x \in S, y \in R} w(y, x)$$

Pour construire des ordres médians ou une classe d'items sélectionnés, on a recouru à des méthodes d'énumération. Pour améliorer leur efficacité il est nécessaire de disposer de bornes supérieures aux optima que l'on cherche à atteindre. Elles sont calculées à l'aide d'ordres dont l'écart au tournoi est optimisé de façon approchée. Dans la section 2, on part de deux heuristiques classiques pour introduire une optimisation locale et une méthode basée sur une décomposition du tournoi. Elles sont originales et permettent d'améliorer toute solution approchée.

Dans la section 3 nous reprenons la méthode *Branch & Bound* développée dans (11; 2; 7) et nous l'adaptions au problème de l'énumération de tous les ordres médians. L'étude d'un petit tournoi, correspondant au choix de conférenciers invités lors d'un colloque et qui admet plusieurs ordres médians, nous amène à étudier la question du choix d'un de ces ordres. La section 4 est dédiée au problème de sélection de k éléments selon des ordres, que l'on résout aussi par énumération.

II DES HEURISTIQUES POUR UN "BON" ORDRE

La construction d'un ordre optimal à partir d'un tournoi résulte d'une méthode *Branch & Bound* qui nécessite la connaissance d'une borne supérieure de l'écart d'un ordre au tournoi. Ces bornes sont indispensables pour établir un ordre médian, car elles permettent de limiter la taille de l'arborescence de recherche. Nous ne reprenons que deux heuristiques classiques, la méthode gloutonne et une méthode de score avant de présenter nos améliorations. La méthode de Smith

2. On peut aussi tenir compte de l'écart entre x et y , la différence des rangs ou, dans le cas de mesures quantitatives, des différences entre les notes de x et de y . Dans ce cas, $T(x, y)$ est la somme des différences positives entre les valeurs de x et de y et une procédure de normalisation des notes est préconisée

3. Un tournoi qui admet un ordre O tel que $W_o(O) = 0$ est dit *transitif*.

& Payne (21) (retournement de l'arc qui supprime le plus de 3-circuits), étendue aux tournois pondérés ((2)) s'est avérée sans efficacité pour les problèmes pondérés, surtout quand leur taille n augmente.

Il existe d'autres approches qui relèvent de l'optimisation stochastique, à commencer par le Recuit Simulé (Metropolis), l'optimisation en voisinage variable (12) ou la méthode de bruitage (9). Les paramètres à prendre en compte, les temps de calcul et les codes propres à leurs auteurs font que nous ne les considérons pas dans cette étude. Bien évidemment, toute solution heuristique peut servir de point de départ à toute procédure d'optimisation.

2.1 La méthode gloutonne

Elle procède du même principe que la procédure *Branch & Bound*, sauf qu'elle ne développe pas d'arborescence des sections commençantes et choisit à chaque itération l'item qui promet un ordre de moindre coût. On calcule donc les sommes colonnes de la table des poids du tournoi $Sum(y) = \sum_{x \in X} w(x, y)$, contribution de y à l'écart d'un ordre commençant par x .

$i=0$; Tant qu'il existe un élément non placé

- soit x non placé tel que $Sum(x)$ est minimum
- $i++$; $o_i = x$; x est placé ;
- pour tout y non placé
 - $Sum(y) = Sum(y) - w(x, y)$;

Cette heuristique est clairement en $O(n^2)$.

2.2 Les méthodes de score

Ce sont des méthodes basées sur des fonctions de score. Chaque ordre attribue des points à chaque item et le score d'un item est la somme éventuellement pondérée, des points acquis. Suivant la signification des points, l'ordre consensus est l'ordre croissant ou décroissant des scores. L'exemple type est la procédure de Borda ((4)), rival de Condorcet, pour désigner les vainqueurs d'une élection. Son score est le rang du candidat (1 point pour le premier, 2 pour le second, etc) et l'ordre final est l'ordre croissant de la somme des rangs. Dans le cas de préordres, il faut attribuer à chaque item le rang moyen des ex-aequo.

Un autre exemple classique est le championnat des pilotes de Formule 1 ; chaque course attribue des points (non proportionnels) aux dix premiers, et 0 pour les autres. Le classement est l'ordre décroissant de la somme des points acquis par les pilotes.

2.3 Une optimisation locale

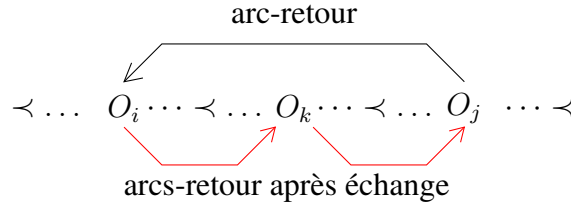
Etant donné un ordre $O = (o_1, o_2, \dots, o_n)$ sur X , la procédure d'optimisation locale cherche par échange d'items un ordre d'écart au tournoi plus petit. Pour tout élément o_j , soit o_i le dernier élément placé avant lui tel que $w(o_j, o_i) > 0$; on a donc $i < j$, et (o_j, o_i) est le plus petit arc-retour partant de o_j . Le cas où o_i et o_j sont consécutifs correspond à une procédure très classique : en transposant ces deux éléments, on diminue l'écart au tournoi, puisqu'on supprime un arc-retour. Sinon, on a intérêt à échanger o_j et o_i si et seulement si les items placés entre o_i et o_j ne créent pas d'arcs-retour de poids plus grand. C'est ce que l'on vérifie en sommant les valeurs des arcs-retour après échange.

Soit $O = (o_1, o_2, \dots, o_n)$ un ordre total.

