Association Mathématique du Québec

# Article

## Agrégation d'évaluations partielles par jury

François Bédard,
GIRO inc.

On connaît tous certaines épreuves où les candidats sont évalués par un jury : des sports olympiques, les expo-sciences et autres concours scientifiques, certaines séries télévisées, etc. Toutes ces épreuves doivent résoudre un problème commun : à partir des notes accordées par les juges aux candidats, il faut obtenir un classement final des participants, ou à tout le moins un gagnant.

On peut d'abord formaliser un peu plus le problème à résoudre. Le jury est composé de $J$ juges qui évaluent $C$ candidats. On notera $N_{j,c}$ la note accordée par le juge $j$ au concurrent $c$. Il faut alors agréger ces notes pour obtenir un classement final qu'on représentera par la permutation $P_1, P_2, ..., P_C$ où $P_k$ est le concurrent en $k^e$ position du classement final. Si on considère la possibilité d'égalités dans le classement, une simple permutation n'est plus suffisante pour représenter le classement final. Nous considérerons donc que le classement final est *l'ensemble* des permutations qui correspondent aux classements équivalents, mais en général, on en parlera comme d'une seule permutation.

Il existe plusieurs méthodes d'agrégation reconnues, dont on fera d'abord un survol. Mais toutes ces méthodes sont basées sur la contrainte que *tous* les juges doivent évaluer *tous* les candidats. Que faire si des considérations logistiques empêchent de respecter cette contrainte ? Ce contexte requiert le développement d'une nouvelle méthode d'agrégation, qui sera le sujet de cet article.

## Méthode directe

La méthode la plus simple et la plus directe pour obtenir le classement global est de trier les candidats en ordre décroissant de la somme des notes qui leur ont été accordées par les juges. La note globale du concurrent $c$ est alors :

$$G_c = \sum_{j=1}^{J} N_{j,c} \ .$$

Et bien sûr, la permutation $P_1, P_2, ..., P_C$ est choisie telle que

$$\forall k \in \{1, 2, ..., C-1\} : G_{P_k} \geq G_{P_{k+1}} \ .$$

Cette méthode est de loin la plus naturelle et la plus employée. Elle donne généralement de très bons résultats lorsque le barême d'évaluation est très précis et que les juges sont des professionnels du domaine. Ces deux conditions assurent une grande uniformité dans les notes accordées par les différents juges.

Par contre, la méthode directe est déficiente lorsque ces conditions ne sont pas remplies. Les écarts entre les méthodes d'évaluation des différents juges résultent en un déséquilibre dans l'influence de chaque juge sur le résultat final. Ainsi, si tous les juges ont accordé des notes variant uniformément entre 9 et 10, sauf un qui a accordé des notes variant uniformément entre 5 et 10, l'influence de ce dernier juge sur le résultat final est 5 fois supérieure à celle de chacun des autres juges.

## Méthodes de normalisation linéaire

Les méthodes de normalisation linéaire consistent à appliquer une transformation linéaire (ou affine, ce qui est équivalent) aux notes des juges avant de les additionner pour obtenir le classement global. La transformation sert à s'assurer que chaque juge a une influence équivalente sur le classement global.

Par exemple, on peut normaliser à l'aide de la moyenne $\mu_j$ et de l'écart-type $\sigma_j$ des notes de chaque juge $j$ :

$$N'_{j,c} = \frac{N_{j,c} - \mu_j}{\sigma_j} \ .$$

Bien sûr, si un juge $j$ a donné la même note à tous les candidats, $N'_{j,c}$ est indéfini pour tous les candidats $c$. Mais dans ce cas, on n'a qu'à ignorer complètement ce juge...

On construit alors la permutation $P_1, P_2, ..., P_C$ d'une façon similaire à la méthode directe, mais en modifiant la définition de $G_c$ pour utiliser les notes normalisées :

$$G_c = \sum_{j=1}^{J} N'_{j,c}$$
$$\forall k \in \{1, 2, ..., C-1\} : G_{P_k} \geq G_{P_{k+1}} \ .$$

De façon générale, les méthodes de normalisation linéaires sont trop peu naturelles pour être utilisées dans des compétitions grand public, mais sont intéressantes dans d'autres contextes.

Bien que les méthodes de normalisation linéaires réduisent l'influence des écarts entre les méthodes d'évaluation des différents juges, elles sont impuissantes devant un autre problème : la manipulation des résultats. Il est connu que toutes les méthodes d'agrégation peuvent être affectées par

des manipulations malveillantes, mais la méthode directe et toutes les méthodes de normalisation linéaires sont parmi celles qui permettent le plus facilement à un juge de favoriser indûment un ou des candidats, ou encore d'en défavoriser d'autres.

## Règle de Borda

Le problème intrinsèque avec la méthode directe et les méthodes de normalisation linéaire, qui permettent de favoriser ou défavoriser indûment un concurrent, est connu depuis fort longtemps. Déjà, en 1784, Borda a publié une méthode qui réduisait ce problème [2].

