MATHÉMATIQUES ET SCIENCES HUMAINES

IRÈNE CHARON-FOURNIER
ANNE GERMA
OLIVIER HUDRY

Utilisation des scores dans des méthodes exactes déterminant les ordres médians de tournois

Mathématiques et sciences humaines, tome 119 (1992), p. 53-74
<http://www.numdam.org/item?id=MSH_1992__119__53_0>

Math. Inf. Sci. hum. (30e année, n° 119, *1992*, p. 53-74)

UTILISATION DES SCORES DANS DES MÉTHODES EXACTES DÉTERMINANT LES ORDRES MÉDIANS DE TOURNOIS

Irène CHARON-FOURNIER, Anne GERMA et Olivier HUDRY [1]

RÉSUMÉ — *Dans cet article, nous utilisons un paramètre σ(T) défini à partir des scores d'un tournoi T pour déterminer les ordres médians de T. Ce paramètre évalue un éloignement entre le tournoi T et les tournois transitifs ayant le même nombre de sommets. Appelant i(T) le nombre minimum d'arcs à inverser pour rendre T transitif, et n le nombre de sommets de T, nous proposons d'abord deux algorithmes linéaires en n calculant i(T) et un ordre médian de T pour les tournois T tels que σ(T) soit égal à 1 ou 2. Puis nous donnons des résultats expérimentaux sur l'utilisation de σ(T) dans une méthode arborescente par séparation et évaluation progressive ("Branch and Bound") déterminant, pour tout tournoi T, i(T) et les ordres médians de T.*

ABSTRACT — On the use of scores in exact methods computing the median orders of tournaments.
In this paper, we use a parameter σ(T) defined from the scores of a tournament T to determine the median orders of T. This parameter measures a remoteness between the tournament T and the transitive tournaments with the same number of vertices. Calling i(T) the minimum number of arcs that it is necessary to reverse to make T transitive, and n the number of vertices of T, we give first two linear (in n) algorithms to compute i(T) and a median order of T for T such that σ(T) is equal to 1 or 2. Then we give experimental results on the use of σ(T) in a Branch and Bound method designed to compute i(T) and the median orders of T.

1. INTRODUCTION

Un tournoi T_n d'ordre n est un graphe orienté complet antisymétrique à n sommets : entre toute paire de sommets distincts i et j, il existe un et un seul des deux arcs (i, j) ou (j, i). Le score s_i d'un sommet i est le demi-degré extérieur de i, c'est-à-dire le nombre d'arcs ayant i pour origine. Le vecteur-score d'un tournoi T_n est le vecteur ordonné $(s_1, s_2, \ldots, s_n)$, les sommets étant numérotés de façon à avoir $s_1 \leq s_2 \leq \ldots \leq s_n$. H.G. Landau [6] a donné une condition nécessaire et suffisante pour qu'un vecteur $(s_1, s_2, \ldots, s_n)$ soit le vecteur-score d'un tournoi T_n (pour les définitions et les résultats de base, voir J.W. Moon [7]).

THÉORÈME 1. (H.G. Landau) : $(s_1, s_2, \ldots, s_n)$ avec $s_1 \leq s_2 \leq \ldots \leq s_n$ est le vecteur-score d'un tournoi si et seulement si $\sum_{k=1}^{i} s_k \geq \frac{i(i-1)}{2}$ pour $1 \leq i \leq n-1$ et $\sum_{k=1}^{n} s_k = \frac{n(n-1)}{2}$. ♦

En particulier, le vecteur-score du tournoi transitif est le vecteur (0, 1, ..., n-1).

Dans [3], nous avons étudié certaines propriétés du paramètre σ(T) défini pour tout tournoi T de la façon suivante.

DEFINITION 1. Soit $s = (s_1, s_2, \ldots, s_n)$ le vecteur-score d'un tournoi T ; on pose alors

[1] Ecole Nationale Supérieure des Télécommunications, 46 rue Barrault, 75634 Paris Cedex 13 France.

$$\sigma(T) = \frac{1}{2}\sum_{i=1}^{n} |s_i - i + 1|$$

Ce paramètre évalue un éloignement entre le tournoi T et les tournois transitifs ayant le même nombre de sommets. En particulier, $\sigma(T)$ vaut 0 si et seulement si T est transitif. Le paramètre $\sigma(T)$ fournit des indications sur le problème de P. Slater ([9]). Celui-ci consiste à déterminer, dans un tournoi T donné, un ensemble minimum d'arcs dont l'inversion rend T transitif (cet ensemble minimum n'est d'ailleurs pas nécessairement unique).

DÉFINITION 2. Soit T un tournoi. On appelle "indice de Slater" de T et on note i(T) le nombre minimum d'arcs à inverser dans T pour rendre T transitif.

L'indice de Slater i(T) correspond à la mesure d'une distance entre le tournoi T et l'ensemble des tournois transitifs sur les sommets de T ([2]). Le tournoi transitif obtenu en inversant ces i(T) arcs est caractéristique d'un ordre total, appelé "ordre médian" de T ([2]). L'ordre médian de T est donc un ordre à distance minimum de T. Nous parlerons plus loin de "distance" entre un ordre total (quelconque) O et un tournoi T : il s'agira du nombre d'arcs à inverser dans T pour obtenir O. Pour plus de commodité, un tournoi transitif sera confondu avec l'ordre total qui lui est sous-jacent.

Nous allons d'abord nous intéresser, dans les deux premières parties, à la complexité du problème de Slater pour les tournois T vérifiant $\sigma(T) \in \{1, 2\}$. La dernière partie est consacrée à l'utilisation du paramètre $\sigma(T)$ dans une méthode arborescente pour la résolution du problème de Slater ; nous montrons, à partir de résultats expérimentaux, que sa prise en compte dans la fonction d'évaluation de cette méthode permet d'accélérer considérablement la recherche d'un ordre médian d'un tournoi, et d'envisager de déterminer la valeur exacte de i(T) pour des tournois possédant davantage de sommets.

2. COMPLEXITÉ DU PROBLEME DE SLATER POUR $\sigma = 1$

Nous avons rappelé plus haut en quoi consiste le problème de P. Slater ([9]). D.H. Younger ([12]) a établi une formulation équivalente de ce problème : déterminer un ensemble minimum d'arcs d'un tournoi T dont la destruction dans T donne un graphe sans circuit ("Feedback arc set problem"). Les efforts jusqu'à présent entrepris n'ont pas permis d'établir si ce problème admet un algorithme de résolution polynomial, ou s'il est NP-difficile... Notons cependant que plusieurs problèmes proches de celui-ci sont NP-difficiles. Ainsi le problème du "Feedback arc set" est NP-difficile si on considère l'ensemble des graphes antisymétriques (non nécessairement complets) [5]. De même, dans un tournoi valué, la détermination d'un ensemble d'arcs de poids minimum dont l'inversion transforme le tournoi en un tournoi transitif constitue un problème NP-difficile (voir par exemple [11]).

Dans cette partie, nous donnons un algorithme linéaire permettant de déterminer l'indice de Slater ainsi qu'un ordre médian de tout tournoi T tel que $\sigma(T)$ égale 1. Pour cela, nous supposerons que les sommets de T sont numérotés selon l'ordre croissant des scores. Nous aurons besoin de plus du résultat suivant :

Lemme 2. Soit T un tournoi dont les sommets sont numérotés selon l'ordre croissant des scores. Le nombre d'arcs ayant leur origine dans $\{x_1, x_2, ..., x_p\}$ et leur extrémité dans $\{x_{p+1}, x_{p+2}, ..., x_n\}$ est inférieur ou égal à $\sigma(T)$ pour tout p.

Preuve . Voir [3]. ♦

THÉORÈME 3. La détermination de l'indice de Slater et d'un ordre médian d'un tournoi T à n sommets et de paramètre $\sigma(T) = 1$ est linéaire en n.

Preuve . On suppose les sommets $\{x_1, \ldots, x_n\}$ rangés par ordre croissant des scores. La valeur de σ étant 1, la suite de scores est du type (0, 1, ..., j-3, j-2, j, j, j+1,..., k-3, k-2, k-2, k, k+1,..., n-3, n-2, n-1). On suppose les sommets classés de telle sorte que l'arc entre x_j et x_{j+1} (tous deux de score j) aille de x_{j+1} vers x_j. L'allure de T est illustrée par la figure 1, dans laquelle tous les arcs manquants sont orientés de la droite vers la gauche.

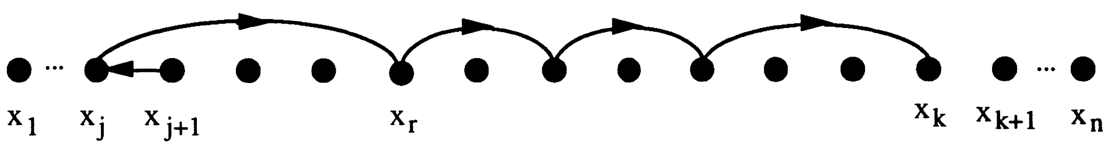

Figure 1

Pour rendre T transitif, il est nécessaire et suffisant de rendre transitif le sous-tournoi T' engendré par $\{x_{j+2}, x_{j+3}, \ldots, x_n\}$, lui-même de paramètre $\sigma(T') \leq 1$, et de détruire les circuits qui peuvent encore subsister après cette opération. Or, si on appelle x_r le fils de x_j d'indice r supérieur à j (d'après le lemme 2, si x_r existe, il est unique) dans T, les circuits qui subsistent, par exemple le circuit (x_j,x_r,x_{j+1}), passent tous nécessairement par l'arc (x_j,x_r), et on peut donc tous les détruire en inversant l'arc (x_j,x_r), de sorte que i(T) = i(T')+1. On propose alors pour déterminer l'indice de Slater et un ordre médian de T l'algorithme décrit ci-dessous, qui utilise la structure de données suivante :

- les sommets de T sont numérotés par 1, 2, ..., n ;
- A est un tableau de longueur n contenant la description de T : A[j] indique le sommet d'indice r dominé par le sommet d'indice j avec r > j, si un tel sommet existe ;
- X est un tableau contenant les sommets rangés par scores croissants ;
- S est un tableau de longueur n tel que S[j] = score de X[j] ;
- Y est un tableau de longueur n qui contient, à la fin de l'exécution de l'algorithme, les sommets de T rangés selon un ordre médian.