Tant qu'il y a échange

- Pour $j=2$ à n

- soit (o_j, o_i) le plus petit arc retour d'origine o_j
- $Som = \sum_{i>k>j} w(o_k, o_j) + w(o_i, o_k)$
- si $w(o_j, o_i) > Som$ échanger o_i et o_j



Ainsi l'intervalle $(o_i, o_{i+1}, \dots, o_{j-1}, o_j)$ devient $(o_j, o_{i+1}, \dots, o_{j-1}, o_i)$. Cette procédure est appliquée itérativement tant qu'il existe des transpositions de ce type à réaliser. Pour tout item, il suffit de remonter vers le dernier item précédent dominé et d'effectuer des sommations dans l'intervalle ainsi défini. Chaque itération est en $O(n^2)$ dans le cas le pire (celui de l'ordre induit par un tournoi transitif pour lequel on cherche en vain un intervalle à retourner).

Dans la suite de ce texte, nous appelons *BestH* la meilleure des deux heuristiques après application de la procédure d'optimisation locale.

2.4 Décomposition d'un tournoi

Devant des problèmes de grande taille ($n \gg 30$), les heuristiques classiques sont peu efficaces. D'où l'idée de décomposer le tournoi en sous-tournois, c'est à dire partitionner les items de façon que chaque classe contiennent des éléments voisins dans un "bon" ordre. Il suffit alors de calculer un ordre consensus pour chaque classe et de les concaténer pour obtenir un ordre total, certes approché, mais plus proche du tournoi que les heuristiques globales. Nous montrons, à l'aide de simulations sur des tournois aléatoires, que cette stratégie de décomposition est efficace.

2.4.1 Une décomposition régulière

L'ordre donné par l'une des heuristiques précédentes induit une décomposition régulière. Etant donné un nombre de classes q , il suffit de construire les classes comme des intervalles de cet ordre. Les n/q premiers vont dans la première classe, les n/q suivants dans la seconde, etc. On obtient ainsi une partition en classes équilibrées, notée P_R .

2.4.2 Une partition des items basée sur une distance

En considérant la table W des poids des arcs du tournoi, on peut associer à chaque item x une bipartition : Soit x_+ , les éléments de X qui devraient être placés avant x parce qu'ils le dominent et x_- , qui devraient être placés après x parce qu'il les domine.

$$x_+ = \{z \in X | w(z, x) > 0\} \text{ et } x_- = \{z \in X | w(x, z) > 0\}.$$

A l'aide de ces bipartitions, on construit un indice de dissimilarité sur X ,

$$D(x, y) = \Delta(x_+, y_+) + \Delta(x_-, y_-),$$

dans lequel Δ est la distance de la différence symétrique : $\Delta(A, B) = |\{x \in \{A \setminus B \cup B \setminus A\}\}|$. On notera que D n'est pas une distance, car il se peut que $D(x, y) = 0$ si $w(x, y) = w(y, x) = 0$.

Proposition Si T est un tournoi transitif,

- deux éléments consécutifs dans l'ordre optimal ont une distance égale à 2 ;
- $D(x, y)$ est proportionnelle à la différence des rangs de x et y dans l'ordre médian.

Preuve

Soit $x \prec y$ deux éléments consécutifs dans l'ordre correspondant à un tournoi transitif. Les classes x_+ et y_+ (resp. x_- et y_-) ne diffèrent que d'un seul élément, x (resp. y) et donc $D(x, y) = 2$. De même, si x et y sont séparés par k items dans l'ordre médian, $D(x, y) = 2(k + 1)$. Ainsi les valeurs sont croissantes quand on s'écarte de la diagonale, et D est une distance (car il n'y a pas d'indifférence dans les préférences).

Ainsi, les faibles valeurs de D correspondent à des éléments voisins dans un ordre médian et les classes obtenues doivent contenir des éléments consécutifs. Le choix du nombre de classes q sera conduit par une procédure de simulation. Pour construire, à partir de D une partition en q classes, notée P_M , on cherche à minimiser la somme des distances intra-classe par une procédure itérative à la manière des k -means. On part de la partition atomique qui ne contient que des singletons. A chaque itération un item est transféré vers la classe pour laquelle sa moyenne de distance est minimum. La procédure s'arrête quand il n'y a plus d'item à déplacer.

Algorithme de décomposition

Soit $O = (o_1, o_2, \dots, o_n)$ un ordre total sur X

1. Découper X en q classes ordonnées $\{X_1, \dots, X_q\}$
 - soit en classes consécutives équilibrées suivant O
 - soit en classes calculées suivant la distance D
2. Calculer un ordre consensus O_i pour chaque classe X_i
3. Concaténer les ordres O_1, \dots, O_q suivant le rang moyen des éléments de X_i dans O
4. Appliquer la procédure d'optimisation locale

On obtient ainsi un *ordre composé* noté $Comp_R$ (resp. $Comp_M$), suivant la méthode de partitionnement choisie, et son écart au tournoi est calculé.

2.4.3 Complexité, efficacité

Le calcul de la table des $O(n^2)$ valeurs de distances est en $O(n^3)$, puisque pour chaque paire d'items on compare les positions relatives de n éléments. L'algorithme de décomposition régulière est linéaire. La partition P_M est obtenue par un algorithme itératif, dont on ignore le nombre d'itérations (comme pour les k -means), mais qui est bien connu pour son efficacité. Après, on applique l'heuristique gloutonne à chaque classe et la procédure d'optimisation à l'ordre composé.

Notons que la méthode de l'*Ordre composé* est rapide. Il faut compter 1''20 pour P_R , 19''30 pour P_M sur un tournoi à 1000 items sur un MacPro portable.