À partir des notes de chaque juge, on obtient un classement des candidats par ce juge, c'est-à-dire une permutation $P_{j,1}, P_{j,2}, ..., P_{j,C}$ où $P_{j,k}$ est le concurrent en $k^e$ position du classement du juge $j$. Bien sûr, on doit avoir :

$$\forall j \in \{1, 2, ..., J\}; \forall k \in \{1, 2, ..., C-1\} : N_{j,P_{j,k}} \geq N_{j,P_{j,k+1}} \ .$$

Par la suite, on assigne un nombre entier de points à chaque position, soit $C-1$ points pour la $1^{re}$ position puis en décroissant jusqu'à 0 points pour la $C^e$ position. On construit alors la permutation de façon similaire aux deux méthodes précédentes, mais en modifiant la définition de $N'_{j,c}$ et de $G_C$ :

$$N'_{j,c} = C - (k : P_{j,k} = c) \ ,$$

$$G_c = \sum_{j=1}^{J} N'_{j,c} \ ,$$

$$\forall k \in \{1, 2, ..., C-1\} : G_{P_k} \geq G_{P_{k+1}} \ .$$

La règle de Borda est donc essentiellement une méthode de normalisation *non* linéaire où seules les positions dans le classement de chaque juge sont utilisées. Les valeurs numériques des notes des juges sont complètement inutiles et bien qu'on ait défini la règle à partir de notes numériques, chaque juge pourrait fournir directement un classement sans notes. Par conséquent, la règle de Borda peut éliminer complètement l'influence des écarts entre les méthodes d'évaluation des différents juges. Elle est également très simple à comprendre et peut sans difficulté être utilisée dans n'importe quelle compétition.

Évidemment, la règle de Borda n'empêche pas un juge d'avantager ou de désavantager un concurrent, mais elle l'empêche de le faire plus qu'un autre juge. Dans certains cas pourtant, ce n'est qu'une mince consolation...

## Méthode de la position majoritaire

La méthode de la position majoritaire repose d'abord sur la règle de Borda en utilisant les positions plutôt que les notes. Mais le classement final se concentre sur les positions « centrales » de chaque concurrent et ignore délibérément les positions « extrêmes ». C'est cette amélioration qui rend très difficile la manipulation des résultats par les juges.

Pour commencer, les notes de chaque juge sont utilisées pour obtenir un classement des candidats pour ce juge comme avec la règle de Borda. Par la suite, chaque concurrent est évalué selon la médiane – et non la somme ou la moyenne – des positions accordées par les différents juges. On construit alors la permutation $P_1, P_2, ..., P_C$ de façon similaire aux trois méthodes précédentes, mais en modifiant la définition de $G_C$ et en inversant le tri puisqu'on veut obtenir les plus petits $G_C$ en premier :

$$G_C = médiane_{j\in\{1,2,...,J\}}(k : P_{j,k} = c) \ ,$$

$$\forall k \in \{1, 2, ..., C-1\} : G_{P_k} \leq G_{P_{k+1}} \ .$$

Même si la méthode de la position majoritaire est assez complexe à saisir, elle est utilisée dans certaines compétitions sportives, dont le patinage artistique. Il s'agit d'une des méthodes d'évaluation les plus justes puisqu'elle évite les pièges les plus communs.

## Contraintes relatives aux juges et aux candidats

Toutes les méthodes de classement ci-haut se basent sur une contrainte qui contribue à leur force, mais qui peut être difficile à respecter dans certains contextes : *tous* les juges doivent évaluer *tous* les candidats.

Si chaque juge n'évalue qu'un sous-ensemble des candidats, seule la méthode directe peut être utilisée pour agréger les résultats. Les normalisations, linéaires ou non, ne sont pas utilisables dans ce contexte. En effet, la transformation appliquée aux notes dépend beaucoup trop du sous-ensemble de candidats évalués. Par exemple, un juge qui a évalué uniquement les meilleurs candidats désavantagera les moins bons candidats de son sous-ensemble en leur accordant des notes normalisées faibles. La règle de Borda et la méthode de la position majoritaire sont encore pires dans ce contexte puisque les positions des candidats ne sont plus comparables d'un juge à l'autre étant donné que les candidats ont été comparés à des candidats différents.

Dans certains cas, le nombre élevé de candidats et le manque de disponibilité des juges pose un problème pratique. Il est difficile de s'assurer que *tous* les juges évaluent *tous* les candidats. J'ai été confronté à ce problème lors d'une expo-sciences avec des élèves du primaire. Il y avait 12 candidats pour chaque niveau ($4^e, 5^e, 6^e$) et un certain nombre de juges assignés à chaque niveau. Or, les juges n'avaient pas le temps d'évaluer les projets et présentations des 12 candidats dans le peu de temps qui leur était accessible.

Il a donc fallu développer une autre technique d'agrégation des résultats qui tienne compte de cette contrainte.

## Théorie du choix social

Pour être équitable, la méthode d'agrégation à utiliser doit respecter certains principes de base. D'abord, chaque juge doit évaluer seulement un sous-ensemble des candidats, selon ce qui est raisonnable en fonction du temps disponible. Ensuite, le barème d'évaluation laissant beaucoup de place à la subjectivité et le jury n'étant pas composé de juges professionnels, il faut une méthode qui ne soit pas directement basée sur les valeurs numériques des notes des juges. De plus, il est toujours préférable de limiter la possibilité de manipulation des résultats, bien que ça n'apparaisse pas être un problème important dans le cadre d'une expo-sciences de niveau primaire.

L'élaboration de différentes méthodes pour déterminer un classement final à partir d'évaluations d'un jury a donné naissance à la théorie du choix social, qui pourrait nous être utile. En particulier, cette théorie définit plusieurs « bonnes propriétés » d'une méthode d'agrégation. Truchon [7] fait un survol intéressant de ces bonnes propriétés pour analyser la méthode d'agrégation utilisée en patinage artistique.