ALGORITHME 1
- poser i(T):=0 ;
- pour j variant de 1 à n, faire :
  ** si S[j]=j-1, alors Y[j]:=X[j]* *(* le score de Y[j] est et restera j-1 *)*
  ** sinon* *(* X[j] et X[j+1] ont même score : S[j] = S[j+1] = j *)*
    · i(T):=i(T)+1 ;
    · si A[j] = j+1, alors échanger X[j] avec X[j+1] et A[j] avec A[j+1] ;
      (quand deux sommets X[j] et X[j+1] sont de même score, on *)*
      (veut que ce soit X[j+1] qui domine X[j], et non le contraire *)*
    · r:=A[j] ; *(* r est l'indice du sommet domminé par X[j] avec r > j *)*
    · S[r]:=S[r]+1 ; *(* on inverse l'arc de X[j] à X[r] *)*
    · S[j]:=S[j]-1 ;
    · Y[j]:=X[j] ;
    · si S[r]=S[r+1]+1, alors *(* on conserve l'ordre croissant des scores dans X *)*
      - échanger X[r] et X[r+1] ;
      - échanger S[r] et S[r+1] ;
      - échanger A[r] et A[r+1]
- fin du pour.

Le lemme 2 permet de conclure que la structure de T est entièrement contenue dans A : tous les arcs ayant le sommet j pour origine sont orientés vers des sommets d'indice plus petit, sauf au plus un arc pour chaque j ; c'est précisément cet arc qu'indique A[j]. On déduit immédiatement que la complexité de l'algorithme est linéaire en n. ♦

Remarques
1. L'hypothèse, simplificatrice, selon laquelle les sommets sont déjà rangés par scores croissants n'est pas indispensable. Au moment d'acquérir les données, c'est-à-dire ce que contient A, on peut aisément tenir à jour les tableaux X et S pour qu'il en soit ainsi, après avoir initialisé X et S de la façon suivante : X[j] := j et S[j] := j-1. $\sigma(T)$ valant 1, la mise à jour due à l'acquisition de chaque arc (j, A[j]) peut alors être effectuée à l'aide d'un nombre constant d'opérations, et l'algorithme reste ainsi linéaire.

2. Il est facile d'autre part d'adapter cet algorithme pour la détermination de tous les ordres médians d'un tournoi T de paramètre $\sigma(T) = 1$. En effet si, lors d'une itération, l'indice r est strictement supérieur à j+2, seule l'inversion de (X[j], X[r]) permet de détruire tous les circuits passant par X[j] ; au contraire si r est égal à j+2, chacun des trois arcs (X[j], X[j+2]), (X[j+1], X[j]) et (X[j+2], X[j+1]) peut être inversé pour constituer un ordre médian. Pour obtenir tous les ordres médians, il suffit donc d'envisager successivement chacune de ces trois inversions (mais dans ce cas, l'algorithme n'est bien sûr plus nécessairement polynomial, puisqu'il peut alors y avoir un nombre exponentiel d'ordres médians).

3. COMPLEXITÉ DU PROBLÈME DE SLATER POUR $\sigma = 2$

Nous nous intéressons maintenant au cas des tournois T tels que $\sigma(T) = 2$. Nous allons montrer que la détermination de l'indice de Slater de tout tournoi T tel que $\sigma(T) = 2$ reste linéaire en n (théorème 11). Auparavant, nous rappelons un résultat dû à R. Remage et W.A. Thompson (lemme 4), puis nous établissons six lemmes (lemmes 5 à 10) dont découlera le théorème 11. Pour cela, nous supposerons encore dans toute cette partie que les sommets de T sont numérotés selon l'ordre croissant des scores.

Notations. Nous utiliserons les notations suivantes : on note [x, T] la distance minimum de T à l'ensemble des tournois transitifs dont x est le sommet de score 0 (on dira aussi un tournoi finissant par x), et [{x,y}, T] la distance minimum de T à l'ensemble des tournois transitifs dont {x, y} sont globalement les sommets de scores 0 et 1 (on dira aussi un tournoi finissant par x et y, ou finissant par {x, y}).

Lemme 4 . (R. Remage et W.A. Thompson, 1966) : Soit T un tournoi et OM un ordre médian de T. Si x et y sont deux sommets consécutifs dans OM, l'orientation de l'arc entre x et y est la même dans OM que dans T.

Preuve. Voir [8]. Ce lemme exprime le fait que les arcs entre deux sommets consécutifs d'un ordre médian quelconque constituent un chemin hamiltonien de T. ♦

Lemme 5 . Etant donné un tournoi T d'indice de Slater i(T) alors
- étant donné un sommet a de score 1 de T, il existe un tournoi transitif à distance i(T) ou i(T)+1 de T finissant par a : $i(T) \le [a, T] \le i(T)+1$;
- étant donné un sommet b de score 2 de T, il existe un tournoi transitif à distance i(T), i(T)+1 ou i(T)+2 de T finissant par b : $i(T) \le [b, T] \le i(T)+2$;
- étant donnés deux sommets a et b de scores respectifs 1 et 2, il existe un tournoi transitif à distance au plus i(T)+2 de T finissant par {a,b} : $i(T) \le [\{a,b\}, T] \le i(T)+2$.

Preuve. Soit T' = T - {a}. L'indice de Slater de T' est au plus i(T) : tout ordre médian de T duquel on ôte le sommet a donne un ordre total à distance au plus i(T) de T'. Pour rendre T' transitif, il suffit donc d'inverser au plus i(T) arcs. On inverse de plus dans T l'unique arc issu de a. Le tournoi obtenu est transitif, finit par a, et est à distance au plus i(T)+1 de T. D'autre part, par définition de i(T), il faut bien inverser au moins i(T) arcs de T pour obtenir un ordre total finissant par a. D'où le résultat pour la première proposition du lemme 5.

La preuve est exactement la même pour b (cette fois on inverse, d'une part les arcs nécessaires dans T-{b}, d'autre part les deux arcs d'origine b). Elle est en fait encore semblable pour {a,b} : en effet, les deux sommets a et b étant liés par un arc (puisque ce sont deux sommets d'un tournoi), il "sort" deux arcs de {a,b}, et il suffit d'inverser ces deux arcs ainsi que les i(T-{a,b}) ≤ i(T) arcs dont l'inversion rend transitif T-{a,b} pour obtenir un tournoi transitif finissant par {a,b}. Remarquons que, dans ce dernier cas, on n'inverse pas l'arc entre a et b. ♦

Lemme 6 . Soit T un tournoi, a et b deux sommets de scores respectifs 1 et 2 de T. Si on a [{a,b},T] = i(T)+2 alors, si on pose T' = T-{a,b}, i(T) = i(T'). De plus : l'arc entre a et b va nécessairement de b vers a, il existe un sommet x de T dominé par a et b, et tout ordre médian de T finit par ... > b > a > x (c'est-à-dire que x est dominé par tous les autres sommets, a par tous les sommets sauf x, et b par tous les sommets sauf x et a).

Preuve. On a clairement i(T') ≤ i(T) ≤ i(T')+2. Or, d'après l'hypothèse du lemme 6, pour mettre {a,b} à la fin d'un ordre à distance minimale de T, il faut i(T)+2 inversions d'arcs et, d'après la preuve du lemme 5, il suffit de i(T')+2 inversions. Ceci prouve que i(T) = i(T').

Soit OM(T) un ordre médian de T. Sa trace sur T' est bien sûr un ordre de T' et, puisque i(T)=i(T'), c'est un ordre médian de T'. Les arcs à inverser pour obtenir OM(T) ne sont donc incidents ni à a ni à b qui de ce fait conservent dans OM(T) les scores qu'ils avaient dans T. Supposons que a domine b, et soit y un fils de b dans T ; après avoir rendu T' transitif, il reste le circuit (a,b,y), et on ne peut pas avoir i(T) = i(T') ; donc b domine a. Soit x le fils de a dans T et y le fils de b autre que a. Si on avait x ≠ y, après inversion des i(T) arcs requis pour obtenir OM(T), il subsisterait le circuit (b,a,x). On a donc x = y, et T possède les arcs (b, a), (b, x), (a, x) et aucun autre arc dont a ou b serait l'origine. Puisque les arcs adjacents à a et b ne sont pas inversés, seul x est, dans OM(T), dominé par a, seuls x et a sont dominés par b, et ceci achève la preuve du lemme. ♦

Lemme 7. Supposons qu'un sommet a de score minimum dans T ait pour score 1, et appelons x l'unique sommet dominé par a. Soit T' = T-{a}. L'égalité i(T) = i(T') est vraie si et seulement s'il existe un ordre médian de T' finissant par x.

Preuve . D'après le lemme 5, on a déjà i(T') ≤ i(T) ≤ i(T')+1. S'il existe un ordre médian de T' finissant par x, on en déduit, sans inverser d'arc supplémentaire, un ordre médian de T finissant par ... > a > x, et donc i(T) = i(T'). Réciproquement, si i(T) = i(T'), la trace sur T' d'un ordre médian de T est un ordre de T', et, puisqu'il utilise dans ce cas au plus i(T') inversions, c'est un ordre médian de T'. Ceci implique que, pour obtenir un ordre médian de T, on n'inverse pas d'arc incident à a : seul x est et reste dominé par a. L'ordre médian finit donc par ... > a > x, et sa trace sur T' est un ordre médian de T' finissant par x. ♦

Lemme 8. Soit T un tournoi vérifiant $\sigma(T) = 2$. Le vecteur-score de T commence alors par 01, 02, 1113, 11224, 1123, 1124, 113, 1222 ou 1223, et contient au plus un score nul, au plus trois scores égaux à 1 et au plus trois scores égaux à 2.

Preuve. Laissée au lecteur. ♦

Lemme 9. Soit T un tournoi ne possédant aucun sommet de score nul mais deux sommets x et y de score 1 tels que x domine y. Alors il existe un ordre médian de T finissant par x > y.

Preuve. Soit OM un ordre médian de T finissant par y. D'après le lemme 4, x n'étant pas à la fin de OM, x doit précéder dans OM un sommet z tel que (x,z) soit un arc de T. Ceci n'est possible que si z = y : tout ordre médian finissant par y finit par x > y.

