Problèmes de recrutement

Emmanuel Hyon, Eric Thierry

Devoir du 02/12/2002

Un grand concours national est organisé par différentes écoles. A l'issue de moult épreuves, communes ou pas, chaque école établit un classement qui lui est propre des étudiants qui ont passé le concours. D'autre part, chaque étudiant a une liste de souhaits où il classe les écoles par ordre de préférence.

Le concours terminé, il s'agit d'affecter les étudiants aux écoles. Pour effectuer le recrutement définitif, les écoles se sont mises d'accord pour procéder de manière très centralisée. La politique choisie a été de collecter les listes des classements de toutes les écoles ainsi que les listes de souhaits de tous les étudiants. Par exemple les tableaux ci-dessous représentent les informations collectées pour les étudiants a, b, c, d ayant concouru pour les écoles A, B, C, D.

A	b > a > d > c
B	a > d > c > b
C	d > a > b > c
D	a > d > b > c

a	A > B > C > D
b	D > C > A > B
c	B > A > D > C
d	C > B > A > D

Le tableau de gauche donne sur chaque ligne pour chaque école le classement des étudiants, du premier au dernier. Le tableau de droite donne sur chaque ligne pour chaque étudiant le classement des écoles en commençant par la préférée.

A partir de toutes ces listes de préférences, l'objectif est de proposer un plan d'affectation des étudiants aux écoles remplissant toutes les écoles, et le plus juste possible. On suppose qu'il y a m écoles E_1, ..., E_m et n étudiants e_1, ..., e_n. De plus, chaque école E_i peut recruter p_i étudiants au maximum. Une ***affectation*** est une application (partielle car tous les étudiants ne seront pas forcément affectés) de l'ensemble des étudiants dans l'ensemble des écoles respectant les nombres de places. Autrement dit c'est un ensemble de couples du type (e, E) où e est un étudiant affecté l'école E, avec les conditions qu'un étudiant apparait dans au plus un couple et chaque école E_i apparait dans exactement p_i couples.

Pour qu'une affectation soit considérée comme satisfaisante, on va définir la notion d'***affectation stable***. Et pour étudier le problème, on va d'abord regarder le cas particulier où chaque école n'a qu'un poste à offrir, avant de regarder le cas général.

1 Cas où chaque école ne recrute qu'un étudiant

C'est donc le cas très particulier où pour toute école E_i, le nombre de places p_i vaut 1.

1.1 Listes complètes et $m = n$.

On suppose que chaque école a classé la totalité des étudiants, et inversement chaque étudiant a classé la totalité des écoles. Les listes sont complètes. On suppose aussi $m = n$, et ici une affectation est une bijection de l'ensemble des étudiants dans l'ensemble des écoles. On dit qu'une affectation $\mathcal{A}$ est ***instable*** s'il existe un étudiant e et une école E tels que :

- $(e, E) \notin \mathcal{A}$.
- $(e, F) \in \mathcal{A}$ et $(f, E) \in \mathcal{A}$, mais e préfère E à F, et E préfère e à f.

Autrement dit les associations (e, F) et (f, E) ne sont pas satisfaisantes pour e et E. Dans ce cas, l'échange qui consiste à faire les associations (e, E) et (f, F) est appelé un ***échange améliorant***.

Si cette situation n'existe pas, l'affectation est dite ***stable***. L'objectif est de trouver des affectations stables lorsqu'elles existent.

1.1.1 Montrer déjà qu'il existe des cas où il existe plusieurs affectations stables : trouver pour n écoles E_1, ..., E_n et n étudiants e_1, ..., e_n des listes de préférences pour lesquelles il existe au moins n affectations stables.

1.1.2 Montrer qu'une succession quelconque d'échanges améliorants ne mène pas nécessairement à un couplage stable (même s'il en existe un).

Voilà un algorithme qui a été proposé pour calculer des affectations stables :

Tous les étudiants sont libres et toutes les écoles sont vides ;
Tant qu'il existe une école E vide faire
    prendre e le premier étudiant de la liste de E à qui E n'a pas encore proposé de poste ;
    si e est libre, alors affecter provisoirement e à E ;
    sinon
        si e préfère E à son école provisoire E', alors
            affecter provisoirement e à E (et E' redevient vide) ;
        sinon
            e rejette la proposition de E (qui reste vide) ;
Retourner comme affectation finale les couples qui restent.

1.1.3 Montrer que cet algorithme s'arrête toujours et que l'on obtient bien à la fin une affectation stable, ce qui prouve donc qu'il existe toujours une affectation stable dans le cas étudié.

1.1.4 Quelle est la complexité dans le pire des cas de cet algorithme ? (l'algorithme est peu détaillé, donc précisez brièvement les structures de données que vous utilisez pour l'implanter et commenter précisément la complexité, en particulier comptez le nombre maximum de propositions faites par les écoles aux étudiants).

1.2 Qui est le plus satisfait ?

On va voir que cet algorithme bien que fournissant des affectations stables comme souhaité, privilégie quand même l'un des partis.

1.2.1 L'algorithme tel qu'il est formulé peut donner lieu à plusieurs exécutions différentes. Montrer cependant que toutes les exécutions possibles de l'algorithme renvoient la même affectation stable à la fin.

1.2.2 Montrer que avec cette affectation stable, chaque école a le meilleur étudiant qu'elle puisse avoir dans toute affectation stable.