Un théorème qui s'applique à n'importe quelle méthode d'agrégation est le théorème d'Arrow [1]. Il stipule que dès qu'il y a au moins trois candidats et au moins deux juges, il n'existe pas de méthode d'agrégation satisfaisant les bonnes propriétés d'universalité (le classement final doit toujours être uniquement défini bien que les égalités soient permises), de non-dictature (aucun juge ne dicte le classement final indépendamment des autres juges), d'unanimité (lorsque tous les juges classent un certain candidat devant un certain autre, le classement final doit faire de même) et d'indifférence des candidats non pertinents (si on modifie les classements des juges mais sans jamais changer l'ordre relatif de deux candidats spécifiques, ces deux candidats doivent être classés dans le même ordre dans le nouveau classement final). Ce théorème montre donc qu'une méthode « parfaite » n'existe pas pour résoudre notre problème, ce qui, bien sûr, ne nous empêche pas de définir une méthode « imparfaite » qui possède plusieurs bonnes propriétés.

Malheureusement, les définitions de ces propriétés et les théorèmes qui en découlent sont pratiquement tous basés sur la contrainte que *tous* les juges ont évalué *tous* les candidats. Par conséquent, on a peu à tirer de cette théorie pour résoudre notre problème.

Par exemple, une approche très connue, appelée méthode Condorcet et publiée en 1785 [4], propose que si une majorité absolue de juges a classé un candidat devant tous les autres candidats, alors ce candidat devrait obtenir la première position dans le classement final. Or, dans le cas qui nous intéresse, il est impossible qu'une majorité absolue de juges ait classé un candidat devant tous les autres, tout simplement parce que les juges n'ont pas évalué tous les candidats. On pourrait être tenté d'adapter la règle pour la restreindre aux candidats que chacun des juges a évalués, mais on obtient alors une règle inadéquate. On peut le montrer en considérant un cas simple où on a 3 candidats A, B et C et 7 juges. Tous les juges classeraient les candidats dans le même ordre A, B,

C, mais chacun des juges n'a évalué que 2 des 3 candidats : 3 juges ont classé A et B, 4 juges ont classé B et C. Une majorité absolue de juges (4 sur 7) a donc classé B devant tous les autres *qu'ils ont évalués*, mais de toute évidence, c'est quand même A qui doit occuper la première position du classement final.

Certaines autres approches de la théorie du choix social peuvent néanmoins servir de base pour la résolution de notre problème. En 1959, Kemeny a publié une approche [6] qui, bien que définie dans le cadre habituel où tous les juges ont évalué tous les candidats, peut être étendue pour s'appliquer au cas où ce cadre n'est pas respecté. Kemeny a proposé de considérer comme mesure de la qualité d'un classement final le nombre de désaccords entre ce classement et les différents classements des juges. Un désaccord est défini ici comme une inversion d'une paire de candidats entre deux classements. Ce qui est intéressant avec cette approche est qu'une telle inversion ne dépend pas des candidats *autres* que les deux de la paire. Elle s'applique donc bien à notre problème où, justement, les candidats *autres* sont différents d'un juge à l'autre.

Dans notre contexte, la seule information comparable d'un juge à l'autre est la comparaison entre deux candidats, c'est-à-dire « *est-ce que le concurrent c est meilleur, équivalent ou pire que le concurrent d ?* ». À partir de cette information, nous décrirons une méthode, que nous appellerons la méthode par comparaison qui permet d'obtenir un classement final.

## Méthode par comparaison

La méthode par comparaison est la solution que nous proposons au problème où chacun des juges n'a évalué qu'un sous-ensemble des candidats. Elle est basée exclusivement sur la comparaison entre deux candidats par chacun des juges. On notera $M_{j,c,d}$ le résultat de la comparaison entre les candidats $c$ et $d$ par le juge $j$ :

$$M_{j,c,d} = \begin{cases} 1, \text{ si le juge } j \text{ considère que } c \text{ est meilleur que } d, \\ 0, \text{ sinon.} \end{cases}$$

Notons d'abord que cette définition s'applique tout aussi bien au cas habituel où chaque juge a évalué tous les candidats qu'au cas où chaque juge n'a évalué qu'un sous-ensemble des candidats. Si le juge $j$ n'a pas évalué à la fois les candidats $c$ et $d$, on a $M_{j,c,d} = 0$.

On remarque que nous avons laissé tomber les comparaisons d'égalité entre deux candidats. En effet, si un juge accorde la même note à deux candidats, on considère qu'il ne fournit aucune information sur la comparaison entre eux. On pourrait bien entendu argumenter que ce juge fournit comme information que ces deux candidats sont « près » l'un de l'autre, mais nous n'avons pas considéré cet aspect qui relève plutôt de la gestion des égalités.

La nature des comparaisons entre candidats permet de nous assurer de l'équité entre les juges. Tout d'abord, ces comparaisons ne fournissent aucune information, directe ou indirecte, sur les candidats qu'un juge n'a pas évalués. Ensuite, elles sont indépendantes de la valeur numérique des notes

accordées par le juge. On n'en retire qu'une idée de position relative, un peu comme dans la règle de Borda ou la méthode de la position majoritaire.

Il reste à déterminer comment on peut agréger ces données en un classement final. L'agrégation peut alors être vue comme un problème d'optimisation. Il faut déterminer le classement final $P_1, P_2, ..., P_C$ qui respecte le mieux possible les comparaisons recueillies.