2.4.4 Simulations et résultats

Sur des profils aléatoires

On génère des profils d'ordres totaux en tirant au hasard des permutations d'ordre n (17). Chaque profil permet de calculer le tournoi avec les poids des arcs (matrice W). Après application de l'optimisation locale, on dispose de deux ordres totaux. La meilleure des solutions, notée BestH, est celle qui minimise l'écart au tournoi.

On se fixe alors le nombre de classes q . Pour le partitionnement des items, on calcule D avec les bipartitions $P_x = (x_+ | x_-)$. On a :

- la partition régulière (P_R) en suivant l'ordre donné par BestH qui aboutit à l'ordre $Comp_R$;

- une partition (P_M) qui cherche à minimiser les moyennes des distances intra classes et qui aboutit à l'ordre $Comp_M$;

Pour BestH et chacun de ces ordres, on calcule leur écart au tournoi.

Les tests portent sur 100 profils de mêmes paramètres ; à nombre d'items fixé n , on peut faire varier le nombre d'ordres totaux dans le profil m et le nombre de classes de la décomposition q . Chaque ligne de la Table 1 donne l'écart moyen aux tournois ainsi générés.

Il est clair que les ordres composés sont meilleurs que la meilleure des heuristiques classiques. Soulignons que ce sont des écarts moyens et que, pour un profil ou pour un tournoi particulier, les deux décompositions peuvent être testées. Mais on ne sait pas quelle méthode est la plus efficace, ni combien de classes assurent un écart minimum. La partition P_M semble donner les meilleurs résultats, mais peut être que le nombre de classes est trop petit, comme le suggère le nombre croissant de classes pour $n = 500$.

n	m	q	BestH	$Comp_R$	$Comp_M$
50	10	2	153	152	151
50	20	3	259	251	249
50	30	4	330	319	318
100	10	3	755	731	719
100	20	3	1236	1203	1187
100	30	3	1569	1534	1522
200	30	4	6825	6678	6614
200	50	5	9047	8820	8784
200	100	6	13129	12762	12782
500	100	5	86600	85336	85107
500	100	10	86600	84670	85635
500	100	15	86600	84636	86078

TABLE 1 : Valeurs moyennes des écarts obtenus par les heuristiques sur des tournois générés par des permutations aléatoires.

Sur des tournois à écart borné

En tirant au hasard des ordres totaux indépendants, on génère des tournois qui sont loin d'être transitifs et donc les ordres obtenus sont d'écart élevé. Et il est impossible de savoir s'ils sont proches de l'optimum. D'un certain point de vue, il est inutile de chercher un ordre consensus à ces profils, puisqu'aucune tendance commune ne peut émerger. Nous traçons des tournois générés à partir d'un ordre unique en transposant un certain nombre d'items. Soit t le paramètre qui définit le nombre de transpositions.

On génère m ordres totaux à partir de l'ordre naturel en échangeant, dans chacun, t paires d'items aléatoirement choisis. Le tournoi pondéré est alors calculé selon la procédure majoritaire classique. Si t est faible par rapport à n , la tendance générale est alors l'ordre naturel, qui a toutes les chances d'être médian.

L'essai porte sur deux familles d'ordres ; des ordres à 100 (resp. 300) items, sur lesquels on effectue 20 (resp. 100) transpositions avant de calculer le tournoi pondéré. Les décompositions sont en 5 (resp. 10) classes.

n	m	t	q	$BestH$	$Comp_R$	$Comp_M$	$OrdNat$
100	10	20	5	81.7	73.1	66.3	111.9
100	20	20	5	22.6	21.1	19.5	23.9
100	30	20	5	5.3	5.1	5.2	5.9
300	30	100	10	2065	1721	1434	1555
300	50	100	10	800	677	558	544
300	100	100	10	56.1	50.6	49.1	45.9

TABLE 2 : Valeurs moyennes des écarts obtenus par les heuristiques sur des tournois à écart borné.

Plus le nombre d'ordres est grand, plus le tournoi est transitif et l'ordre naturel proche de l'optimal. La seconde méthode de décomposition s'avère très efficace quand on s'en éloigne.

2.5 Conclusion

La stratégie de décomposition est toujours gagnante et donc il est préférable d'assembler des ordres obtenus sur les sous-tournois en un ordre unique, plutôt que de traiter le tournoi d'un seul bloc. De plus, pour des tournois "presque transitifs", les ordres obtenus par décomposition restent proches de l'ordre optimal.

Donc, pour un tournoi particulier, je commencerai par chercher le nombre optimal de classes à l'aide du découpage régulier qui est très rapide, puisqu'il n'y a même pas de distance à calculer. Et autour de cette valeur, je testerai les algorithmes de partitionnement. Un dernier essai sur un tournoi à 1000 items a donné par une décomposition régulière en 15 classes le plus faible écart, et ce très nettement.

III ENUMÉRATION DES ORDRES OPTIMAUX

Construire un ordre total π dont l'écart au tournoi $W_o(\pi)$ est minimum (le problème de Kemeny (15)), revient à rendre le tournoi transitif par retournement d'arcs, ceux dont la somme des poids est minimum. Ce problème, *minimum feedback arc set*, a une longue histoire algorithmique, qui débute avec Slater (19), Ramage & Thomson (20), Bermond (3) pour les tournois non pondérés, puis tout un groupe centré autour du Centre d'Analyse et de Mathématiques Sociales (CNRS, EHESS) avec (16; 11; 2; 7; 14) pour les tournois pondérés.

