Quadrature n° 81 (2011) 16-22

De la maternelle à l'école d'ingénieur

Applications des mariages stables

par Loïc FÉVRIER*

Résumé.

La vie de tous les jours étant une grande source d'inspiration algorithmique, nous verrons dans quelle mesure l'observation d'une classe d'enfants turbulents nous amènera à expliquer comment les étudiants de classe préparatoire sont affectés aux écoles d'ingénieurs de manière optimale... pour les écoles.

I Mise en situation

Imaginez une classe d'enfants se déplaçant dans la rue et que l'institutrice ne réussit pas à faire se tenir la main deux par deux : Aïcha ne veut pas être avec Chen et refuse de lui tenir la main, Clara est avec Arthur mais préférerait être avec Battista qui, lui, est avec Betty... Résultat : tout le monde court à droite et à gauche ! Voyons comment nous pouvons aider cette pauvre institutrice.

On pourrait demander aux enfants avec qui ils accepteraient d'être, puis chercher un appariement respectant ces contraintes. Il s'agit d'un problème bien connu mais qui dans ce cas pourrait poser quelques soucis. Par exemple, imaginons que tous les garçons n'acceptent d'être qu'avec Clara, jamais on ne pourrait trouver un appariement. On ne peut pas non plus gérer la notion de préférences, on risquerait de voir les enfants se lâcher la main pour changer de binôme.

Le fait d'accepter ou pas n'étant donc pas suffisant, on pourrait demander aux enfants d'indiquer à quel point ils aiment leurs camarades du sexe opposé (par exemple une note entre 1 et 100). À chaque paire garçon-fille possible, on peut donc associer un score et l'objectif serait de trouver l'appariement qui maximiserait la somme des scores de ses paires, c'est-à-dire le « bonheur global ». À nouveau, des algorithmes sont connus pour résoudre ce problème mais il s'agit d'une optimisation globale qui va donc peut-être défavoriser certains enfants pour le bien de tous, chose bien difficile à comprendre pour eux. À nouveau, l'appariement ne sera peut-être pas très stable et les enfants vont se lâcher la main.

La notion centrale ici est cette question de stabilité : si Battista est avec Betty et Arthur avec Clara, mais que Battista et Clara préféreraient tous les deux être ensemble plutôt qu'avec leurs binômes actuels, alors ils vont vouloir se donner la main, et donc lâcher les mains de Betty et d'Arthur. On ne veut pas qu'une telle situation se produise et on va donc chercher à trouver un appariement stable, c'est-à-dire que jamais deux enfants préféreront être ensemble plutôt qu'avec leurs binômes actuels.

Il s'agit du problème dit des *mariages stables* dont nous allons expliquer une solution possible et dont nous verrons le lien avec l'affectation des étudiants de classes préparatoires dans les écoles d'ingénieurs.

Pour les lecteurs intéressés souhaitant se renseigner sur les différents types d'appariements dont nous avons parlé, le premier (accepter ou pas les autres enfants) s'appelle un *couplage maximum* et le second (donner une note) est un problème d'affectation qui peut se résoudre par exemple avec *l'algorithme Hongrois*.

* loic.fevrier@ens.fr

II Un algorithme pour le problème des mariages stables

On suppose donc qu'il y a autant de garçons que de filles et que chacun a classé ses camarades du sexe opposé par ordre de préférence. Notre objectif est alors de trouver un appariement stable : il ne doit pas exister de binôme instable, c'est-à-dire un garçon et une fille préférant être ensemble plutôt qu'avec leur binôme actuel.

Arthur	Clara	>	Betty	> Aïcha
Battista	Clara	>	Betty	> Aïcha
Chen	Betty	>	Clara	> Aïcha
Aïcha	Arthur	>	Battista	> Chen
Betty	Battista	>	Chen	> Arthur
Clara	Chen	>	Battista	> Arthur

Figure 1. Listes de préférences des enfants.

On peut voir sur la figure 1 un exemple de listes de préférences et sur la figure 2 l'ensemble des appariements possibles, les étiquettes des flèches représentant les binômes instables (Garçon-Fille) permettant de passer d'un appariement à l'autre : en empruntant une flèche, on casse deux binômes, le binôme instable est formé et les deux enfants restants forment un autre binôme. Par exemple, la situation instable dont nous avons parlé précédemment correspond au passage entre l'appariement en bas à gauche et celui au milieu à gauche.

On peut remarquer sur ce graphe qu'il existe deux appariements stables, ceux dont ne part aucune flèche, il n'y a donc pas unicité de la solution.

Nous conseillons au lecteur de chercher par lui-même un algorithme pour le problème des mariages stables avant de passer à la suite de cet article.

II.1 Algorithme

Une première idée d'algorithme consisterait à prendre un appariement quelconque puis à éliminer un binôme instable et continuer ainsi jusqu'à arriver sur un appariement stable. Mais, si on regarde le graphe des appariements, on voit qu'il est possible que l'algorithme ne s'arrête jamais pas car il y a un cycle dans ce graphe.