Comme chaque comparaison entre deux candidats a le même poids, l'objectif est donc d'obtenir le classement final qui maximise le nombre de comparaisons respectées ou qui minimise le nombre de comparaisons contredites.

Heureusement, on peut d'abord prouver que maximiser le nombre de comparaisons respectées est équivalent à minimiser le nombre de comparaisons contredites. Soit $MR$ le nombre de comparaisons respectées et $MC$ le nombre de comparaisons contredites. On a :

$$MR = \sum_{j=1}^{J} \sum_{k=1}^{C} \sum_{l=k+1}^{C} M_{j,P_k,P_l} \ ,$$
$$MC = \sum_{j=1}^{J} \sum_{k=1}^{C} \sum_{l=1}^{k-1} M_{j,P_k,P_l} \ .$$

Donc

$$MR + MC = \sum_{j=1}^{J} \sum_{k=1}^{C} \sum_{l=k+1}^{C} M_{j,P_k,P_l} + \sum_{j=1}^{J} \sum_{k=1}^{C} \sum_{l=1}^{k-1} M_{j,P_k,P_l} \ = \sum_{j=1}^{J} \sum_{k=1}^{C} \sum_{\substack{l=1 \\ l \neq k}}^{C} M_{j,P_k,P_l} \ .$$

Par définition, $M_{j,C,C} = 0$, donc on peut rajouter les termes où $l = k$ dans la somme :

$$MR + MC = \sum_{j=1}^{J} \sum_{k=1}^{C} \sum_{l=1}^{C} M_{j,P_k,P_l} \ .$$

Finalement, puisque $P_1, P_2, ..., P_C$ est une permutation de $\{1, 2, ..., C\}$, on obtient :

$$MR + MC = \sum_{j=1}^{J} \sum_{c=1}^{C} \sum_{d=1}^{C} M_{j,c,d} \ .$$

$MR + MC$ est donc une constante (une fois les évaluations des juges connues) et par conséquent, maximiser $MR$ est équivalent à minimiser $MC$. Pour la suite, nous allons décrire comment minimiser $MC$, ce qui par ailleurs se rapproche de la définition de désaccord minimal de Kemeny.

Les comparaisons des différents juges étant toutes traitées de la même façon, on peut d'abord agréger le tableau à trois dimensions $M$ en une matrice à deux dimensions $M'$ représentant la somme des comparaisons pour les différents juges :

$$M'_{c,d} = \sum_{j=1}^{J} M_{j,c,d} \ .$$

On doit alors minimiser

$$MC = \sum_{k=1}^{C} \sum_{l=1}^{k-1} M'_{P_k,P_l} \ .$$

## *Branch and Bound* et fonction de coût

Pour résoudre le problème et minimiser $MC$, on peut utiliser un algorithme de type *branch and bound* [3], [5]. Un tel algorithme détermine les solutions optimales à un problème en les évaluant « toutes ». Mais comme le nombre de solutions est souvent astronomique ($C!$ dans notre cas), ce type d'algorithme limite le nombre de solutions réellement évaluées de deux façons : en évaluant les solutions les plus prometteuses en premier, puis en n'évaluant pas certains ensembles de solutions lorsqu'on peut déterminer qu'ils ne contiennent aucune solution optimale. Notons que malgré ces optimisations dans le calcul des solutions optimales, un algorithme de type *Branch and Bound* détermine les véritables solutions optimales, non des approximations.

Tout ceci est basé sur une fonction de coût qui associe à chaque solution *partielle* un coût, de façon à ce que toute solution qui complète la solution partielle ait nécessairement un coût au moins aussi élevé. L'efficience de ce type d'algorithme provient de deux qualités de la fonction de coût. D'abord, la fonction de coût doit donner une bonne évaluation du coût réel (c'est-à-dire le coût minimal des solutions qui complètent la solution partielle) parce que plus l'évaluation est près du coût réel, plus petit est le nombre de solutions partielles qui doivent être évaluées pour trouver les solutions optimales. Également, la fonction de coût doit rester suffisamment simple pour que son temps de calcul ne devienne pas prohibitif. Ces deux qualités étant diamétralement opposées, tout est question d'équilibre...

Dans notre cas, une solution partielle sera un « début de permutation » $P_1, P_2, ..., P_K$ où $0 \leq K \leq C$, qui représente les candidats aux $K$ premières positions de la solution. Lorsque $K = C$, on a évidemment une solution complète.

On peut considérer que le coût minimal d'une solution partielle est le nombre de comparaisons déjà contredites dans les $K$ premières positions et déjà contredites par le fait que les candidats non encore positionnés seront tous après la position $K$. Il s'agit clairement d'un minimum, mais il n'est pas nécessairement près du minimum réel puisque le classement des $C - K$ autres positions pourraient résulter en de nombreuses comparaisons contredites, même dans la meilleure solution.