Depuis Hudry (13), le problème a été prouvé NP-difficile. Le calcul s'effectue par une méthode de *Branch & Bound* (11) dont nous rappelons tout d'abord le principe. On développe une arborescence dont les sommets sont des *sections commençantes*. Ce sont des débuts d'ordres totaux qui peuvent se prolonger en un ordre total optimal. C'est à dire que le poids de cette section est au plus égal à la borne supérieure B_o calculée par l'une des heuristiques précédentes.

Une feuille de l'arborescence courante est donc une chaîne (o_1, o_2, \dots, o_k) sur une partie $X_s \subset X$ et notons $Y = X - X_s$ l'ensemble des éléments non classés. Le poids W_o^k de cette section est égal à la somme des poids des arcs-retour internes à cette section plus la somme des poids des arcs dont l'origine y est hors cette section et l'extrémité dans cette section.

$$W_o^k(o_1, o_2, \dots, o_k) = \sum_{1 \leq i < j \leq k} w(o_j, o_i) + \sum_{y \in Y} \sum_{j=1, \dots, k} w(y, o_j).$$

Cette valeur est inférieure ou égale au poids de tout ordre total ayant cette section commençante, puisqu'elle ne tient pas compte de l'ordre non encore déterminé sur Y .

En prolongeant l'arborescence, à chaque itération, par la feuille de poids minimum, on aboutit à un ordre total sur X dont l'écart est minimum. Sa valeur est déterminée par le premier ordre total obtenu. Si le tournoi contient beaucoup de circuits, cette méthode peut devenir inapplicable et ce dès que n atteint quelques dizaines d'items (2). Pour deux raisons ; l'une est la taille de l'arborescence qui peut dépasser le million de noeuds⁴ et l'autre le temps de calcul, à cause de la recherche de la feuille de l'arborescence de poids minimum que l'on prolonge d'abord pour être sûr d'aboutir à un premier ordre total optimum. Cette difficulté a été contournée par Woïgard (22) en adaptant une structure de données plus élaborée.

De façon plus efficace, pour obtenir tous les ordres médians, au lieu de développer l'arborescence à partir d'une feuille de moindre poids (stratégie du meilleur d'abord), on la développe en largeur d'abord, ce qui évite d'avoir à chercher à chaque pas la feuille de poids minimum.

Algorithme d'énumération des ordres médians

Soit B_0 une borne supérieure de l'écart d'un ordre total au tournoi et A une arborescence initiale de racine \emptyset à laquelle sont attachés les éléments de X tels que $Sum(x) = \sum_{y \in X} w(y, x) \leq B_0$.

Pour toute feuille S_k de A

1. $S_k = (o_1, o_2, \dots, o_k)$ d'écart W_o^k ;
2. Pour tout y non placé dans S_k
 - $W_o^{k+1} = W_o^k + \sum_{z \in X - S_k} w(z, y)$
 - Si $W_o^{k+1} \leq B_0$ prologer A avec la section $(o_1, o_2, \dots, o_k, y)$, attachée à S_k

La taille de l'arborescence dépend fortement de la qualité de B_0 . De fait, on énumère tous les ordres totaux dont l'écart est inférieur à B_0 , borne que l'on met à jour au fil de l'énumération, s'il y a lieu. On peut même fixer une borne inférieure à B_0 pour laquelle l'arborescence n'aboutit à aucun ordre si elle est inférieure à l'écart minimum, et ce très rapidement.

3.1 Le tournoi Jobim 2016

Nous prenons comme exemple tout au long de ce paragraphe le choix de conférenciers invités pour le colloque national de bioinformatique, Jobim. Chaque membre du comité de programme désigne les conférenciers qu'il voudrait faire venir et il les range dans un ordre préférentiel. Les membres du comité ne se sont pas trop dispersés et ont nommé 10 candidats ; nous avons retenu ceux qui étaient dans au moins la moitié des ordres. Il est resté $n = 7$ candidats (les 3 autres étant éliminés). Dans le profil, nous avons repris les ordres donnés. Pour chacun, on a enlevé les éliminés et ajouté, en fin de classement et tous ex-aequo, les candidats non retenus pour obtenir un préordre total.

1. $(x_1 \prec x_2 \prec x_3 \prec x_4 \prec x_5, x_6, x_7)$
2. $(x_5 \prec x_1 \prec x_2 \prec x_4 \prec x_6 \prec x_3, x_7)$
3. $(x_7 \prec x_6 \prec x_2 \prec x_3 \prec x_1, x_4, x_5)$
4. $(x_5 \prec x_3 \prec x_4 \prec x_1, x_2, x_6, x_7)$
5. $(x_4 \prec x_7 \prec x_5 \prec x_6 \prec x_2 \prec x_1, x_3)$
6. $(x_6 \prec x_1 \prec x_7 \prec x_2 \prec x_3, x_4, x_5)$

4. Pour chaque noeud, selon une structure de données élaborée dans (11), il suffit de mémoriser 3 entiers.

TABLE 3 : Préordres totaux des 6 membres du comité de programme.

Le tournoi majoritaire pondéré résultant de ces préférences est donné dans la Table 4.

W	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	-	1	1	1	0	0	1
x_2	0	-	4	2	0	0	0
x_3	0	0	-	1	0	0	0
x_4	0	0	0	-	0	2	2
x_5	1	0	1	0	-	1	0
x_6	1	1	2	0	0	-	0
x_7	0	1	1	0	1	0	-

TABLE 4 : Tournoi majoritaire résultant des préordres de la Table 3.