Supposons maintenant que OM finisse par z $\neq$ y. Appelons t le sommet dominé par y dans T. D'après le lemme 4, y précède t immédiatement dans OM. Celui-ci est donc de la forme OM = ... > y > t > (...) > z, avec k sommets entre t et z (k $\geq$ 0). Alors il est facile de voir que l'ordre O obtenu à partir de OM en rejetant y à la dernière position et en ne changeant pas le reste permet d'économiser k $\geq$ 0 inversions par rapport à OM. Cet ordre est donc médian lui-même, et finissant par z > y ; ceci n'est possible que si z = x d'après ce qui précède, et O convient. ♦

Lemme 10 . Il existe un algorithme, linéaire en n, qui détermine l'indice de Slater d'un tournoi d'ordre n et de paramètre $\sigma \leq 2$, et qui calcule les quantités [x,T], [y,T] et [{x,y},T] pour tout sommet x de score 1 (s'il en existe) et tout sommet y de score 2 (s'il en existe),

Preuve. Nous allons montrer que l'algorithme 2 donné plus bas convient. Cet algorithme est récursif. Sa partie récursive ne s'applique qu'à n assez grand (par exemple n > 5) ; pour n petit (par exemple n $\leq$ 5), on peut supposer qu'on applique une autre procédure, non décrite ici, qui calcule la distance entre le tournoi T auquel on applique l'algorithme et tout ordre total défini sur les sommets de T.

L'algorithme consiste, à partir d'un tournoi T, à supprimer de T un certain sommet de score 0, 1 ou 2 pour obtenir un tournoi T' tel que σ(T') soit inférieur ou égal à 2 ; puis, en appliquant l'algorithme à T', on détermine la valeur de i(T') et des quantités [x', T'], [y', T'] et [{x',y'}, T'] pour tout sommet x' de score 1 dans T' (s'il en existe) et tout sommet y' de score 2 dans T' (s'il en existe) ; la connaissance de ces valeurs permet alors de calculer les quantités homologues dans T. La description et la preuve de l'algorithme se font cas par cas, suivant les valeurs des plus petits scores de T.

L'algorithme 2 utilise les notations suivantes :

- X[j] désigne le $j^{ième}$ sommet de T selon l'ordre des scores croissants ;
- S[j] représente le score de X[j].

ALGORITHME 2

Si n ≤5, appliquer une procédure particulière qui calcule la distance entre T et tout ordre total ; sinon, faire :

- *si S[1] = 0, alors faire* (* *le plus petit score vaut 0* *)
    * *a := X[1];* (* *a désigne le sommet — unique — de score 0* *)
    * *T' := T — {a};*
    * *appliquer l'algorithme à T';*
    * *i(T) := i(T');*
    * *si S[2] = 1, alors faire* (* *T ne peut posséder qu'un seul sommet de score 1* *)
        · *b := X[2];* (* *score de b = 1 dans T et 0 dans T'* *)
        · *[b, T] := i(T)+1;*
        · *pour tous les sommets c de score 2 (s'il y en a), faire*
            [c, T] := i(T)+2;
            [{b,c}, T] := i(T)+2;
        · *fin du pour;*
    * *sinon* (* *S[2] = 2 nécessairement* *)

```
            · pour tous les sommets c de score 2, faire                              (* il y en a au moins un *)
                  [c, T] := [c, T'] + 1;
            · fin du pour;
      * fin du si;
- fin du si;

- si S[1] = 1 et S[2] = 1, alors faire                          (* les deux plus petits scores valent 1 *)
      * si S[3] = 1, alors faire                                (* les trois plus petits scores valent 1, *)
                                                                 (* et il n'y a pas de score égal à 2 *)
            · a := X[1];                           (* a est de score 1 et domine un sommet de score 1 *)
            · T' := T — {a};
            · appliquer l'algorithme à T';
            · i(T) := i(T') + 1;
            · pour les trois sommets x de score 1, faire [x, T] := i(T);       (* y compris [a, T] *)
            · fin du pour;
      * fin du si;
      * si S[3] = 2, alors faire                              (* le vecteur-score commence par 112... *)
            · a := celui des deux sommets X[1] ou X[2] qui domine l'autre;        (* a de score 1 *)
            · b := l'autre de ces deux sommets                                    (* b de score 1 *)
            · c := X[3];                                               (* c de score 2 ; c domine a *)
            · T' := T — {a};                                           (* b et c de score 1 dans T' *)
            · appliquer l'algorithme à T';
            · i(T) := [b, T'];
            · [a, T] := i(T')+1;
            · [b, T] := i(T);
            · pour tous les sommets c de score 2, faire                             (* il y en a 1 ou 2 *)
                  si c domine b, alors faire [c, T] := i(T')+2;
                  sinon [c, T] := i(T')+1;
                  fin du si;
                  [{a, c}, T] := [c, T']+1;
                  [{b, c}, T] := i(T')+1;
            · fin du pour;
      * fin du si;
      * si S[3] = 3, alors faire               (* le vecteur-score commence nécessairement par 1133... *)
            · a := celui des deux sommets X[1] ou X[2] qui domine l'autre;        (* a de score 1 *)
            · b := l'autre de ces deux sommets;                                   (* b de score 1 *)
            · T' := T — {a};                                           (* b et c de score 1 dans T' *)
            · appliquer l'algorithme à T';
            · i(T) := [b, T'];
            · [a, T] := i(T')+1;
            · [b, T] := i(T);
      * fin du si;
- fin du si;                                                         (* le cas s[3] > 3 est impossible *)

- si S[1] = 1 et S[2] = 2, alors faire       (* le vecteur-score commence nécessairement par 122... *)
      * a := X[1];                                                                (* a de score 1 *)
      * si a domine un sommet de score 2 qui lui-même domine un autre sommet de score 2,
alors faire
            · b := le sommet dominé par a;                                       (* b de score 2 dans T *)
            · c := un sommet de score 2 dominé par b;             (* il peut y avoir deux tels sommets *)
            · T' := T — {a};                         (* b de score 2 dans T' et c de score 1 dans T' *)
            · appliquer l'algorithme à T';
```

· *appliquer l'algorithme à T';*
· *si [b, T'] = i(T'), alors faire i(T) := i(T');*
· *sinon faire i(T) := i(T') + 1;*
· *fin du si;*
· *[a, T] := i(T') + 1;*
· *[b, T] := [b, T'];*
· *[c, T] := [{b,c}, T'] + 1;*
· *[{a,b}, T] := [b, T'];*
· *[{a,c}, T] := [c, T'] + 1;*
· *si S[4] = 2, alors faire* (* le vecteur-score commence par 1222... *)
d := X[4]; (* d de score 2 dans T et 1 dans T' *)
si b domine d, alors faire [d, T] := [{b,d}, T'] + 1;
sinon faire (* b est dominé par d *)
- *si [b, T'] = i(T') alors faire [d, T] := i(T) + 2;*
- *sinon faire [d, T] := i(T) + 1;*
- *fin du si;*

fin du si;
[{a,d}, T] := [d, T'] + 1;
· *fin du si;*

* *fin du si;*
* *si le sommet dominé par a est de score strictement supérieur à 2, alors faire*
· *si S[4] = 2, alors faire* (* le vecteur-score commence par 1222... *)
T' := T — {a};
appliquer l'algorithme à T';
i(T) := i(T') + 1;
[a, T] := i(T);
pour les trois sommets x de score 2, faire
- *[x, T] := i(T) + 1;*
- *[{a, x}, T] := i(T);*

fin du pour;
· *sinon faire* (* le vecteur-score commence par 1223... *)
b := celui des sommets de score 2 qui domine l'autre;
c := l'autre de ces deux sommets;
T' := T — {b}; (* dans T', a est de score 1 et c de score 2 *)
appliquer l'algorithme à T';
si i(T') = [{a,c}, T'], alors faire i(T) := i(T');
sinon faire i(T) := i(T') + 1;
fin du si;
si i(T') = [{a,c}, T'] ou i(T') = [{a,c}, T'] + 1, alors faire [a, T] := i(T);
sinon faire [a, T] := i(T) + 1; (* i(T') = [{a,c}, T'] + 2 *)
fin du si;
[b, T] := i(T') + 2;
[{a,b}, T] = [a, T'] + 1;
[{a,c}, T] = [{a,c}, T'];
si [{a,c}, T'] = i(T') ou [{a,c}, T'] = i(T') + 1, alors faire [c, T] = i(T) +1;
sinon, faire [c, T] = i(T) + 2;
fin du si;
· *fin du si;*

* *fin du si;*
* *si a domine un sommet de score 2 qui ne domine aucun autre sommet de score 2, alors faire* (* ceci entraîne que le vecteur-score commence par 1223... *)
· *c := le sommet dominé par a;*
· *b := l'autre sommet de score 2;* (* b domine c *)

· *T' := T — {b};*
· *appliquer l'algorithme à T';*
· *si [{a,c}, T'] = i(T'), alors faire*
i(T) := i(T');
[a, T] := i(T) + 1;
· *sinon faire*
i(T) := i(T') + 1;
[a, T] := [a, T'] + 1;
· *fin du si;*
· *[b, T] := i(T') + 2;*
· *[c, T] := i(T);*
· *[{a, b}, T] :=[a, T'] + 1;*
· *[{a, c}, T] := i(T);*
* *fin du si;*
\- *fin du si.*

Le lemme 8 montre que tous les cas sont envisagés par l'algorithme 2.

Complexité de l'algorithme

On suppose que le tournoi T est codé, comme pour l'algorithme 1, uniquement à l'aide des arcs orientés d'un sommet de faible score vers un sommet de score plus élevé : à chaque sommet x, on associe l'ensemble E(x) des (au plus deux) sommets dominés par x et de score plus élevé, et on place ces ensembles dans un tableau A selon l'ordre croissant des scores. Le lemme 2 montre que le codage de T est linéaire en n.

De plus le passage d'un tournoi T à un tournoi T' = T — {x} où x est un sommet de T de score 0, 1 ou 2 peut se faire en temps constant. En effet, il est nécessaire et suffisant d'ôter de A l'ensemble E(x), de retirer x des ensembles E(y) où il apparaissait, et de faire en sorte que le tableau A' résultant soit toujours ordonné selon les scores croissants. La première opération consiste en fait à décaler "à droite" les cases de A associées à des sommets de score inférieur ou égal à celui de x pour écraser la case associée à x, puis à ignorer définitivement la première case de A ; le score de x étant 0, 1 ou 2, un tel décalage se fait en un temps constant. La deuxième opération peut aussi être trivialement effectuée en un temps constant : seuls les sommets de score inférieur ou égal à celui de x sont concernés... Quant à la troisième (tri de A'), elle n'est finalement pas plus complexe. La suppression de x provoque une diminution d'une unité des scores de tous les sommets, sauf de ceux contenus dans E(x) : il y en a au plus deux. Ceci revient, à une constante additive près (égale, à une itération quelconque de l'algorithme, au nombre de sommets retirés depuis le début de celui-ci), à augmenter d'une unité les scores des éléments de E(x). La forme particulière des vecteurs-score compatibles avec $\sigma \leq 2$ fait que les éléments de E(x) ne peuvent se déplacer "à droite" que d'un nombre borné de positions. Par conséquent la mise à jour du tableau A' peut se faire en un temps constant, et permet en outre de connaître immédiatement les scores des sommets restants (à une constante additive près).

