Problèmes de recrutement

Jean-Baptiste Rouquier, Jean-François Pineau

D'après un DM par Emmanuel Hyon et Éric Thierry

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. Réciproquement, 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 centralisée. Ainsi un organisme indépendant collecte les les listes des classements des écoles ainsi que les listes de souhaits des candidats. 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.

définitions

- On numérote les écoles $E_1 \dots E_m$ et les étudiants $e_1 \dots e_n$.
- On note p_i le nombre de place de E_i.
- Une ***affectation*** est une fonction de l'ensemble des étudiants dans l'ensemble des écoles. Cette fonction n'est pas toujours définie car certains étudiants n'ont pas d'école ou préfèrent redoubler. Elle doit respecter le nombre de places de chaque école.
  Autrement dit, c'est un ensemble de couples (e, E) où e est un étudiant affecté l'école E, tel que chaque étudiant apparaisse dans au plus un couple, et chaque école E_i apparaisse dans au plus 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 qu'il y a autant d'étudiants que d'écoles : $m = n$. Une affectation est donc ici une bijection entre l'ensemble des étudiants et l'ensemble des écoles.

Affectation stable Une affectation $\mathcal{A}$ est dite ***instable*** s'il existe un étudiant e et une école E tels que :
- $(e, E) \notin \mathcal{A}$. ($e$ n'est pas affecté à E)
- $(e, F) \in \mathcal{A}$ et $(f, E) \in \mathcal{A}$
- $e : E > F$ et $E : e > f$ (e préfère E à F et E préfère e à f).

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

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).

Algorithme 1 : proposition pour calculer des affectations stables

```
Tous les étudiants sont libres et toutes les écoles sont vides;
while il existe une école E vide do
    soit e le meilleur étudiant de la liste de E à qui E n'a pas encore
    proposé de poste
    if e est libre then
        affecter provisoirement e à E
    else
        if e préfère E à son école provisoire E'
        then affecter provisoirement e à E (et E' redevient vide)
        else e rejette la proposition de E (qui reste vide)
return, en tant qu'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 cas de cet algorithme ?

L'algorithme est peu détaillé, donc précisez brièvement les structures de données que vous utilisez pour l'implanter et commentez 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 l'une des parties.

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

1.2.3 Montrer qu'en revanche chaque étudiant a la pire école qu'il puisse avoir dans toute affectation stable.