On constate que tous les candidats sont dominés au moins une fois. Il n'y a donc pas de *vainqueur de Condorcet*. Pour déterminer un ordre total ou une bipartition de poids minimum on applique l'heuristique gloutonne, la taille du tournoi ne nécessitant pas une décomposition. Ici, le sommet le moins dominé est x_5 ; un ordre commençant par x_5 n'aura qu'un seul arc-retour de poids 1 (x_7, x_5). Ce sommet étant "placé", on retient x_1 qui n'entraîne qu'un autre arc-retour de poids 1 (x_6, x_1). Et ainsi de suite. On aboutit à l'ordre glouton :

$$O_g = (x_5 \prec x_1 \prec x_2 \prec x_4 \prec x_6 \prec x_7 \prec x_3),$$

et $W_o(O_g) = 5$ avec $\{(x_7, x_5), (x_6, x_1), (x_6, x_2), (x_7, x_2), (x_3, x_4)\}$, comme arcs-retour, tous de poids 1, donc $B_o = 5$. Un ordre optimal est au plus à écart 5 du tournoi.

La procédure de Borda appliquée aux préordres initiaux (en donnant un rang moyen aux candidats à égalité) conduit à deux paires d'ex-aequo $\{x_1, x_5\}$ et $\{x_4, x_7\}$. En respectant les orientations des arcs du tournoi, on aboutit à :

$$O_b = (x_2 \prec x_5 \prec x_1 \prec x_6 \prec x_4 \prec x_7 \prec x_3)$$

qui présente 6 arcs-retour et $W_o(O_b) = 7$, un écart supérieur à O_g .

La procédure de recherche d'un ordre médian aboutit à un autre ordre O_m à écart 5, ce qui confirme l'optimalité de B_o .

$$O_m = (x_4 \prec x_7 \prec x_5 \prec x_6 \prec x_1 \prec x_2 \prec x_3).$$

On lance donc la procédure d'énumération qui donne les 13 ordres médians suivant :

1. $O_1 = (x_1 \prec x_2 \prec x_4 \prec x_7 \prec x_5 \prec x_6 \prec x_3)$
2. $O_2 = (x_1 \prec x_4 \prec x_7 \prec x_5 \prec x_6 \prec x_2 \prec x_3)$
3. $O_3 = (x_4 \prec x_5 \prec x_6 \prec x_1 \prec x_7 \prec x_2 \prec x_3)$
4. $O_4 = (x_4 \prec x_7 \prec x_5 \prec x_6 \prec x_1 \prec x_2 \prec x_3)$
5. $O_5 = (x_5 \prec x_1 \prec x_2 \prec x_4 \prec x_6 \prec x_7 \prec x_3)$
6. $O_6 = (x_5 \prec x_1 \prec x_2 \prec x_4 \prec x_7 \prec x_6 \prec x_3)$
7. $O_7 = (x_5 \prec x_1 \prec x_4 \prec x_6 \prec x_7 \prec x_2 \prec x_3)$
8. $O_8 = (x_5 \prec x_1 \prec x_4 \prec x_7 \prec x_6 \prec x_2 \prec x_3)$
9. $O_9 = (x_5 \prec x_4 \prec x_6 \prec x_1 \prec x_7 \prec x_2 \prec x_3)$
10. $O_{10} = (x_5 \prec x_6 \prec x_1 \prec x_2 \prec x_3 \prec x_4 \prec x_7)$

11. $O_{11} = (x_5 \prec x_6 \prec x_1 \prec x_2 \prec x_4 \prec x_7 \prec x_3)$
12. $O_{12} = (x_5 \prec x_6 \prec x_1 \prec x_7 \prec x_2 \prec x_3 \prec x_4)$
13. $O_{13} = (x_7 \prec x_5 \prec x_6 \prec x_1 \prec x_2 \prec x_3 \prec x_4)$

TABLE 5 : Ordres médians à écart 5 du tournoi de la Table 4.

3.2 Simulations

Est-ce que ces difficultés sont fréquentes ou exceptionnelles ? Pour en avoir le coeur net, nous avons tiré au hasard des profils d'ordres totaux. Pour obtenir des chiffres crédibles pour les invités d'un colloque, nous avons fixé à $n = 20$ le nombre de candidats et à $m = 31$ le nombre d'experts, un nombre impair pour éviter les égalités et donc les arcs de poids nul.

Le plus simple est de partir d'ordres aléatoires, de calculer le tournoi majoritaire, puis d'énumérer tous les ordres médian et de compter le nombre moyen de solutions en répétant 100 fois ce tirage. Mais en tirant des ordres aléatoires, on simule une situation bien improbable ; qu'il n'y ait pas de consensus véritable entre les experts non plus que de renommée relative entre les candidats. Pour graduer ce degré d'entente entre les experts, nous sommes partis de l'ordre naturel et simulons les votes en effectuant t échanges de deux items tirés au hasard, pour tous les ordres. Ainsi, si $t = 0$ tout le jury est d'accord, et quand t croît on s'approche du désordre aléatoire qui correspond à $t = 30$.

Pour ces quatre jeux d'essais, $t = 10, 15, 20, 30$, nous tirons donc 100 profils et comptons la valeur moyenne d'écart donnée par l'heuristique gloutonne (O_g), l'ordre de Borda (O_b), un ordre médian (O_m), le nombre moyen d'ordres optimaux ($NbOpt$), le nombre maximum ($NbMax$), la taille moyenne de l'arborescence ($Tree$), ainsi que son maximum ($Tree_{Max}$).

t	O_g	O_b	O_m	$NbOpt$	$NbMax$	$Tree$	$Tree_{Max}$
10	9.9	14.8	9.2	2.1	18	187	6695
15	23.2	27.2	20.5	3.3	28	2116	53544
20	35.0	38.7	30.2	4.0	40	12331	964554
30	42.5	47.0	36.7	4.1	40	47766	1391501

TABLE 6 : Ecart moyen des heuristiques, d'un ordre médian, les nombres moyens et maximum d'ordres optimaux et les nombres moyens et maximum de nœuds de l'arbre de recherche.