D'autre part le lemme 8 montre que toutes les boucles du type "pour" sont exécutées un nombre borné (par 3) de fois.

Enfin, l'algorithme appliqué à un tournoi à n sommets ne nécessite sa propre application à un tournoi à n-1 sommets qu'une seule fois. Il est clair alors que l'algorithme 2 est linéaire en n : si on appelle f(n) le nombre d'opérations nécessaires pour déterminer i(T) pour un tournoi T à n sommets (et de paramètre $\sigma(T) \leq 2$), on a alors $f(n) \leq f(n-1) + K$, où K est une constante ; d'où la complexité annoncée.

Preuve de l'algorithme

Montrons maintenant que l'algorithme 2 calcule bien ce qu'il faut. Cinq cas se présentent, le dernier étant lui-même subdivisé en quatre sous-cas.

1) Le plus petit score de T est 0
Soit a l'unique sommet de score 0. Posons T' = T - {a} ; on a évidemment i(T) = i(T'). Il reste à déterminer, pour tout sommet x de score 1 et tout sommet y de score 2 les quantités [x, T], [y, T] et [{x,y}, T]. Nous détaillons ici un des deux cas, l'étude (triviale) de l'autre étant laissée au lecteur.

Nous supposons ici qu'aucun sommet n'est de score 1. D'après le lemme 8, il y a alors au moins un sommet de score 2. Soit c un tel sommet ; le score de c dans T' vaut 1. En appliquant l'algorithme à T', on connaît [c, T'], c'est-à-dire le nombre minimum d'arcs à inverser dans T' pour transformer T' en un tournoi transitif finissant par c. Pour transformer T en un tournoi transitif finissant par c, il faut inverser, d'une part l'arc (c,a), d'autre part assez d'arcs pour que, en oubliant le sommet a, on obtienne un tournoi transitif finissant par c, et ces arcs sont, bien sûr, distincts de (c,a) : d'où [c, T] $\geq$ 1 + [c, T']. Réciproquement, en inversant [c, T'] arcs, on sait construire un tournoi transitif sur les sommets de T' finissant par c ; en inversant de plus l'arc (c,a), on transforme T en un tournoi transitif finissant par c à l'aide de [c, T'] + 1 inversions. D'où finalement [c,T] = [c,T']+1.

2) La suite des scores commence par 1 1 1
D'après le lemme 8, il n'y a pas dans ce cas de score égal à 2. Les trois sommets de score 1 jouent des rôles symétriques et forment un circuit qui est une composante fortement connexe du tournoi (voir figure 2, dans laquelle les traits gras résument tous les arcs manquants).

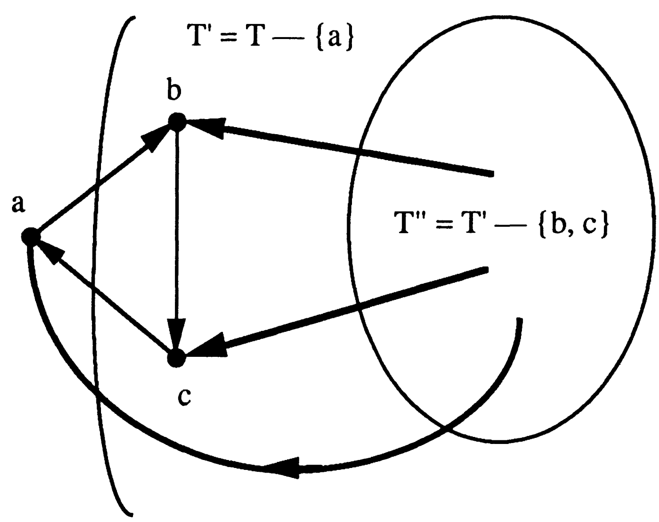

Figure 2

Soit a un sommet de score 1. Posons T' = T — {a} et T'' = T — {a, b, c}. Il est clair que l'on a i(T') = i(T'') et i(T) = i(T'') + 1 = i(T') + 1. De plus il est trivial de constater qu'il existe des ordres médians de T finissant par a, b ou c, ce qui montre la relation [x, T] = i(T) pour x $\in$ {a, b, c}.

3) La suite des scores commence par 112
Soit a le sommet de score 1 qui domine l'autre, et b celui-ci ; appelons aussi c un sommet de score 2 (d'après le lemme 8, il y a 1 ou 2 sommets de score 2) et x le sommet dominé par b (x étant éventuellement un sommet de score 2). Posons T' = T — {a} ; b et c sont de score 1 dans T' (voir figure 3).

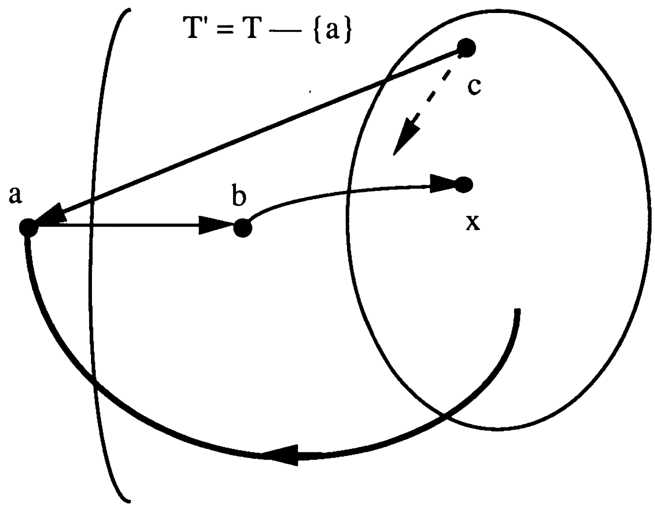

Figure 3

α - calcul de i(T) et de [b, T]

Si [b, T'] = i(T'), il existe un ordre médian de T' finissant par b, et alors, d'après le lemme 7, i(T) = i(T') = [b, T'] et [b, T] = i(T).

Sinon on a, d'après le lemme 5, [b, T'] = i(T') + 1. Supposons que l'on ait i(T) = i(T'), et soit O un ordre médian de T. i(T) = i(T') implique que l'on n'a inversé aucun arc incident à a. Par conséquent, dans O, a domine b et est dominé par x : il a donc fallu inverser l'arc (b,x), et, d'après le lemme 4, b ne peut qu'être à la fin de O. Alors l'ordre obtenu à partir de O en supprimant a est un ordre total finissant par b et à distance i(T') de T', contradiction avec [b, T'] = i(T') + 1. D'où i(T) ≥ i(T') + 1. Or on peut transformer T' en un ordre total finissant par b à l'aide de i(T') + 1 inversions ; il suffit alors d'insérer a juste avant b dans cet ordre total, ce qui donne un ordre total sur T à une distance égale à i(T') + 1 = [b, T'], et cet ordre finit par b.

Dans les deux cas nous avons donc les égalités i(T) = [b, T'] = [b, T].

β - calcul de [a, T]

D'autre part, en inversant l'arc (a,b) et les i(T') arcs qui rendent T' transitif, on obtient un ordre total sur T finissant par a ; comme il est clairement impossible d'obtenir un tel ordre avec seulement i(T') inversions (puisqu'alors on n'inverserait aucun arc incident à a), on a [a, T] = i(T') + 1.

γ - calcul de [c, T]

Examinons maintenant la valeur de [c, T] pour tout sommet c de score 2 dans T (le score de c vaut 1 dans T'). Comme il faut au minimum rendre T' transitif et inverser l'arc (c,a) pour obtenir un ordre total sur T finissant par c, on a [c, T] ≥ i(T') + 1.

Si on suppose que b domine c, d'après le lemme 9, T' admet un ordre médian OM(T') finissant par b > c. En inversant l'arc (c,a), on peut insérer a dans OM(T') sans inversion supplémentaire, d'où [c, T] = i(T') + 1.

Supposons que c domine b. En inversant l'arc (c,b), on revient au cas précédent, d'où [c, T] ≤ i(T') + 2. Soit O un ordre total finissant par c et à distance [c, T] de T. Pour avoir [c, T] = i(T') + 1, il faut rendre T' transitif et n'inverser que l'arc (c,a) parmi les arcs incidents à a. La trace de O sur T' donne un ordre médian OM(T') de T' finissant par c. D'après le lemme 4, b est placé juste avant x dans OM(T'), et OM(T') contient donc ...>b>x>... Mais alors il est impossible de restituer au sommet a sa place dans O sans inverser un arc supplémentaire : placer a avant b oblige à inverser au moins l'arc (x,a), et le placer après b entraîne l'inversion de (a,b). L'égalité [c, T] = i(T') + 1 est donc impossible. par conséquent, on a [c, T] = i(T') + 2.

δ - calcul de [{a,c}, T] et de [{b,c}, T]

Pour mettre a et c globalement à la fin, il est nécessaire et suffisant d'inverser d'une part l'arc (a,b) et d'autre part assez d'arcs dans T' pour mettre c à la fin d'un ordre total sur T'. D'où [{a,c}, T] = [c, T'] + 1.

Pour mettre globalement b et c à la fin, il faut inverser l'arc (c,a) et au moins i(T') arcs. D'après le lemme 9, il existe un ordre médian de T' finissant globalement par b et c ; en insérant a juste avant b et c dans cet ordre, on obtient un ordre total sur T à distance i(T') + 1 et finissant globalement par b et c. D'où [{b,c}, T] = i(T') + 1.

4) La suite des scores commence par 113

Soient a et b les deux sommets de score 1, avec a qui domine b. Posons T' = T-{a} ; b est de score 1 dans T' (voir figure 4).

