Le problème d'affectation sur concours des étudiants

Jean-Baptiste Rouquier

Corrigé

1 Cas où chaque école n'a qu'une place

1.1 Listes complètes et $m = n$

1.1.1 Une affectation stable n'est pas toujours unique.

Considérons en effet les listes de la table 1. Soit $0 \leq k \leq n-1$, alors

E_1	$e_2 > e_3 > e_4 > \cdots > e_n > e_1$
$\vdots$	
E_j	$e_{j+1} > \cdots > e_n > e_1 > \cdots > e_j$
$\vdots$	
E_n	$e_1 > e_2 > \cdots > e_n$
e_1	$E_1 > E_2 > \cdots > E_{n-1} > E_n$
$\vdots$	
e_i	$E_i > \cdots > E_n > E_1 > \cdots > E_{i-1}$
$\vdots$	
e_n	$E_n > E_1 > E_2 > \cdots > E_{n-1}$

TAB. 1 – listes pour lesquelles il existe plusieurs affectations stables

l'affectation $(e_i, E_{i+k \bmod n}), 1 \leq i \leq n$ est stable (avec $\bmod n$ défini sur $1..n$). e_i accepterait d'aller dans les écoles E_i à $E_{i+k-1 \bmod n}$ à la place de celle où il est (en l'occurrence, $E_{i+k \bmod n}$). Si E_j fait partie de ces écoles, E_j s'écrit $E_{i+l \bmod n}, 0 \leq l \leq k-1$. E_j a reçu $e_{j-k \bmod n}$ et accepterait n'importe quel étudiant à la place de celui qu'elle a reçu, sauf les étudiants $e_{j-k+1 \bmod n}$ à e_j. Ces étudiants refusés sont $e_{i+l-k+1 \bmod n}$ à $e_{i+l \bmod n}$, parmi lesquels figure en particulier e_i. Il n'y a donc pas d'échange améliorant.

1.1.2 Une succession d'échanges améliorants ne mène pas toujours à une affectation stable.

Considérons en effet les listes de la table 2 et commençons avec l'affectation $\{(a, A), (b, B), (c, C)\}$. a préfère B qui l'accepte, effectuons l'échange et nous obtenons l'affectation $\{(b, A), (a, B), (c, C)\}$. Mais de même, (c, B) est alors un

A	$a > c > b$
B	$c > a > b$
C	$c > a > b$

a	$B > A > C$
b	$C > A > B$
c	$A > B > C$

TAB. 2 – exemple de listes de préférences données par les étudiants et de classements donnés par les écoles

échange améliorant, qui aboutit à $\{(b,A),(c,B),(a,C)\}$. On continue ainsi et l'on revient à la position de départ, il existe donc une suite infinie d'échanges améliorants. Ceci est résumé dans la table 3.

affectation	échange améliorant	justification
$\{(a,A),(b,B),(c,C)\}$	(a,B)	$a : B > A$ et $B : a > b$
$\{(b,A),(a,B),(c,C)\}$	(c,B)	$c : B > C$ et $B : c > a$
$\{(b,A),(c,B),(a,C)\}$	(c,A)	$c : A > B$ et $A : c > b$
$\{(c,A),(b,B),(a,C)\}$	(a,A)	$a : A > C$ et $A : a > c$
$\{(a,A),(b,B),(c,C)\}$	$\dots$	$\dots$

TAB. 3 – une suite infinie d'échanges améliorants. Chaque échange aboutit à l'affectation de la ligne suivante.

1.1.3 Un algorithme pour calculer des affectations stables

Terminaison Une école ne propose jamais sa place deux fois à un étudiant. Or le nombre de places est fini.

Montrons également que chaque école se voit attribuer un étudiant par l'algorithme. Un étudiant ne quitte une école que s'il est immédiatement dans une autre école. Un étudiant libre n'a donc reçu aucune proposition.

S'il existe une école vide E à la fin de l'algorithme (supposé par l'absurde), c'est qu'elle est allée jusqu'au bout de sa liste de préférence (sinon l'alorithme continue). Puisque $m \leqslant n$, il y a aussi un étudiant libre e. e n'a reçu aucune proposition, en particulier pas de la part de E, donc E n'est pas allée jusqu'au bout de sa liste de préférence. Contradiction.

Correction Soit E une école, e l'étudiant qui lui a été attribué par l'algorithme. Soit f un étudiant mieux classé que e dans l'école E. Soit F l'école où f est finalement affecté. Alors f préfère F à E, donc il n'existe pas d'échange améliorant dans l'affectation renvoyée par l'algorithme.

En effet, f ne peut quitter une école que pour une école mieux classée dans ses préférences. En particulier lorsque E l'a appelé (car E a appelé les étudiants jusqu'à e), f a soit accepté (pour accepter plus tard la proposition de F), soit était déjà dans une école qu'il préférait à E. Donc f préfère bien F à E.

Nous avons ainsi prouvé (constructivement) qu'il existe toujours une affectation stable dans ce cas particulier.