La différence entre l'écart donné par l'heuristique et l'écart optimal paraît faible, mais il y a beaucoup d'ordres entre ces deux valeurs, ce qui souligne la supériorité (en moyenne) de O_g sur O_b . Le nombre moyen de solutions optimales reste limité, mais le nombre maximum observé est important et justifie le paragraphe ci-dessous pour décider d'un classement unique des candidats. Enfin la taille de l'arbre est importante, mais les calculs sont très rapides, quelques secondes pour les 100 problèmes les plus difficiles.

3.3 Choisir parmi les ordres médians

En présence de plusieurs ordres médians on aimerait n'en garder qu'un pour établir un ordre consensus unique. La tâche n'est pas facile. Sur les 13 ordres optimaux du tournoi Jobim, il y a 4 premiers, x_1, x_4, x_5 et x_7 , dont deux d'entre eux peuvent être derniers. Tous les candidats, sauf x_3 , sont dans les deux premiers rangs. Voici quelques solutions :

3.3.1 *Le plus central des ordres médians*

De tous ces ordres, quel est le plus central ? C'est celui dont la somme des distances aux autres ordres médians est minimum. Il suffit de calculer la distance de Kendall (nombre de paires d'items en désaccord entre deux ordres) et d'en faire les sommes ; c'est O_7 le plus central.

3.3.2 *Le plus souvent dans les k premiers*

Si l'on avait un seul invité, on prendrait x_5 parce qu'il est le plus souvent le premier dans l'ensemble des ordres médians. Et en seconde position vient celui qui, en dehors de x_5 est le plus souvent premier ou second ; c'est x_1 . Et ainsi de suite ; on place au rang k un candidat non placé qui est le plus souvent (plus que les autres) à un rang inférieur ou égal à k . Cette procédure fonctionne comme un scrutin uninominal à un tour qui est répété k fois. Ici, on aboutit encore au même ordre O_7 .

Ce n'est pas équivalent à la procédure de Borda appliquée aux ordres médians, même si ici elle donne le même résultat, car on ne tient aucun compte des mauvais classements. Par contre on n'aboutit pas nécessairement à un ordre médian, c'est pourquoi je préconise l'autre solution au choix d'un ordre unique.

IV A LA RECHERCHE D'UNE SÉLECTION

Dans ce paragraphe, nous cherchons à sélectionner les k meilleurs éléments parmi n en minimisant le nombre de préférences qui vont à l'encontre de ce choix. Précisons qu'ici le paramètre k est fixé. Les arcs-retour entre les sélectionnés S ou entre les rejetés R ne comptent pas, puisqu'ils ne contredisent en rien la sélection finale. On cherche donc deux classes et on ne compte que les arcs-retour qui vont de la classe R vers S .

$$W_p(S|R) = \sum_{x \in S, y \in R} w(y, x)$$

Entre deux classes complémentaires de sommets dans un graphe orienté, on parle de *cocircuit*. On cherche donc un cocircuit dont les parties sont de cardinal fixé et dont le poids des arcs-retour est minimum.

Du point de vue méthodologique, ce problème est différent de celui de l'ordre consensus. Souvent, le problème de sélection est traité via un classement de tous les items, en retenant les k items les mieux classés. Nous verrons que, pour minimiser W_p , c'est une erreur ! Comme précédemment, un ordre total sur X dont l'écart au tournoi soit faible est nécessaire car il fournit une borne supérieure au critère W_p . Cet ordre permet de fixer le poids minimum d'un cocircuit d'arcs-retour sur une classe S de cardinal k , notée B_p^k .

Pour illustrer ce paragraphe, nous reprenons le tournoi des conférenciers invités au colloque Jobim, avec $k = 2$. Examinons tout d'abord les ordres donnés par les heuristiques classiques :

Selon l'ordre glouton $O_g = (x_5 \prec x_1 \prec x_2 \prec x_4 \prec x_6 \prec x_7 \prec x_3)$, une bipartition avec deux invités, x_1 et x_5 donnent $W_p^2(x_1, x_5 | x_2, x_3, x_4, x_6, x_7) = 2$ (avec les arcs-retour $\{(x_6, x_1), (x_7, x_5)\}$), donc $B_p^2 = 2$.

L'ordre de Borda $O_b = (x_2 \prec x_5 \prec x_1 \prec x_6 \prec x_4 \prec x_7 \prec x_3)$ suggère une partition à deux invités, avec $W_p^2(x_2, x_5 | x_1, x_3, x_4, x_6, x_7) = 4$, le double du précédent.

4.1 Une sélection optimale

Ce problème semble encore plus difficile⁵ que celui de l'ordre médian, car la construction de cette bipartition n'est pas cumulative. Bien sûr, on est tenté de prendre comme "premier" invité le candidat le moins dominé. Mais il se peut qu'un candidat plus dominé puisse être retenu par la suite avec celui qui le domine fortement. La cause en est la non monotonie de la fonction de poids W_p quand le nombre de sélectionnés augmente, alors que W_o l'était quand on prolongeait les sections commençantes.

Prenons l'exemple du petit tournoi pondéré de la Table 7 :

W	x_1	x_2	x_3	x_4	x_5	x_6
x_1	-	5	0	1	1	1
x_2	0	-	5	1	1	1
x_3	5	0	-	1	1	1
x_4	0	0	0	-	1	1
x_5	0	0	0	0	-	1
x_6	0	0	0	0	0	-

TABLE 7 : Tournoi pour illustrer la non monotonie de W_p^k quand k augmente.