1.2.3 Par contre, montrer que avec cette affectation stable, chaque étudiant a la pire école qu'il puisse avoir dans toute affectation stable.

1.3 Listes complètes et $m < n$.

On suppose toujours que les listes de préférences sont complètes. Par contre le nombre total de places est inférieur au nombre d'étudiants, $m < n$.

Ici une affectation $\mathcal{A}$ est une application injective de l'ensemble des écoles dans l'ensemble des étudiants. L'affectation est ***instable*** s'il existe un étudiant e (affecté à F ou affecté nulle part) et une école E (où f est affecté) tels que e n'est pas affecté à E et :

- E préfère e à f et (e préfère E à F ou bien e n'est affecté nulle part).

1.3.1 Dans ce cas, est-ce qu'il existe toujours une affectation stable ? Si oui, comment en calculer une ?

1.4 Listes incomplètes et $m = n$

En réalité, les listes collectées ne sont pas forcément toutes complètes : chaque étudiant ne se présente pas à toutes les écoles, les écoles ne classent pas tous les étudiants et certains étudiants abandonnent en cours de route l'idée d'intégrer certaines écoles. On suppose donc que les listes des écoles (resp. des étudiants) classent au moins un étudiant (resp. une école) mais pas forcément tous (resp. toutes). On cherche alors une affectation stable $\mathcal{A}$ comme dans le cas des listes complètes et $m = n$ avec en plus la condition que $(e, E) \in \mathcal{A}$ si et seulement si E est dans la liste de e et inversement.

Les tableaux ci-dessous présentent un exemple de listes incomplètes.

A	b > a
B	a > d > c > b
C	d > a > c
D	a > d > b > c

a	A
b	D > A > B
c	B > A > D > C
d	B > A > D

1.4.1 Montrer qu'il y a maintenant des cas pour lesquels il existe une affectation mais aucune affectation stable.

1.4.2 Montrer que l'on peut ajouter un étudiant fictif e et une école fictive E (avec des listes de préférences complètes que l'on précisera) et compléter les listes des autres tels que le théorème suivant soit vérifié :

Théorème. *Il existe une affectation stable pour le système à listes complètes dont un couple est (e, E) si et seulement si il existe une affectation stable pour* *le système initial à listes incomplètes.*

1.4.3 Montrer aussi, dans le cas des listes *complètes*, **que s'il existe une affectation stable contenant le couple (e, E) dans lequel E est le dernier choix de e et e est le dernier choix de E, alors toutes les affectations stables contiennent le couple (e, E).**

1.4.4 Par conséquent, comment résoudre le cas des listes incomplètes ? Avec quelle complexité ?

1.5 Réputation des écoles

Un autre facteur intervient : la réputation des écoles implique que les listes de préférence des étudiants sont presque identiques. Pour simplifier, on suppose qu'il existe un classement $E_1 > \ldots > E_m$ de toutes les écoles qui est reconnu par tout le monde. On suppose que $m = n$.

On se place dans le cas des listes complètes, tous les étudiants ont donc la même liste de préférence $E_1 > \ldots > E_m$.

1.5.1 Dans ce cas particulier, montrer qu'il existe une unique affectation stable.

1.5.2 Quelle est la complexité pour la calculer ? (donnez le meilleur algorithme que vous pouvez)

2 Cas général

On revient maintenant au cas général décrit en introduction où les nombres p_i de places offertes sont quelconques. On suppose qu'il y a moins de places que d'étudiants, c'est à dire $p_1 + \ldots + p_m \leq n$. Une affectation, définie comme dans l'introduction, est dite ***instable*** pour les mêmes critères que dans la partie 1.3.

2.1.1 Répondre aux questions suivantes (en justifiant bien vos réponses) : est-ce qu'il existe toujours une affectation stable ? comment calculer une affectation stable ? avec quelle complexité ?

3 Casse-tête au service hébergement

En épilogue, l'école Y recrute $2n$ étudiants qu'elle se charge de loger dans sa résidence à raison de deux étudiants par chambre. Après une grande fête d'intégration qui a permis à tous les étudiants de se rencontrer, l'école demande à chaque étudiant de fournir une liste par ordre de préférence des étudiants avec qui il souhaite partager sa chambre. L'école réclame des *listes complètes*, c'est à dire où tous les étudiants sont classés.

Muni de toutes ces listes et pour maintenir une bonne ambiance dans la résidence, le service hébergement cherche une ***partition stable*** des étudiants. Une ***partition*** est une partition des $2n$ étudiants en n paires de deux étudiants. Une partition est ***instable*** s'il existe deux paires $\{a, b\}$ et $\{c, d\}$ telles que a préfère d à b et d préfère a à c.

3.1.1 Montrer qu'il existe des cas pour lesquels on ne peut trouver aucune partition stable des étudiants.

En fait, le problème d'affectation stable des parties précédentes est plus facile que le problème de la partition stable.

3.1.2 Montrer que pour toute instance du problème d'affectation de n étudiants dans n écoles avec listes complètes, on peut construire une instance du problème de partition stable avec $2n$ personnes tel que les partitions stables correspondent exactement aux affectations stables de l'instance initiale.

Pour note, il existe un algorithme en $\mathcal{O}(n^2)$ qui permet à partir des listes complètes des $2n$ étudiants de déterminer s'il existe une partition stable et d'en calculer une si oui. Mais c'est un peu plus compliqué ...