Complexité Précisons les structures de données utilisées (dans le style d'une interface pour OCaml) :

```
type ecole = {
  classement : int array;
  mutable index : int
    (*le premier étudiant de [classement]
      à qui l'école n'a pas encore proposé de poste. *)}

type etudiant = {
  mutable affectation : int option;
    (*l'école où il est affecté. [None] au début.*)
  preferences : int array
    (*preferences.(i) est le rang de l'école i sur la liste de
      cet étudiant. Ainsi l'on peut savoir en temps constant
      s'il préfère l'école i à l'école j.*)}

val ecoles : ecole array (*le tableau contenant les écoles*)
val etudiants : etudiant array (*besoin d'un commentaire ?*)

val ecoles_vides : int Queue.t
  (*la pile LIFO des écoles vides. FIFO conviendrait aussi.*)
```

Ceci permet de faire un passage dans la boucle en temps constant, c'est donc ce que l'on prendra comme opération élémentaire : une proposition d'une école à un étudiant.

Nous avons déjà vu (en montrant la terminaison) que l'algorithme effectuait au plus $O(n^2)$ opérations élémentaires. La table 4 (où les listes sont identiques) nous montre un cas où il faut $O(n^2)$ opérations élémentaires. En effet la seule affectation stable est $\{(e_i, E_i), \quad 1 \leq i \leq n\}$. Comme chaque école appelle les étudiants dans l'ordre de sa liste, il faut $\frac{n(n+1)}{2}$ étapes.

e_i	$E_1 < \cdots < E_n$

E_i	$e_1 < \cdots < e_n$

TAB. 4 – une borne inférieure pour la complexité de l'algorithme

1.2 Qui est le plus satisfait ?

1.2.1 La question suivante montre que, quelle que soit l'implémentation de l'algorithme, on obtient la même affectation stable à la fin.

La phrase « Chaque école obtient le meilleur étudiant possible parmi l'ensemble des affectations stables. » définit un unique étudiant pour chaque école. L'algorithme a montré que c'est une affectation.

1.2.2 Chaque école obtient par cet algorithme le meilleur étudiant qu'elle puisse avoir dans toute affectation stable.

À un instant dans le déroulement de l'algorithme, notons t le nombre d'étapes déjà effectuées (c'est à dire le nombre de passages dans la boucle). Montrons par induction sur t : *si e refuse ou quitte E alors il n'existe pas d'affectation* *stable contenant* (e, E). Ceci prouvera le résultat annoncé : une école n'appelle un étudiant que si elle ne peut recevoir aucun des précédents sur sa liste.

Initialisation Il n'y a pas de refus lors de la première étape.

Hérédité Supposons par l'absurde qu'il existe $\mathcal{A}$ une affectation stable contenant (e, E). Si e refuse E à l'étape t, c'est qu'il est dans une école F qu'il préfère. Soit f l'étudiant affecté à F dans $\mathcal{A}$. On a ainsi $\{(e, E), (f, F)\} \subseteq \mathcal{A}$.

- Si f est mieux classé que e dans F, c'est que F a appelé f qui l'a refusée ou quittée (puisque F a ensuite appelé e qui a accepté). Et ce à une étape antérieure à t, l'hypothèse de récurrence nous permet de conclure que $\mathcal{A}$ (qui contient (f, F)) n'est pas stable : contradiction.
- Si au contraire f est moins bien classé que e dans F, alors (e, F) est un échange améliorant : contradiction.

1.2.3 Inversement, chaque étudiant a la pire école qu'il puisse avoir dans toute affectation stable.

Soit e un étudiant et E l'école que l'algorithme lui affecte. Soit F une école moins bien classée selon e. Montrons qu'une affectation $\mathcal{A}$ contenant (e, F) n'est pas stable.

Soit f l'étudiant affecté à E dans $\mathcal{A}$. En résumé, $\{(e, F), (f, E)\} \subseteq \mathcal{A}$.

- Si E préfère e à f, (e, E) est un échange améliorant.
- Sinon, rappelons que d'après la question précédente, l'algorithme à donné à E le meilleur candidat qu'elle puisse avoir dans une affectation stable. Il n'existe donc pas d'affectation stable donnant à E un candidat qu'elle préfère à e. En particulier, E préfère f à e donc il n'existe pas d'affectation stable contenant (f, E). $\mathcal{A}$ n'est pas stable.

Autre solution Soient en effet e un étudiant, E l'école qui lui est donnée par l'algorithme, F une école moins bien classée que E sur sa liste, $\mathcal{A}$ une affectation stable contenant (e, F). On veut montrer que $\mathcal{A}$ ne peut pas exister. Soit f l'étudiant affecté à E dans $\mathcal{A}$. Résumons : $\{(e, F), (f, E)\} \subseteq \mathcal{A}$.

- Si f est moins bien classé que e dans E, alors (e, E) est un échange améliorant : contradiction.
- Si f est mieux classé que e dans E, c'est que E a appelé f qui l'a refusée ou quittée (puisque E a ensuite appelé e qui a accepté). La question précédente a montré qu'alors il n'existe par d'affectation stable contenant (f, E) : contradiction.