S'il n'y a qu'un invité, c'est x_4 de poids 3, s'il y en a deux, c'est l'une des trois paires $\{x_1, x_2\}$, $\{x_1, x_3\}$, $\{x_2, x_3\}$ de poids 5. Mais si l'on en prend trois, c'est $\{x_1, x_2, x_3\}$ car cette partie domine tous les autres et $W_p^3(x_1, x_2, x_3 | x_4, x_5, x_6) = 0$.

Donc, si l'on veut un seul item, il suffit de comparer les sommes colonnes de la table du tournoi. Pour 2 ou 3 éléments, il faut effectuer quelques boucles imbriquées. Mais si l'on veut une partie à $k > 3$ éléments, je ne vois pas de méthode optimale autre que l'énumération des parties à k éléments parmi N .

4.2 Des bornes efficaces

Heureusement, cette énumération ne porte pas nécessairement sur toutes les parties de X à k éléments. D'abord un ordre total construit à l'aide d'une heuristique fournit une borne supérieure B_p^k au poids du cocycle retour. Ensuite, pour chaque item x , on peut établir deux bornes, $B_S(x)$ (resp. $B_R(x)$) correspondant aux poids minimum des arcs-retour si $x \in S$ (resp. $x \in R$). Si x est sélectionné ($x \in S$), il peut être extrémité d'éventuels arcs-retour. Il y en a au plus $N - k$ dont le poids est supérieur ou égal à la somme des $N - k$ plus petites valeurs des arcs d'extrémité x , c'est à dire les $N - k$ plus petites valeurs de la colonne x de la table W :

$$B_S(x) = \sum_{i=1, \dots, N-k} \text{Min}\{w(y_i, x)\}_{y_i \neq x}.$$

De même, si x est rejeté ($x \in R$), il peut être à l'origine de k arcs-retour, soit au minimum la somme des k plus petites valeurs de la ligne de x .

$$B_R(x) = \sum_{i=1, \dots, k} \text{Min}\{w(x, y_i)\}_{y_i \neq x}.$$

Ces deux bornes se calculent facilement par tri des lignes et des colonnes de W . Elles permettent souvent d'affecter x à l'une ou l'autre classe si B_S ou B_R dépassent la borne supérieure B_p^k connue pour une bipartition avec k sélectionnés. Comme nous le verrons dans les simulations

5. Non pas au sens de la théorie de la complexité, mais du calcul pratique

ci-dessous, c'est ce qui advient pour plus de la moitié des candidats. On peut donc procéder à une énumération sur un nombre restreint d'items parmi lesquels on ne cherche plus qu'un nombre petit de candidats pour compléter ceux qui sont sélectionnés d'office.

Pour une énumération efficace des parties de cardinal fixé d'un ensemble, Nijenhuis & Wilf (17) proposent un algorithme qui, pour passer d'une partie à la suivante, supprime et ajoute un seul élément. Ainsi, le poids de la nouvelle partie peut être calculé à partir du poids de l'ancienne en $O(n)$. Mais la complexité de cette énumération reste liée à $\binom{n}{k}$.

Pour le tournoi Jobim, s'il n'y a qu'un seul invité, x_5 est le seul à recevoir un arc de poids minimum. Si l'on veut deux invités, x_1 et x_5 n'en reçoivent que deux, et c'est la seule paire qui réalise ce score. Mais s'il y a trois invités, on trouve deux triplets de score 3, $\{x_1, x_5, x_6\}$ et $\{x_1, x_5, x_7\}$; le second n'est même pas un début d'ordre médian !

4.3 Simulations again

Ici aussi, nous avons voulu (i) comparer les heuristiques, (ii) tester la difficulté du calcul et (iii) savoir si les bornes permettent d'affecter directement des candidats. Pour les mêmes tournois que précédemment, avec 31 juges et 20 candidats, nous cherchons à sélectionner 5 invités. Nous mesurons le poids moyen des cocircuits-retour obtenus par l'heuristique gloutonne, par l'ordre de Borda, puis l'énumération fournit la valeur optimale. Nous indiquons de plus le nombre moyen de candidats dont l'affectation est fixée par les bornes ($NbFix$), le nombre moyen de solutions optimales et le maximum obtenu sur 100 tirages.

t	O_g	O_b	O_{opt}	$NbFix$	$NbSol$	$NbMax$
10	1.8	4.1	1.7	18	1.2	3
15	5.6	7.1	5.3	15	1.2	3
20	9.3	10.8	8.6	12	1.2	3
30	11.4	12.8	10.7	11	1.2	3

TABLE 8 : Valeurs moyennes des poids des cocircuits-retour, du nombre moyen d'item fixés et des nombres moyen et maximum de classes optimales.

L'heuristique gloutonne est proche de l'optimum et fait nettement mieux que la procédure de Bordat. On notera qu'à contrario des ordres, il y a peu de bipartitions optimales, le plus souvent une seule, même pour des classements aléatoires. De façon imprévue, l'énumération porte sur moins de la moitié des candidats, parfois aucun car ils sont tous affectés d'office.

V CONCLUSIONS

Dans ce texte, nous avons essentiellement réalisé trois tâches nouvelles :

- calculé de meilleures bornes supérieures de l'écart d'un ordre total à un tournoi grâce à l'optimisation locale et à la décomposition du tournoi ;
- séparé les deux problèmes de classement (ordre total) et de sélection d'un nombre fixé de candidats ;
- proposé une méthode optimale de sélection dans le cas où n est petit, à l'aide de bornes efficaces.