On peut grandement améliorer la qualité de cette fonction de coût en « réduisant » la matrice $M'$. En effet, on remarque que si $c \neq d$, exactement un de $M'_{c,d}$ ou $M'_{d,c}$ est inclus dans $MC$. Par conséquent, on peut « réduire » la matrice de façon à ce qu'au moins un des deux soit nul :

$$M"_{c,d} = M'_{c,d} - min\{M'_{c,d}, M'_{d,c}\} \ .$$

La valeur $MC$ à minimiser est donc :

$$MC = \sum_{k=1}^{C} \sum_{l=1}^{k-1} \left( M"_{P_k,P_l} + min\{M'_{P_k,P_l}, M'_{P_l,P_k}\} \right)$$

$$= \sum_{k=1}^{C} \sum_{l=1}^{k-1} M"_{P_k,P_l} + \sum_{k=1}^{C} \sum_{l=1}^{k-1} min\{M'_{P_k,P_l}, M'_{P_l,P_k}\} \ .$$

Par symétrie, parce que $M_{j,c,c} = 0$ et parce que $P_1, P_2, ..., P_C$ est une permutation de $\{1, 2, ..., C\}$, on obtient :

$$MC = \sum_{k=1}^{C} \sum_{l=1}^{k-1} M"_{P_k,P_l} + \frac{\sum_{c=1}^{C} \sum_{d=1}^{C} min\{M'_{c,d}, M'_{d,c}\}}{2} \ .$$

Le deuxième terme étant une constante (une fois les évaluations des juges connues), minimiser $MC$ est équivalent à minimiser $MC'$, où

$$MC' = \sum_{k=1}^{C} \sum_{l=1}^{k-1} M"_{P_k,P_l} \ .$$

Cette simplification permettra à notre fonction de coût de fournir une évaluation beaucoup plus près du coût réel. En effet, en supposant que les évaluations des juges soient relativement cohérentes, il devient probable que compléter une solution partielle puisse se faire à coût nul ou très faible.

## Algorithme de résolution

On peut résumer l'algorithme de résolution ainsi :

1. On calcule tous les $M"_{c,d}$.
2. On fixe $MeilleuresSolutions \leftarrow \emptyset$, $MeilleurCoût \leftarrow \infty$, $K \leftarrow 0$, et $CoûtPartiel \leftarrow 0$.
3. On « essaie » chacun des candidats $c \notin \{P_1, ..., P_K\}$ à la position $K+1$, en commençant par ceux qui semblent les plus prometteurs, c'est-à-dire en ordre croissant du nombre de comparaisons contredites par le placement du candidat $c$ à la position $K + 1$ , soit $Coût_c = \sum_{d \notin \{P_1,...,P_K,c\}} M"_{d,c}$ .
   (a) Toute solution qui commence par $P_1, ..., P_K, c$ aura donc un coût d'au moins $CoûtPartiel + Coût_c$ . Si $CoûtPartiel + Coût_c > MeilleurCoût$, on ignore ce choix de candidat de même que tous les suivants qui ne peuvent qu'être plus coûteux.
   (b) Autrement, on incrémente $K \leftarrow K + 1$, on fixe $P_K \leftarrow c$ et on incrémente $CoûtPartiel \leftarrow CoûtPartiel + Coût_c$.
   (c) Si $K < C$, on réexécute récursivement l'étape 3 pour compléter la solution. Au retour, on remet $K$ et $CoûtPartiel$ à leurs valeurs précédentes avant d'essayer le prochain candidat.
   (d) Si $K = C$, on a un classement complet :

i. Si $Co\hat{u}tPartiel < MeilleurCo\hat{u}t$, on fixe $MeilleurCo\hat{u}t \leftarrow Co\hat{u}tPartiel$, et $MeilleuresSolutions \leftarrow \{(P_1, P_2, ..., P_C)\}$.

ii. Si $Co\hat{u}tPartiel = MeilleurCo\hat{u}t$, on modifie plutôt $MeilleuresSolutions \leftarrow MeilleuresSolutions \cup \{(P_1, P_2, ..., P_C)\}$.

4. Lorsque tous les « essais » ont été faits, les solutions optimales sont données par $MeilleuresSolutions$.

Cet algorithme peut être extrêmement lent : il prend un temps en $\Theta(C! + JC^2)$ dans le pire des cas. Le pire cas est le cas dégénéré où les $C!$ solutions potentielles sont toutes optimales (ce qui se produit si tous les $M"_{c,d}$ sont nuls) et il n'est pas possible de faire mieux dans ce cas puisqu'on veut obtenir toutes les solutions optimales.

Malgré ce pire cas désastreux, cet algorithme permet de trouver toutes les solutions optimales très rapidement pour les cas normaux où les évaluations des juges sont relativement cohérentes et où il y a donc un petit nombre de solutions optimales et beaucoup de très mauvaises solutions.

## Post-optimisation

L'expérience a montré qu'il y a souvent plusieurs solutions optimales, ce qui n'est pas étonnant étant donné la nature discrète des données qui servent de base à l'optimisation. Il est donc utile d'ajouter une post-optimisation qui permet d'agréger les solutions optimales en un plus petit ensemble de solutions « suroptimales » .

On veut donc choisir la solution optimale qui représente le mieux *l'ensemble* des solutions optimales. On veut, dans un certain sens, la « moyenne » des solutions optimales, tout en respectant la contrainte supplémentaire que cette « moyenne » doit elle-même être une solution optimale. Pour la déterminer, on a utilisé une minimisation de la somme des carrés des différences entre la position de chaque candidat dans une solution optimale et sa position moyenne à travers toutes les solutions optimales.

Soit $S$ l'ensemble des solutions optimales et $P_{s,k}$ le candidat à la $k^e$ position dans la solution $s$. On calcule la moyenne des positions de chaque candidat :

$$\mu_c = \frac{\sum_{s \in S}(k : P_{s,k} = c)}{\#S} \ .$$

Puis on doit trouver les solutions optimales qui minimisent

$$\acute{E}cart_s = \sum_{c=1}^{C}((k : P_{s,k} = c) - \mu_c)^2 \ .$$

Une fois les solutions optimales obtenues par l'algorithme de type *branch and bound*, un algorithme très simple, qui évalue l'écart pour chaque solution optimale, permet de déterminer les solutions « suroptimales » . Ces solutions forment le classement final recherché. Bien entendu, il pourrait y avoir plus d'une solution « suroptimale » , auquel cas on doit accepter des égalités.