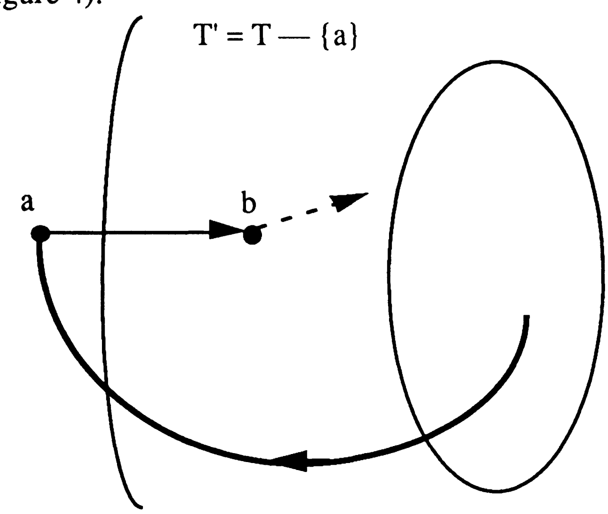

Figure 4

D'après le lemme 7, on sait que i(T) = i(T') si et seulement s'il existe un ordre médian de T' finissant par b ; sinon i(T) = i(T')+1. D'où dans les deux cas i(T) = [b,T'].

D'autre part, on obtient un ordre de T finissant par b à partir d'un ordre de T' finissant par b sans aucune inversion supplémentaire d'arc car a vient naturellement juste avant b. D'où [b, T] = [b, T'] = i(T).

Enfin, pour obtenir un ordre finissant par a, il est nécessaire et suffisant de rendre T' transitif et d'inverser l'arc (a,b) ; d'où [a, T] = i(T')+1.

5) La suite des scores commence par 1 2 2

Appelons a le sommet de score 1. D'après le lemme 8, il peut y avoir deux ou trois sommets de score 2. Nous distinguons quatre cas notés 5-1, 5-2, 5-3 et 5-4 définis ci-après, en fonction du fils de a :

- cas 5-1 : a domine un sommet b de score 2 qui lui-même domine un autre sommet c de score 2 : le sous-tournoi engendré par a, b, c est un circuit ;
- cas 5-2 : il y a trois sommets de score 2, et ces trois sommets dominent a ;
- cas 5-3 : il y a deux sommets de score 2, et ces deux sommets dominent a ;
- cas 5-4 : a domine un sommet c de score 2 qui ne domine aucun autre sommet b de score 2 : le sous-tournoi engendré par a, b, c est transitif (il correspond à b > a > c) ;

5-1 : ce cas est illustré par la figure 5.

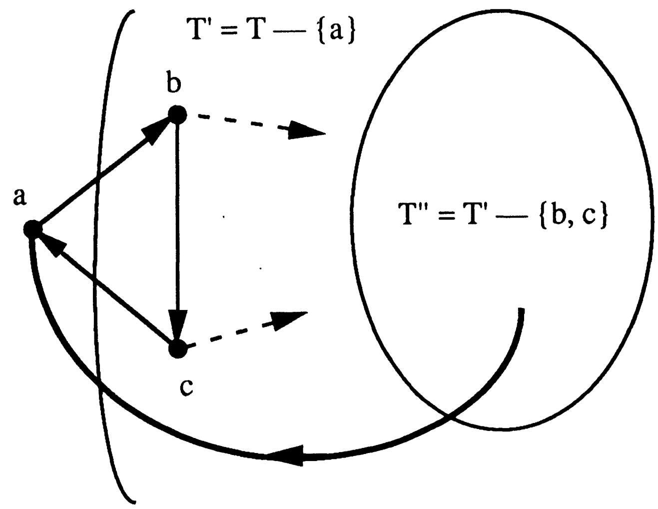

Figure 5

α - calcul de i(T)

Posons T' = T - {a}. Les scores de b et de c dans T' sont respectivement 2 et 1. D'après le lemme 7, i(T) = i(T') si et seulement s'il existe un ordre médian de T' finissant par b, et sinon i(T) = i(T') + 1.

β - calcul de [a, T]

On a [a,T] = i(T') + 1 ; en effet pour obtenir un ordre de T finissant par a il est nécessaire et suffisant d'inverser l'arc (a,b), qui n'est pas dans T', et de transformer T' en un tournoi transitif, ce qui implique au moins i(T') inversions d'arcs de T'.

γ - calcul de [b, T] et de [{a,b}, T]

On a [b, T] = [b, T']. En effet on a clairement [b, T] $\geq$ [b, T'], et par ailleurs si on a un ordre de T' finissant par b, sans inversion supplémentaire on introduit a juste avant b, pour obtenir un ordre de T finissant par b. Ce raisonnement s'applique aussi au calcul de [{a,b}, T] : [{a,b}, T] = [b, T'].

δ - calcul de [{a,c},T]

On a [{a,c}, T] = [c,T'] + 1. En effet, pour obtenir un ordre de T finissant globalement par a et c, il est nécessaire et suffisant d'inverser l'arc (a,b), qui n'est pas dans T', et de transformer T' en un tournoi transitif finissant par c, ce qui implique au moins [c,T'] inversions d'arcs de T'.

ε - calcul de [c, T]

On a [c, T] = [{b,c}, T'] + 1. La preuve est un peu plus longue. Remarquons d'abord l'inégalité suivante : [{b,c}, T'] $\leq$ [c, T'] + 1. En effet, soit T'' défini par T'' = T' - {b,c}, on voit que [c, T'] $\geq$ i(T'') + 1 (c domine un sommet de T''), et [{b,c}, T'] = i(T'') + 2, ce qui prouve la remarque.

Obtenir un ordre finissant par c nécessite toujours l'inversion de l'arc (c,a), qui n'est pas dans T', et aussi d'inverser assez d'arcs de T' pour avoir un ordre sur T' finissant par c ; d'où [c, T] $\geq$ [c, T']+1. Par ailleurs, si on a un ordre à distance minimum de T' et finissant par {b,c}, alors b domine c dans cet ordre ; en inversant l'arc (c,a) et en plaçant a juste avant c et b, on prouve la relation [c, T] $\leq$ [{b,c}, T'] + 1. Si [c, T'] = [{b,c}, T'], la preuve est achevée. Si [c, T'] = [{b,c}, T'] - 1, considérons un ordre sur T' finissant par c à distance [c, T'] de T' (les autres ordres finissant par c à distance supérieure à [c, T'] ne sont trivialement pas intéressants) ; cet ordre finit par ... > b > ... > u > c avec u≠b ; placer a dans cet ordre en laissant c en fin nécessite l'inversion de l'arc (c,a) et en plus, si a vient entre u et c, l'inversion de l'arc (a,b), et si a vient avant u, l'inversion de l'arc (u,a) ; d'où finalement [c, T] ≥ [c, T'] + 2 = [{b,c}, T'] + 1, ce qui achève la démonstration.

ζ - calcul de [d, T] et de [{a,d}, T] pour d≠b, d≠c et d de score 2 (cas d'un vecteur-score commençant par 1222)

Si b domine d, d joue un rôle identique à celui de c, d'où [d, T] = [{b,d}, T'] + 1 et [{a,d}, T] = [d, T'] + 1.

Sinon la situation est illustrée par la figure 6.

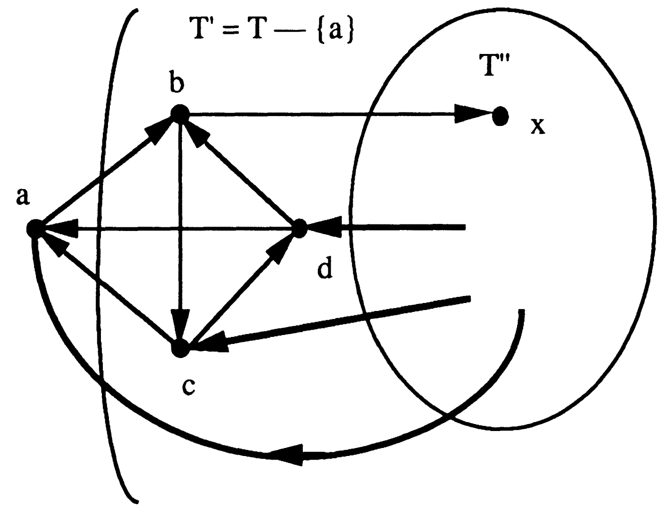

Figure 6

Soit x le sommet dominé par b, autre que c. Posons T'' = T — {a, b, c, d}. Supposons qu'il existe un ordre médian de T'' finissant par x ; il est alors facile de montrer que l'on a i(T') = i (T'') + 1 et [b, T'] = i(T'') + 2 = i(T') + 1. Sinon, n'inverser que les i(T'') arcs nécessaires à rendre T'' transitif et un arc pour détruire le circuit (b,c,d) ne permet pas de détruire les cicuits passant par l'arc (b,x) (il existe au moins un tel circuit puisque x domine au moins un sommet de T'' dans tout ordre médian de T'') ; on a donc dans ce cas i(T') ≥ i(T'') + 2, et il est clair qu'on atteint ce nombre en inversant, outre les i(T'') arcs de T'', les arcs (b,x) et (b,c), ce qui montre de plus la relation [b, T'] = i(T') dans ce cas. Par conséquent, il existe un ordre médian de T'' finissant par x si et seulement si [b, T'] ≠ i(T').

Remarquons que mettre d à la fin d'un ordre défini sur les sommets de T nécessite au minimum l'inversion de i(T'') arcs dans T'', plus les deux arcs dont d est l'origine, plus un arc pour détruire le circuit (a,b,c) : [d, T] ≥ i(T'') + 3.

Si [b, T'] ≠ i(T'), d'après ce qui précède on a i(T') = i(T'') + 1 et il existe un ordre médian de T'' finissant par x, et d'après l'étude de i(T), on a i(T) = i(T') + 1. En choisissant d'inverser (a,b) dans le circuit (a,b,c), on peut placer b juste avant x, et cela est suffisant pour obtenir ce que l'on cherche : [d, T] = i(T'') + 3. D'où finalement [d, T] = i(T) + 1.

Si au contraire [b, T'] = i(T'), aucun ordre médian de T'' ne finit par x et on a les égalités i(T') = i(T) = i(T'') + 2. N'inverser que les arcs nécessaires cités plus haut (les i(T'') de T'', (d,a), (d,b) et un des arcs de (a,b,c)) ne suffit pas à détruire les circuits du type (b,x,y), où y est un sommet de T'' dominé par x après inversion des i(T'') arcs dans T''. D'où [d, T] ≥ i(T'') + 4, et il est trivial d'atteindre ce nombre. Donc dans ce cas, [d, T] = i(T'') + 4 = i(T) + 2.