D'une manière générale, nous préconisons la démarche suivante⁶ :

- S'entendre sur les candidats, à l'issue d'un premier choix libre ou d'une discussion entre les membres du comité de programme, et sur le nombre d'invités ;

6. Les programmes sont disponibles sur simple demande à l'auteur.

- Faire voter le comité qui exprime un ensemble d'ordres totaux, sur les candidats, si possible en évitant les ex-aequo ;
- Construire le tournoi pondéré par les préférences majoritaires ;

Pour le problème de construction d'un ordre unique ;

- Enumérer toutes les ordres médians ;
- Choisir le plus central de ces ordres.

Pour le problème de sélection de $k > 3$ candidats ;

- Calculer les bornes supérieure d'appartenance à l'une ou l'autre classe ;
- Enumérer les bipartitions des candidats dont l'affectation n'est pas fixée.

La principale vertu de cette approche est qu'elle tient au mieux compte des classements initiaux, puisqu'elle contredit un nombre minimum de préférences individuelles et qu'on ne peut faire moins. Même si les algorithmes sont de complexité élevée ils s'appliquent à des données de cette taille, courantes dans bien des situations évoquées dans l'introduction. Pour la sélection, si on ne veut qu'un seul ou deux ou trois sélectionnés les solutions optimales sont toujours calculables.

La procédure d'énumération ne demande pas de place mémoire et peut s'envisager pour des problèmes plus importants. Nous obtenons un groupe de 10 sélectionnés parmi 30 candidats, avec 100 votants assez divergents, puisque les ordres ont été générés avec 50 transpositions aléatoires, en moins d'une minute de calcul. Il y a de quoi choisir les meilleurs crus de l'année, selon un pannel d'experts, ou les "entrants" d'une section du CNRS, autres domaines d'application de cette méthode de sélection.

Références

- [1] Barthélemy J.P., Monjardet B. (1981) The Median Procedure in Cluster Analysis and Social Choice Theory, *Mathematical Social Sciences* 1, 235-267.
- [2] Barthélemy J.P., Guénoche A., Hudry O. (1989). Median linear orders : Heuristics and Branch and Bound algorithm. *European Journal of Operational Research* 42, 555-579.
- [3] Bermond J.C. (1972) Ordre à distance minimale d'un tournoi et graphes partiels sans circuits maximaux, *Mathématiques et Sciences humaines* 37, 5-25.
- [4] Borda J.C. (1784) Mémoire sur les élections au scrutin, in *Histoire de l'Académie Royale des Sciences pour 1781 1784*.
- [5] Caritat M.J.A.N., Marquis de Condorcet. (1785) *Essai sur l'application de l'analyse de la probabilité des décisions rendues à la pluralité des voix*.
- [6] Charon I., Germa A., Hudry O. (1992) Utilisation des scores dans les méthodes exactes déterminant les ordres médians de tournois, *Mathématiques et Sciences humaines* 119, 53-74.
- [7] Charon I., Guénoche A., Hudry O., Woïgard F. (1996) A Bonsai Branch and Bound method applied to voting theory, *Proceedings of "Ordinal and Symbolic Data Analysis"*, OSDA'95, E. Diday et al. (Eds.), Springer Verlag, 309-318.
- [8] Charon I., Hudry O., Woïgard F. (1996) Ordres médians et ordres de Slater des tournois, *Mathématiques et Sciences Humaines* 133, 23-56, 1996.
- [9] Charon I., Hudry O. (2001) The noising methods : a generalization of some metaheuristics, *European Journal of Operational Research* 135(1), 86-101.
- [10] Finke G., Burkard R.E., Rendl F. (1987) Quadratic assignment problems, *Annals of Discrete Mathematics*, 31, 61-82.
- [11] Guénoche A. (1977) Un algorithme pour pallier l'effet Condorcet, *R.A.I.R.O. Recherche Opérationnelle* 11(1), 77-83.

- [12] Hansen P., Mladenović N. (2001) Variable neighborhood search : Principles and Applications, *European Journal of Operational Research* 130(3), 449-467.
- [13] Hudry O. (1989) *Recherche d'ordres médians : complexité, algorithmique et problèmes combinatoires*, Thèse de l'ENST, Paris.
- [14] Hudry O. (2012) On the computation of median linear orders, of median complete preorders and of median weak orders, *Mathematical Social Sciences* 64, 2-10.
- [15] Kemeny J.G. (1959) Mathematics without numbers, *Daedalus* 88, 577-591.
- [16] Monjardet B. (1973) Tournoi et ordre médian pour une opinion, *Mathématiques et Sciences humaines* 43, 55-73.
- [17] Nijenhuis A., Wilf H.S. (1975) *Combinatorial algorithms*, Academic Press.
- [18] Roupin F. (2004) From linear to semidefinite programming : an algorithm to obtain semidefinite relaxations for bivalent quadratic problems, *Journal of Combinatorial Optimization* 8(4), 469-493.
- [19] Slater P. (1961) Inconsistencies in a schedule of paired comparisons, *Biometrika* 48, 303-312.
- [20] Remage R., Thomson W.A. (1966) Maximum likelihood paired comparison rankings, *biometrika*, 53, 143-149.
- [21] Smith A.F.M., Payne C.D. (1974) An algorithm for determining Slater's λ and all nearest adjoining orders, *British Journal of Mathematical and Statistical Psychology* 27, 49-52.
- [22] Woïrgard F. (1997) *Recherche et dénombrement des ordres médians des tournois*, Thèse de l'ENST.