## Bonnes propriétés de la méthode par comparaison

Si on se restreint aux 4 bonnes propriétés du théorème d'Arrow, la méthode par comparaison en respecte 2. Il est d'abord évident qu'elle respecte la propriété de non-dictature. Elle respecte aussi la propriété d'unanimité, ce qu'on peut prouver en remarquant que si tous les juges ont classé un candidat $A$ devant un candidat $B$, le coût de toute solution dans laquelle $B$ serait classé devant $A$ peut être diminué en échangeant simplement les positions de $A$ et de $B$. Cette propriété a cependant bien peu d'intérêt pour notre cas, puisque les juges n'ont pas tous évalué les mêmes candidats.

La méthode par comparaison ne respecte pas la propriété d'indifférence des candidats non pertinents, comme on pourrait s'y attendre puisque, au contraire, elle se fie à la transitivité supposée des comparaisons pour obtenir le classement final. Pour le montrer, considérons un cas de 2 juges qui ont chacun évalué 2 des 3 candidats et obtenu les classements $A < B$ et $B < C$. La méthode obtient comme classement final $A < B < C$. Si on modifie les classements des juges en conservant les classements relatifs de $A$ et $C$, ce qui ne nous limite pas du tout puisqu'aucun juge n'a évalué à la fois $A$ et $C$, on peut obtenir les classements $B < A$ et $C < B$ et dans ce cas le classement final devient $C < B < A$. Puisque $A$ et $C$ sont inversés, la propriété n'est pas respectée.

Un aspect plutôt contre-intuitif de la méthode par comparaison est que la propriété d'universalité n'est pas respectée. Par exemple, avec 3 candidats et 5 juges dont 3 ont obtenu $A < C$, 1 a obtenu $B < A$ et 1 a obtenu $C < B$, on obtient deux solutions : $A < C < B$ et $B < A < C$. Or, cet ensemble de permutations ne correspond à aucun classement unique, même en admettant des égalités. En pratique, nous n'avons jamais rencontré un tel cas, puisqu'il exige beaucoup d'incohérence dans les évaluations des juges.

## Nombre de juges requis

Un aspect à considérer avec la méthode par comparaison est le nombre de juges requis. Comme chaque juge évalue moins de candidats, on peut s'attendre à ce qu'il faille plus de juges pour obtenir des résultats fiables.

Il est facile de voir que chaque juge ayant évalué $N$ candidats fournit au maximum $\frac{N(N-1)}{2}$ comparaisons concernant les candidats qu'il a évalués, soit une comparaison pour chaque paire de candidats. Il s'agit d'un maximum puisqu'en cas d'égalités, le juge fournit moins de comparaisons.

On peut donc évaluer le nombre de juges requis en comparant le nombre de comparaisons à celui qui serait obtenu si une cible de $T$ juges avaient évalué tous les $C$ candidats. Si $J$ juges évaluent chacun $N$ candidats, on obtient $\frac{JN(N-1)}{2}$ comparaisons, contre $\frac{TC(C-1)}{2}$ dans le cas où $T$ juges évaluent tous les $C$ candidats. Il faut donc que

$$J \sim \frac{TC(C-1)}{N(N-1)} \sim T\left(\frac{C}{N}\right)^2 .$$

Le nombre de juges requis pour obtenir des résultats aussi fiables que si chaque juge évaluait tous les candidats croît donc quadratiquement en fonction du facteur de réduction du nombre de candidats à évaluer $\frac{C}{N}$. C'est une limitation qui n'est pas toujours des plus pratiques.

## Choix des sous-ensembles de candidats à évaluer

Avec la méthode par comparaison, il n'est pas nécessaire que chaque juge évalue tous les candidats. De plus, chaque juge peut aussi évaluer un nombre différent de candidats. Il est cependant essentiel de bien répartir les candidats à évaluer parmi les différents juges.

Par exemple, si la moitié des juges évalue la première moitié des candidats et l'autre moitié des juges évalue la deuxième moitié des candidats, on n'aura aucune comparaison entre les première et deuxième moitiés des candidats. Même si tous les juges s'entendent parfaitement dans leurs évaluations, ceci résulte en $\frac{C!}{(\frac{C}{2}!)^2}$ solutions optimales équivalentes.

Même la post-optimisation fournira $2^{(\frac{c}{2})}$ solutions suroptimales. Bref, on obtient des résultats inutilisables, sans compter qu'il faut un temps prohibitif à l'algorithme présenté ici pour les calculer...

Il faut bien sûr maximiser les recoupements entre les sous-ensembles de candidats évalués par les juges tout en maximisant le nombre de comparaisons obtenues pour chaque candidat. Bien que cet aspect serait l'occasion de définir un autre problème d'optimisation, en pratique, nous n'avons pas exploré davantage cet aspect.