Quant à [{a,d}, T], cette quantité vaut trivialement [d, T'] + 1.

5-2 : la figure 7 représente la situation correspondant à ce cas.

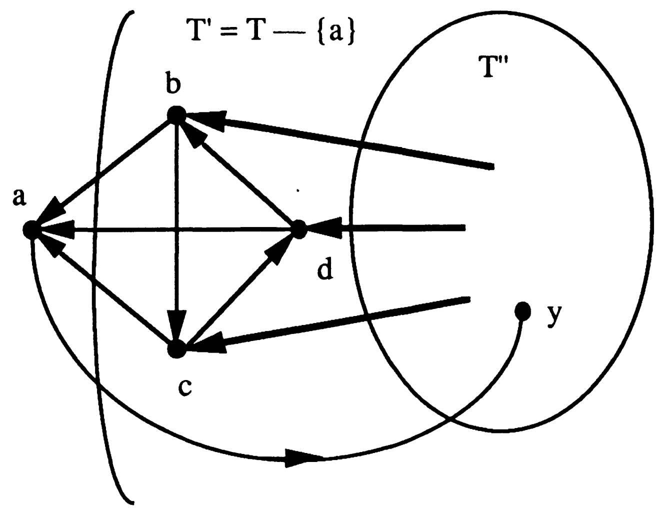

Figure 7

α - calcul de i(T) et de [a, T]

Posons T' = T—{a}. Il est facile de constater que tous les ordres médians de T' finissent par b, c ou d. Il est alors impossible d'insérer a dans un ordre médian de T' sans inverser au moins un arc, d'où i(T) ≠ i(T'). Comme l'inversion du seul arc issu de a suffit pour pouvoir mettre a à l'extrémité de tout ordre médian de T', on a i(T) = i(T') + 1 et aussi [a, T] = i(T).

β- calcul de [x, T] et de [{a,x}, T] pour x de score 2

Ici aussi le résultat [{a,x}, T] = i(T) est trivial pour les trois sommets de score 2. D'autre part, posons T'' = T — {a, b, c, d} ; on a clairement i(T') = i(T'') + 1. Pour mettre b à la fin d'un ordre sur T, il est nécessaire d'inverser les arcs (b,a) et (b,c) (ce qui détruit le circuit (b,c,d)), de rendre transitif T'', et de détruire les circuits (c,a,y) et (d,a,y), où y est l'unique successeur de a dans T. Ceci requiert au moins 3 + i(T'') = i(T) + 1 inversions, et ce nombre est atteint si on décide d'inverser l'arc (a,y). D'où [b, T] = i(T) + 1. Le même raisonnement s'applique à c et d, par symétrie des rôles joués par b, c et d.

5-3 : ce cas est illustré par la figure 8.

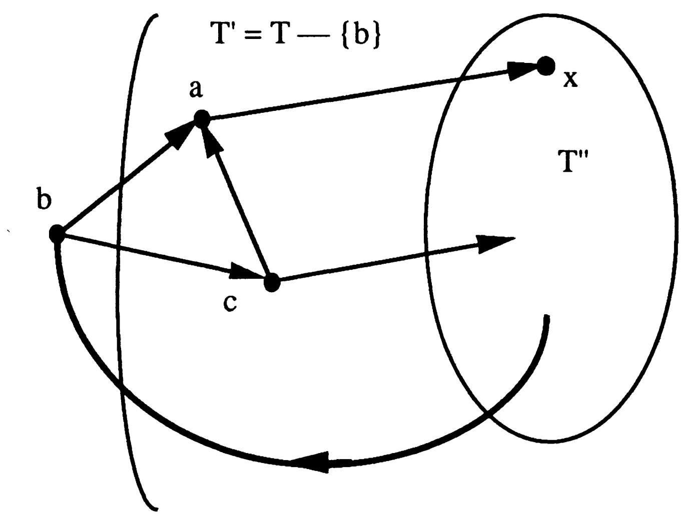

Figure 8

α - calcul de i(T)

Ici on pose T' = T - {b}. Dans T' le score de a est 1 et celui de c est 2. L'application de l'algorithme à T' permet de savoir si [{a,c}, T'] vaut i(T') ou bien i(T')+1, ou sinon, d'après le lemme 5, i(T')+2.

Nous allons prouver que i(T) = i(T') ou i(T')+1 et que i(T) = i(T') si et seulement si [{a,c}, T'] = i(T'). Prouvons tout d'abord cette équivalence. Si [{a,c}, T'] = i(T'), alors on obtient un ordre de T en insérant b, sans inversion d'arc supplémentaire, avant la paire {a,c}. Cet ordre ne nécessite que i(T') inversions d'arcs et est un ordre médian de T. Réciproquement, si i(T) = i(T'), la trace sur T' d'un ordre médian de T, qui est dans tous les cas un ordre de T', est un ordre médian de T', et ne nécessite pas d'inversion d'arc issu de b. C'est donc que l'ordre médian correspondant sur T' finit par {a,c}.

Si [{a,c}, T'] = i(T')+1, on a clairement i(T) ≤ i(T')+1, puisqu'il suffit, comme dans le cas précédent, d'intercaller b avant {a,c}, et d'après l'équivalence précédente, i(T) ≥ i(T')+1.

Enfin, si [{a,c}, T'] = i(T')+2, d'après le lemme 6 il existe un sommet x de T' dominé par a et c et tout ordre médian de T' finit par ... > c > a > x. Partant d'un ordre médian de T', il suffit alors d'inverser l'arc (x,b) pour obtenir un ordre de T utilisant i(T')+1 inversions d'arcs, (et finissant par ... > b > c > a > x), ce qui, d'après l'équivalence établie précédemment, est le minimum que l'on peut espérer. On a donc ici encore i(T) = i(T')+1.

β - calcul de [a, T]

Si on a [{a,c}, T'] = i(T') (respectivement i(T')+1), on a aussi i(T) = i(T') (respectivement i(T')+1) et un ordre médian de T finissant par a. D'où [a,T] = i(T) dans ces deux cas. Dans le cas [{a,c}, T'] = i(T')+2, nous avons vu que les ordres médians de T finissent par ... > b > c > a > x : l'inversion de l'arc (a,x) dans cet ordre médian donne un ordre de T finissant par a à i(T)+1, et aucun ordre médian (à i(T)), ne finit par a.

γ - calcul de [b, T], de [{a,b}, T] et de [{a,c}, T]

En dehors des arcs de T', il faut inverser les deux arcs issus de b pour obtenir un ordre finissant par b : [b, T] = i(T')+2. De même, il est nécessaire et suffisant d'avoir un ordre sur T' finissant par a et d'inverser l'arc (b,c) pour obtenir un ordre finissant globalement par a et b : [{a,b},T] = [a,T']+1. Enfin, [{a,c}, T] = [{a,c},T'] ; en effet on ne peut espérer diminuer le nombre d'arcs à inverser en augmentant le nombre de sommets, et, par ailleurs, ayant {a,c} à la fin de T', il suffit d'intercaler b juste avant ces deux sommets, ce qui se fait sans inversion d'arc supplémentaire.

δ - calcul de [c, T]

Nous décomposons cette étude en trois cas, selon la valeur de [{a,c},T'].

* [{a,c},T'] = i(T') est équivalent, d'après ce que nous avons vu précédemment, à i(T) = i(T'), et tout ordre médian de T finit par ... > c > a. Il faut donc au moins une inversion supplémentaire pour avoir un ordre de T finissant par c, et l'inversion, à partir de l'ordre médian, de l'arc (c,a) est suffisante. On a dans ce cas : [c,T] = i(T)+1.

* Si on a [{a,c},T'] = i(T')+1, alors i(T) = i(T')+1. D'après le résultat plus haut, il existe pour T des ordres finissant par {a,c} nécessitant [{a,c}, T] = [{a,c},T'] = i(T) inversions, et donc i(T)+1 inversions suffisent pour mettre c à la fin de T : [c,T] ≤ i(T)+1. Prouvons par l'absurde que l'on ne peut faire moins. Supposons qu'il existe un ordre médian OM de T finissant par c, et appelons x le sommet dominé par a dans T (donc x≠b et x≠c). OM finit nécessairement par ... > a > x > c. En effet, d'après le lemme 4, a ne peut pas précéder immédiatement c. Il ne peut pas non plus y avoir plus d'un sommet entre eux : sinon, en déplaçant a pour le mettre à la fin, on économiserait globalement au moins une inversion ; et ce sommet ne peut être que x, car a ne peut précéder qu'un sommet qu'il domine dans T (lemme 4). Considérons alors l'ordre OM' obtenu à partir de OM en rejetant a à la fin : OM' finit par ... > x > c > a. OM' est encore un ordre médian (il faut inverser (a, x), mais non plus l'arc (c, a)...). Or pour obtenir OM', il a fallu inverser l'arc (x, b) ; la trace de OM' sur T' est donc un ordre total à distance au plus

i(T) - 1 = i(T') finissant par {a, c}, ce qui est contraire à l'hypothèse. Donc ici [c, T] = i(T) + 1.
* [{a,c}, T'] = i(T')+2, et donc à nouveau i(T) = i(T')+1. Considérons T'' = T'-{a,c}. D'après le lemme 6, on a i(T'') = i(T'). Or pour avoir un ordre sur T', il faut avoir un ordre sur T'', ce qui exige au moins i(T'') inversions, et, pour que l'ordre finisse par c, il faut de plus inverser les deux arcs issus de c qui ne font pas partie des arcs précédents. On a donc [c, T'] = i(T') + 2. Cependant, dans cet ordre, a ne suit pas directement c, sinon, en rétablissant le sens initial de l'arc (c,a), on aurait [{a,c}, T'] < i(T') + 2 ; par suite, on ne peut pas placer b sans faire d'inversion supplémentaire. On a donc [c,T] ≥ i(T')+3. Or on sait obtenir le résultat cherché avec ce nombre d'inversions, en partant d'un ordre de T' finissant par {a,c}, puis en inversant l'arc (c,a) et en plaçant, sans inversion supplémentaire, b avant {a,c}. D'où ici [c,T] = i(T) + 2.

5-4 : ce dernier cas est illustré par la figure 9.

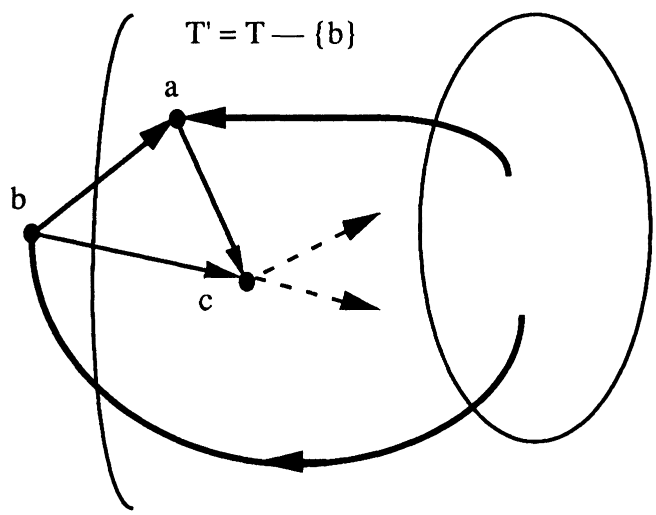

Figure 9

Remarquons d'abord que ce cas n'est compatible qu'avec un vecteur-score commençant par 1223... ; il y a donc exactement deux sommets de score 2 : b et c.

Posons T' = T—{b}. Les inégalités suivantes sont immédiates, d'après le lemme 5 : i(T') ≤ i(T) ≤ i(T')+2. Par des raisonnements analogues à ceux qui ont été détaillés précédemment, on obtient :

α - calcul de i(T)

i(T) = i(T') si et seulement si [{a,c}, T'] = i(T'), et on obtient pour seuls ordres médians de T des ordres finissant par ... > b > a > c. Par ailleurs, on a toujours i(T) ≤ i(T')+1 : d'après le lemme 6, [{a,c}, T'] ≤ i(T')+1 (car l'arc entre a et c est orienté de a vers c), et à partir d'un ordre de T' se terminant par {a,c}, b se place sans inversion supplémentaire.

β - calcul de [c, T] et de [{a,c}, T]

Dans le cas i(T) = i(T'), tout ordre médian finit par ... > b > a > c ; dans le cas i(T) = i(T')+1, il existe aussi des ordres médians finissant par ... > b > a > c. On a donc dans les deux cas [c, T] = i(T). De la même façon [{a,c},T] = i(T).

γ - calcul de [b, T] et de [{a,b}, T]

La trace sur T' d'un tel ordre finissant par b est un ordre de T', qui exige donc au moins i(T') inversions ; pour finir par b il faut de plus inverser les deux arcs issus de b, et tout ceci est suffisant. On a donc [b, T] = i(T')+2. De même, la trace sur T' d'un ordre finissant globalement par a et b donne un ordre de T' finissant par a, ce qui requiert donc [a,T'] inversions. On doit de plus inverser l'arc (b,c) et tout ceci est suffisant. D'où [{a,b}, T] = [a, T']+1.

δ - calcul de [a, T]

Il faut ici distinguer entre plusieurs cas.

Si i(T) = i(T'), les ordres médians de T finissent tous par ... > a > c. L'inversion de l'arc (a,c) conduit au résultat souhaité avec i(T)+1 inversions, et on ne peut l'obtenir avec seulement i(T) inversions. On a donc [a, T] = i(T)+1.

Considérons maintenant le cas i(T) = i(T')+1 et distinguons à nouveau des sous-cas en fonction de la valeur de [a, T'].

- Si [a, T'] = i(T'), en inversant ces i(T') arcs plus l'arc (b,c), on obtient le résultat avec i(T')+1 inversions, et on ne peut faire moins. D'où [a, T] = i(T') + 1 = [a, T'] + 1.

- Si [a,T'] = i(T') + 1, par la même construction que ci-dessus, on obtient le résultat avec au plus i(T) + 1 inversions, et on ne peut l'espérer avec i(T) inversions seulement. Supposons en effet qu'il existe un ordre médian O de T finissant par a : sa trace sur T' est un ordre finissant par a et nécessitant i(T') + 1 = i(T) inversions dans T'. Cet ordre ne finit sûrement pas par ... > c > a, d'après le lemme 4. Il y a donc au moins un sommet x entre c et a. b doit encore trouver sa place dans O : que ce soit après ou avant x, cela implique au moins une inversion supplémentaire ; on dépasse alors nécessairement les i(T) inversions. Par conséquent, si [a,T'] = i(T') + 1, on a [a,T] = i(T) + 1 = i(T') + 2 = [a, T'] + 1. ♦

Remarques.
1) Il est facile d'adapter l'algorithme 2 pour obtenir, outre l'indice de Slater, un ordre médian.
2) Comme pour l'algorithme 1, l'hypothèse simplificatrice selon laquelle les sommets sont initialement rangés par ordre des scores croissants n'est pas indispensable. On peut s'inspirer de l'étude de la complexité faite plus haut pour savoir ce qu'il faut faire au moment de l'acquisition des données pour qu'il en soit ainsi...

THÉORÈME 11. La détermination de l'indice de Slater des tournois T d'ordre n et de paramètre σ(T) = 2 est linéaire en n.

Preuve. Le théorème 11 est conséquence directe du lemme 10. ♦

CONJECTURE. Soit σ un entier donné. Il existe un algorithme polynomial déterminant l'indice de Slater i(T) et un ordre médian pour tout tournoi T à n sommets vérifiant $\sigma(T) = \sigma$.

Notons que cette conjecture n'est pas incompatible avec l'éventualité que le problème de P. Slater soit NP-difficile, l'algorithme de la conjecture pouvant par exemple être exponentiel par rapport à σ.

4. APPLICATION A UNE MÉTHODE ARBORESCENTE DE RECHERCHE D'UN ORDRE MÉDIAN

Le problème de Slater n'étant pas connu pour être polynomial, les méthodes pour déterminer la valeur exacte de l'indice de Slater et au moins un ordre médian sont des méthodes dont la complexité croît exponentiellement en fonction de la taille des données. De nombreuses méthodes – exactes ou approchées – ont été proposées pour la résolution de ce problème (voir [2]), dont les méthodes arborescentes ("Branch and Bound") qui présentent l'avantage de pouvoir calculer tous les ordres médians. L'explosion combinatoire de ces méthodes, inévitable si on veut la solution exacte, se traduit par un temps de calcul et une place mémoire rapidement élevés. De ce fait, l'ordre des tournois sur lesquels on peut appliquer ces méthodes reste relativement faible. Il est donc important de réduire au maximum le nombre de branches de l'arborescence à explorer, de façon à accélérer la méthode et pouvoir l'appliquer à des tournois plus gros. Un moyen d'y parvenir consiste à concevoir une fonction d'évaluation la plus fine possible, afin d'éliminer de l'exploration le plus grand nombre possible de branches inutiles, le plus tôt possible. Dans cette dernière partie, nous indiquons comment on peut utiliser le paramètre $\sigma(T)$ pour gagner du temps, en l'appliquant à la méthode arborescente décrite dans [1].

Rappelons brièvement le principe du Branch and Bound développé dans [1]. Il consiste à construire un arbre dont, à chaque étape de la construction, les feuilles sont des débuts d'ordres totaux, jusqu'à tant que l'une d'entre elles contienne tous les sommets du tournoi T dont on cherche un ordre médian. Chaque sommet de l'arbre est pourvu d'une évaluation mesurant sa qualité, et on décide d'explorer les descendants de la feuille possédant la plus petite évaluation. On augmente alors d'un sommet de T le début d'ordre total correspondant à cette feuille, pour obtenir un début d'ordre total comprenant un sommet de T de plus.

Plus précisément, soit $X = \{x_1, \ldots, x_n\}$ l'ensemble des sommets d'un tournoi T. Un sommet de l'arbre que l'on construit est de la forme $S = x_{i_1}>x_{i_2}>\ldots>x_{i_p}$ (voir figure 10) ; son père dans l'arbre est $x_{i_1}>x_{i_2}>\ldots>x_{i_{p-1}}$, et ses fils sont de la forme $x_{i_1}>x_{i_2}>\ldots>x_{i_p}>x_{j_k}$ ($1 \leq k \leq n-p$) pour tous les sommets x_{j_k} de $X-\{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$ (en fait, il n'est pas toujours utile de tous les considérer, comme le montre le lemme 4 ; voir [1] pour plus de détails).

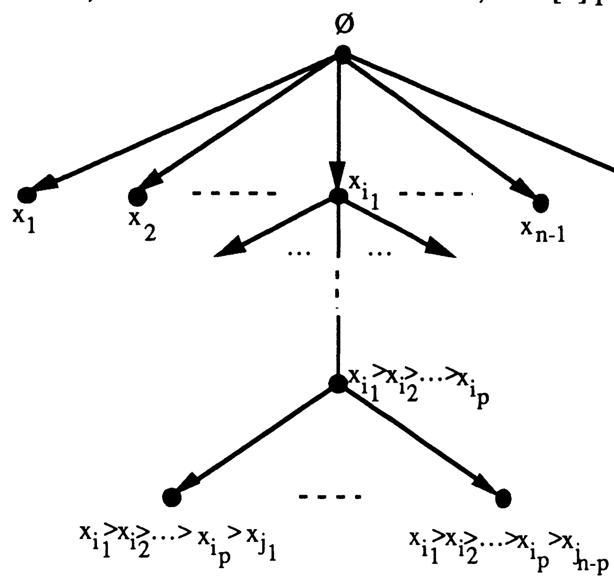

Figure 10

La qualité de chaque sommet S de l'arborescence est mesurée par le nombre d'arcs qu'il faut inverser pour obtenir ce début d'ordre total représenté par S, c'est-à-dire le nombre d'arcs entrant en x_{i_1} + le nombre d'arcs allant de $\{x_j, j \neq i_1\}$ vers x_{i_2} + ... + le nombre d'arcs allant de $\{x_j \text{ avec } j \notin \{i_1, \ldots, i_{p-1}\}\}$ vers x_{i_p}.