## Exemple

L'exemple qui suit est tiré de l'expo-sciences d'élèves de $4^e$ année qui s'est déroulée à l'Île des Soeurs. Par le passé, un jury de 3 juges évaluaient les 12 candidats et le classement était établi à l'aide de la méthode directe. À cause de la disponibilité limitée des juges et de l'horaire de l'expo-sciences, il arrivait fréquemment que les juges n'aient pas le temps d'évaluer les 12 candidats, ce qui aurait pourtant dû être nécessaire vu que la méthode directe était utilisée. De plus, il y avait souvent une grande variabilité dans les évaluations numériques des juges, ce qui introduisait un biais, biais qu'il convenait aussi de réduire parce que les juges auraient pu connaître personnellement certains candidats qu'ils évaluaient. La nouvelle organisatrice m'a donc contacté pour modifier la méthode d'évaluation et il a donc été décidé d'utiliser la méthode par comparaison décrite dans cet article pour régler ou alléger ces problèmes.

Il a été estimé qu'il était raisonnable d'évaluer 8 candidats dans le temps prévu. Il aurait fallu 7 juges pour obtenir des résultats aussi fiables que lorsque 3 juges évaluaient les 12 candidats (196 comparaisons contre 198), mais nous avons décidé de nous contenter de 6 juges (ce qui fournit 168 comparaisons). Ceci exigeait tout de même de trouver 18 juges, puisqu'un total de 36 candidats de 3 niveaux ($4^e$, $5^e$ et $6^e$) participaient à l'expo-sciences. Il aurait été difficile de trouver plus de juges qualifiés.

À défaut d'avoir défini formellement le problème de distribution des candidats aux juges et la meilleure solution possible, nous avons opté pour une distribution ad hoc qui semblait adéquate. Chaque juge évaluait un sous-ensemble différent de candidats, et chaque candidat était évalué par le même nombre de juges.

Un candidat s'est finalement désisté avant le concours, ce qui a laissé 11 candidats. Les classements des 6 juges étaient les suivants :

- 1, 3, 8, 2, 6, 4, 5
- 8, 11, 10, 6, 7, 5, 9
- 1, 11, 3, 10, 5, 9, 4
- 8, 3, 9, 10, 7, 6, 2, 4
- 3, 6, 11, 1, 2, 10, 7
- 8, 11, 1, 7, 2, 9, 4, 5

On peut d'abord remarquer que certains juges ont évalué 8 candidats alors que d'autres n'en ont évalué que 7 à cause du désistement du candidat 12. La méthode par comparaison est parfaitement adaptée à ce cas puisqu'elle ne considère que les comparaisons entre les candidats sans égard à la taille du classement des juges.

À partir du classement de chacun des juges, on doit construire le tableau à trois dimensions $M$. Voici la matrice $M_1$ correspondant aux comparaisons fournies par le juge 1 :

$$M_1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En additionnant ces matrices pour les 6 juges, on bâtit la matrice $M'$ :

$$M' = \begin{pmatrix} 0 & 3 & 2 & 3 & 3 & 1 & 2 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 3 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 3 & 0 & 3 & 2 & 3 & 2 & 1 & 2 & 3 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 2 & 2 & 0 & 2 & 0 & 1 & 1 & 1 \\ 0 & 2 & 0 & 2 & 2 & 1 & 0 & 0 & 2 & 0 & 0 \\ 1 & 3 & 1 & 3 & 3 & 3 & 3 & 0 & 3 & 2 & 2 \\ 0 & 1 & 0 & 3 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 2 & 2 & 3 & 0 & 2 & 0 & 0 \\ 2 & 2 & 1 & 2 & 3 & 1 & 3 & 0 & 3 & 3 & 0 \end{pmatrix}$$

On doit ensuite « réduire » cette matrice pour obtenir la matrice $M$", qui sera celle utilisée par la fonction de coût :

$$M'' = \begin{pmatrix} 0 & 3 & 1 & 3 & 3 & 0 & 2 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 3 & 2 & 3 & 2 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 3 & 3 & 3 & 3 & 0 & 3 & 2 & 2 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 1 & 3 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 2 & 3 & 0 & 3 & 0 & 3 & 3 & 0 \end{pmatrix}$$

Exécuter manuellement l'algorithme complet pour optimiser la solution est bien trop ardu. Mais pour en comprendre le principe, on peut décrire la première itération. Au début, notre solution partielle est vide et on doit déterminer quel sera le candidat à classer en première position. Si on place le candidat 1 à cette position, le coût est de 1, soit la somme de la colonne 1 de $M$". L'algorithme calcule ce coût pour chacun des candidats, puis va essayer de placer à la première position chacun des candidats en commençant par les plus prometteurs : 8 (coût 0), 1 (coût 1), 3 (coût 1), 11 (coût 2), 6 (coût 7), 10 (coût 10), 2 (coût 13), 9 (coût 13), 7 (coût 14) 5 (coût 20) et 4 (coût 23). L'algorithme essaie donc le candidat 8 en première position et se réexécute récursivement pour essayer les candidats restants en $2^e$ position, puis en $3^e$ position, etc., jusqu'à ce qu'il trouve une première solution. Dans ce cas-ci, la première solution trouvée est 8, 11, 1, 3, 10, 6, 7, 2, 5, 9, 4 dont le coût est 1.

Par la suite, l'algorithme continue à chercher des solutions meilleures ou équivalentes, en essayant les autres candidats pour compléter chaque solution partielle, tout en écartant une solution partielle dès que son coût est plus élevé que la meilleure solution trouvée jusqu'à présent.

Avec les données dont nous disposons, l'algorithme identifie finalement 3 solutions optimales, dont le coût est de 1 :

- 8, 11, 1, 3, 10, 6, 7, 2, 5, 9, 4
- 8, 11, 1, 3, 10, 6, 7, 2, 9, 4, 5
- 8, 11, 1, 3, 10, 6, 7, 9, 2, 4, 5

On peut voir que seules les 4 dernières positions varient. L'ambiguïté provient de deux « problèmes » . D'abord, il n'y a aucune comparaison nette entre les candidats 2 et 9, qui sont près un de l'autre. Ensuite, il y a un cycle dans les classements : ils indiquent que 4 est meilleur que 5, que 5 est meilleur que 9 et que 9 est meilleur que 4. La présence de cycles de ce genre est bien connue et se produit aussi lorsque les juges ont évalué tous les candidats.

Le candidat 2 est en moyenne à la position $8\frac{1}{3}$, le candidat 9 en position 9 et les candidats 4 et 5 en position $10\frac{1}{3}$. On peut alors déterminer la solution suroptimale à l'aide d'une minimisation des carrés des écarts avec les positions moyennes, ce qui fournit une seule solution suroptimale qui est la solution finale :

• 8, 11, 1, 3, 10, 6, 7, 2, 9, 4, 5

Il est intéressant de noter que, malgré le cycle impliquant les candidats 4, 5 et 9 et le fait que les candidats 4 et 5 aient la même position moyenne dans les solutions optimales, aucun n'est à égalité dans la solution suroptimale. Plusieurs raisons expliquent ceci : il y a absence de comparaison entre les candidats 2 et 9 mais le candidat 2 se compare favorablement aux candidats 4 et 5 contrairement au candidat 9, ce qui place le candidat 2 en tête ; les coûts du cycle ne sont pas égaux et favorisent le candidat 9 par rapport aux candidats 4 et 5 ; finalement, puisque le candidat 4 se compare favorablement au candidat 5, il ne subsiste aucune égalité.

De façon générale, nous avons observé que l'utilisation d'une optimisation globale suivie d'une réoptimisation parmi les solutions optimales a tendance à éliminer presque toutes les égalités, sauf lorsque deux candidats sont parfaitement équivalents (c'est-à-dire que leurs entrées dans $M$" sont identiques). En pratique, dans les rares cas d'égalité auxquels nous avons été confrontés, nous avons soit ignoré l'égalité lorsqu'elle concerne des candidats qui n'occupent pas les premières positions, soit brisé l'égalité en envoyant un autre juge évaluer les candidats concernés (ce qui exige de tout recalculer et qui, en cas de malchance, pourrait créer d'autres égalités !).

Avec ces évaluations des juges, l'algorithme prend bien moins d'une seconde, sur un ordinateur peu puissant, pour produire la solution finale. On s'attend d'ailleurs à un temps d'exécution très court lorsque les évaluations des juges sont cohérentes, parce que les solutions optimales sont parmi les toutes premières solutions trouvées. La fonction de coût est si précise dans ce cas-ci que la première solution trouvée est déjà une solution optimale. La conséquence est que l'algorithme évalue seulement 7 080 solutions partielles sur près de 44 millions.

## Conclusion

Comme on l'a vu, les méthodes traditionnelles d'agrégation des évaluations par jury ne s'appliquent pas bien au contexte où tous les juges n'ont pas évalué tous les candidats. Pour ce contexte, une autre méthode d'agrégation est nécessaire.

Nous avons décrit une méthode d'agrégation utilisable dans ce contexte. La méthode prend la forme d'un problème d'optimisation qui consiste dans un premier temps à minimiser le nombre de compa-

raisons faites par les juges qui sont contredites par le classement final, et dans un deuxième temps à sélectionner parmi les solutions optimales les solutions « suroptimales » les plus représentatives de l'ensemble des solutions optimales. Un algorithme efficace pour résoudre ce problème d'optimisation a également été présenté.

Cette méthode a quelques applications pratiques et a d'ailleurs été utilisée dans le cadre de concours expo-sciences. La principale difficulté avec cette méthode est le nombre de juges requis pour obtenir des résultats fiables, qui croît quadratiquement en fonction du facteur de réduction du nombre de candidats à évaluer. Pour réduire le temps requis à chacun des juges pour faire son évaluation, il faut donc un grand bassin de juges disponibles.

### Remerciements

Je voudrais remercier France Caron qui, en me demandant de définir une méthode pour évaluer les candidats aux expo-sciences dont elle était responsable, a initié mes réflexions au sujet de l'algorithme présenté ici et qui m'a plus tard proposé d'écrire cet article. Je remercie également GIRO inc. pour son soutien à la production de cet article.

## Références

[1] Arrow, Kenneth J. (1951). *Social Choice and Individual Value*. New York : John Wiley.

[2] Borda (de), Jean-Charles. (1784). *Mémoire sur les élections au scrutin*. Histoire de l'Académie royale des sciences.

[3] Brassard, Gilles et Bratley, Paul. (1987). *Algorithmique : conception et analyse*, Montréal : Presses de l'Université de Montréal.

[4] Condorcet, Marquis Jean Antoine Caritat. (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris : L'imprimerie royale.

[5] Hillier, Frederick S. et Lieberman, Gerald J. (1986). *Introduction to Operation Research*. San Francisco : Holden-Day.

[6] Kemeny, John G. (1959). Mathematics Without Numbers, *Daedalus*, Vol. 88, No. 4, Fall, pp. 577-591.

[7] Truchon, Michel. (2002). *Choix social et comités de sélection : le cas du patinage artistique*. Consulté le 6 février 2012 sur le site du CIRANO http ://www.cirano.qc.ca/pdf/publication/2002RB-02.pdf