L'avantage de cette évaluation E_1 est que celle-ci est très simple à calculer, puisque l'évaluation d'un sommet $x_{i_1}>x_{i_2}>\ldots>x_{i_p}$ peut se faire à partir de celle de son père :

$$E_1(x_{i_1}>x_{i_2}>\ldots>x_{i_p}) = E_1(x_{i_1}>x_{i_2}>\ldots>x_{i_{p-1}}) + \sum_{j \notin \{i_1,\ldots,i_p\}} T(x_j, x_{i_p})$$

où $T(x_j, x_{i_p})$ vaut 1 si l'arc (x_j, x_{i_p}) existe dans le tournoi T, 0 sinon. De plus, si on choisit à chaque itération de développer une feuille de plus petite évaluation, le premier ordre constitué englobant tous les sommets du tournoi de départ (c'est-à-dire une feuille située à une profondeur n) fournit un ordre médian, et il n'est pas nécessaire, contrairement à certaines autres méthodes, de poursuivre la recherche (sauf si on veut tous les ordres médians).

Elle a cependant un défaut : elle ne fait que prendre en compte ce qu'il est nécessaire d'inverser pour construire le début d'ordre total représenté par une feuille $S = x_{i_1} > x_{i_2} > \ldots > x_{i_p}$ de l'arborescence, sans anticiper sur ce qu'il restera à inverser pour compléter S en un ordre total sur l'ensemble des sommets du tournoi. Cette lacune peut être éliminée à l'aide du paramètre σ. En effet, soit O_S un ordre total commençant par S : $O_S = x_{i_1} > x_{i_2} > \ldots > x_{i_p} > \ldots$, à distance minimum de T. Pour construire un tel ordre à partir de S, il est nécessaire et suffisant de rendre transitif le sous-tournoi engendré par les sommets x_j ne figurant pas dans S : $j \notin \{i_1, \ldots, i_p\}$. Soit T_S ce sous-tournoi. Pour rendre T_S transitif, il faut inverser au moins $\sigma(T_S)$ arcs. On peut donc proposer comme nouvelle évaluation de la feuille S de l'arborescence la fonction E_2 suivante, qui tient compte à la fois de ces deux composantes (ce qu'il faut inverser pour construire S et ce qu'il faudra au minimum inverser pour compléter S en un ordre sur l'ensemble des sommets de T) :

$$E_2\left(S = x_{i_1} > x_{i_2} > \ldots > x_{i_p}\right) = \sum_{k=1}^{p} \sum_{j \notin \{i_1, \ldots, i_k\}} T\left(x_j, x_{i_k}\right) + \sigma\left(T_S\right) = E_1(S) + \sigma(T_S),$$

où T_S est le sous-tournoi engendré par les sommets x_j pour $j \notin \{i_1, \ldots, i_p\}$.

Cette nouvelle évaluation peut encore partiellement se calculer à partir de l'évaluation du père (en fait, le premier terme, puisqu'il s'agit encore de E_1). Quant au σ, il faudra reconstituer les scores de T_S pour pouvoir l'évaluer. Mais la perte de temps engendrée par cette reconstitution est largement compensée, comme l'indique le tableau ci-dessous, par le gain de temps que procure la nouvelle évaluation.

n	t_1 (sans σ) (en secondes)	t_2 (avec σ) (en secondes)	t_2 / t_1
12	176	53	30,11%
13	1126	106	9,41 %
14	13800	300	2,17 %

Ce tableau indique le temps CPU nécessaire pour déterminer l'indice de Slater de 200 tournois engendrés aléatoirement (l'orientation des arcs est tirée au hasard, avec la probabilité 0,5 pour chaque direction). Le premier temps t_1 correspond à la méthode arborescente présentée dans [1] sans l'introduction du paramètre σ (évaluation E_1) ; t_2 correspond à cette méthode modifiée pour tenir compte de σ (évaluation E_2) ; la troisième colonne donne le rapport $\frac{t_2}{t_1}$ des temps. Excepté la différence dans les fonctions d'évaluation, les deux algorithmes sont rigoureusement les mêmes, et ont été testés dans les mêmes conditions, sur le même jeu de tests. En particulier, la même solution heuristique (calculée à l'aide de la méthode de A.F.M. Smith et C.D. Payne [10]) a servi pour chaque tournoi à initialiser la borne supérieure de ces deux méthodes.

Un autre paramètre important pour ce type de méthodes est le nombre de sommets que comporte l'arbre de recherche, car il correspond à la place mémoire utilisée. Celle-ci croissant de façon exponentielle avec la taille des données, elle contribue, plus encore peut-être que le temps de résolution, à limiter la taille des données auxquelles appliquer les méthodes arborescentes (c'est du reste à cause de la taille mémoire que nous n'avons pu envisager des valeurs de n supérieures à 14 pour la méthode sans σ...). Le tableau suivant donne, pour les mêmes tests que précédemment, les nombres moyens $\overline{m}_1$ et $\overline{m}_2$ de sommets engendrés par l'arborescence et les nombres maximum Max_1 et Max_2 de sommets rencontrés au cours des 200 essais, pour chacune des deux évaluations. Ces nombres sont complétés par leurs rapports en pourcentage.

n	$\overline{m}_1$	$\overline{m}_2$ (sans σ)	$\overline{m}_2 / \overline{m}_1$ (avec σ)	Max_1	Max_2 (sans σ)	Max_2/Max_1 (avec σ)
12	1960	168	8,57 %	9003	647	7,19 %
13	5106	305	5,97 %	20613	1232	5,98 %
14	18485	785	4,25 %	135490	4856	3,58 %

On constatera que le gain en nombre de sommets engendrés dans l'arborescence croît avec n. Ceci est dû au fait que l'évaluation incluant σ permet d'éliminer plus tôt un plus grand nombre de feuilles de trop grande évaluation, et donc d'éviter leur exploration inutile, surtout dans les niveaux de faible profondeur. Ce phénomène est d'autant plus sensible que n est élevé. En effet, quand n croît, on peut s'attendre en moyenne à voir l'indice de Slater croître aussi. La première évaluation est alors moins à même, principalement dans les niveaux de profondeur peu élevée, de détecter les feuilles qui conduiront finalement à des inversions en nombre important. Alors que la seconde évaluation, fournissant des valeurs plus élevées, même dans les niveaux peu profonds, reste plus proche des valeurs réelles. Pour bien se rendre compte de cette différence, on peut penser à un cas extrême, celui d'un tournoi T tel que σ(T) = i(T). Imaginons de plus que la solution heuristique initiale soit exacte ou même seulement très proche d'une solution optimale (ce qui n'est pas impossible : dans les tests précédents, sur les 200 tournois engendrés à n=13, la méthode de A.F.M. Smith et C.D. Payne donnait la solution exacte pour 107 cas, et une solution nécessitant 1 inversion superflue pour 72 cas). La seconde évaluation conduira alors la recherche directement ou presque vers une solution optimale, détectée (même sans le savoir) dès le début, que n soit élevé ou non ; tandis que la première évaluation, n'anticipant pas sur le nombre d'arcs restant à inverser, pourra très bien fournir un ordre médian rapidement si n est faible, mais, pour n plus grand, sera incapable d'éliminer des feuilles de faible profondeur, car leur évaluation sera beaucoup plus petite que l'indice de Slater, et donc par trop imprécise pour être efficace. Ce n'est qu'après les avoir explorées qu'on pourra s'apercevoir qu'elles n'étaient pas fructueuses...

Une conséquence de ce gain en place et en temps est qu'il est envisageable d'augmenter l'ordre des tournois dont on veut déterminer un ordre médian. Ainsi, alors que les limites en place et en temps que nous nous étions imposées ne nous permettaient pas d'envisager une série d'essais à n=15 pour l'évaluation sans σ, l'évaluation avec σ nous a permis de déterminer l'indice de Slater d'une série de 200 tournois pour n=15, n=16 et n=17 sans sortir des limites fixées. Le tableau suivant montre l'évolution du temps t, du nombre moyen $\overline{m}$ et du nombre maximum Max de sommets de l'arborescence pour ces trois séries.

n	t(secondes)	$\overline{m}$	Max
15	770	1906	9384
16	2143	4420	23797
17	7560	12632	94042

On notera que même pour n=17, le temps et la place mémoire nécessaires pour mener à bien les calculs restent largement inférieurs à ceux nécessaires pour n=14 sans le paramètre σ.

REMERCIEMENTS

Les auteurs souhaitent remercier Alain Guénoche (CNRS, Marseille) pour les discussions stimulantes (entre autres lors de TRAP 3 [4]) qui se trouvent à l'origine de cette étude, ainsi que le rapporteur anonyme dont les suggestions ont permis d'améliorer la lisibilité du présent texte.

BIBLIOGRAPHIE

[1] BARTHELEMY J.-P., GUENOCHE A., HUDRY O., "Median linear orders : heuristics and a branch and bound algorithm", *European Journal of Operational Research* 41 (1989), 313-325.
[1] BARTHELEMY J.-P., MONJARDET B., "The median procedure in cluster analysis and social choice theory", *Mathematical Social Sciences* 1 (1981), 235-267.
[3] CHARON-FOURNIER I., GERMA A., HUDRY O., "Encadrement de l'indice de Slater d'un tournoi à l'aide de ses scores", *Mathématiques, Informatique et Sciences Humaines* 118 (1992).
[4] GUENOCHE A., "Order at minimum distance of a valued tournament", présenté à la Table Ronde *Modélisation, Analyse et Agrégation des Préférences et des Choix (TRAP 3)* (1988), Marseille-Luminy.
[5] KARP R.M., "Reducibility among combinatorial problems" in *Complexity of computer computations*, Miller et Tatcher éditeurs, Plenum Press, New York, 1972, 85-103.
[6] LANDAU H.G. "On dominance relations and the structure of animal societies III. The condition for a score structure", *Bulletin of Mathematical Biophysics* 13 (1953), 1-19.
[7] MOON J.W., *Topics on tournaments*, Holt, New York, 1968.
[8] REMAGE R., THOMPSON W.A., "Maximum likelihood paired comparison rankings", *Biometrika* 53 (1966), 143-149.
[9] SLATER P. "Inconsistencies in a schedule of paired comparisons", *Biometrika* 53 (1961), 143-149.
[10] SMITH A.F.M., PAYNE C.D., "An algorithm for determining Slater's i and all nearest adjoining orders", *British Journal of Mathematical and Statistical Psychology* 27 (1974), 49-52.
[11] WAKABAYASHI Y., *Aggregation of binary relations : algorithmic and polyhedral investigations*, PhD Thesis, Augsbourg, 1986.
[12] YOUNGER D.H., "Minimum feedback arc sets for a directed graph", *IEEE Trans. of the profes. tech. group in circuit theory* 10, 2 (1963), 238-245.