Algorithme de Gale-Shapley (voir [1])

Plutôt que de chercher à stabiliser un appariement instable, on va faire grossir au fur et à mesure un appariement partiel (tous les enfants ne sont pas en binôme) en garantissant qu'à chaque étape, il n'existe pas de binôme instable.

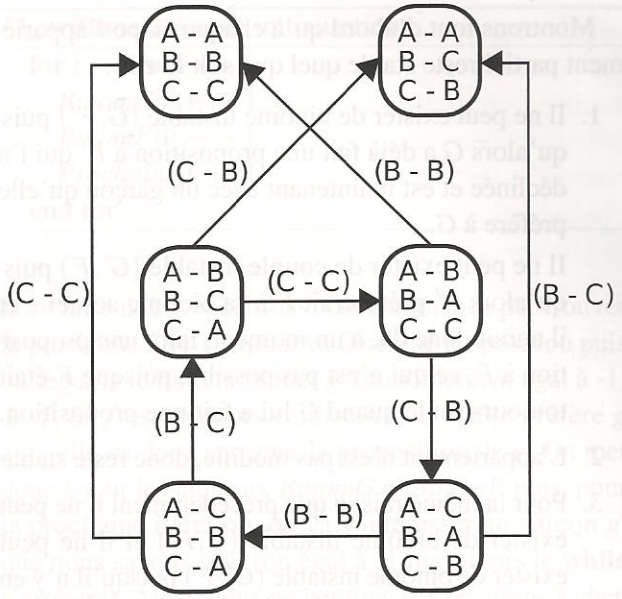

Figure 2. Graphe des appariements.

On commence donc par l'appariement vide (aucun binôme) qui est donc stable puis, tant que cela est possible, on choisit un garçon seul G et celui-ci va faire une proposition de mise en binôme à la fille F qu'il préfère (parmi celles auxquelles il n'a jamais fait de demande auparavant). Trois cas peuvent se présenter :

1. F est encore seule : elle accepte cette proposition et nous ajoutons le binôme (G,F) à l'appariement.
2. F est déjà en binôme avec un garçon qu'elle préfère à G : elle va refuser cette nouvelle proposition.
3. F est déjà en binôme mais elle préfère G à son binôme actuel G' : elle va accepter cette nouvelle proposition, nous enlevons le binôme (G',F) de l'appariement et nous ajoutons le binôme (G,F).

Notons que si G (qui fait ses propositions dans l'ordre de sa liste) est seul, alors il existe nécessairement une fille à qui il n'a pas encore fait de proposition. En effet, toute fille reste en binôme après avoir reçu sa première proposition, donc si G a déjà fait une proposition à toutes les filles, elles sont toutes en binômes et G ne peut être seul.

Exemple 1. *En reprenant l'exemple de la figure 1, nous allons donc avoir cette suite de propositions :*

- *Battista propose à Clara qui accepte.*
- *Arthur propose à Clara qui refuse car elle préfère Battista.*
- *Arthur propose à Betty.*
- *Chen propose à Betty qui accepte et rejette donc Arthur car elle préfère Chen.*
- *Arthur propose à Aïcha qui accepte.*

Montrons tout d'abord qu'à chaque étape l'appariement partiel reste stable quel que soit le cas :

1. Il ne peut exister de binôme instable (G,F') puisqu'alors G a déjà fait une proposition à F' qui l'a déclinée et est maintenant avec un garçon qu'elle préfère à G.
   Il ne peut exister de couple instable (G',F) puisqu'alors G' préférerait F à sa binôme actuelle et il aurait donc dû, à un moment, faire une proposition à F, ce qui n'est pas possible puisque F était toujours seule quand G lui a fait une proposition.
2. L'appariement n'est pas modifié, donc reste stable.
3. Pour la même raison que précédemment il ne peut exister de binôme instable (G,F') et il ne peut exister de binôme instable (G',F) puisqu'il n'y en avait pas dans l'appariement avant modification.

Montrons maintenant la terminaison de l'algorithme, c'est-à-dire que, quelles que soient les listes de préférences, l'algorithme s'arrêtera après un temps fini : à chaque étape une proposition est faite et, comme un garçon ne fera jamais deux propositions à la même fille, le nombre total de propositions faites est un nombre fini et l'algorithme termine.

En conclusion, l'algorithme nous retournera toujours un appariement stable et nous avons au passage prouvé qu'il existait toujours un tel appariement, quelles que soient les listes de préférences.

Optimalité pour les garçons Nous avons vu, sur l'exemple, qu'il pouvait exister plusieurs appariements stables et l'algorithme de Gale-Shapley nous donne un appariement particulier. Nous allons maintenant étudier ses propriétés.

Définition 2. *Un **binôme potentiel** pour un garçon G (respectivement une fille F) est une fille F (respectivement un garçon G) tel qu'il existe un appariement stable dans lequel G et F forment un binôme.*

Définition 3. *Un appariement est dit **garçon-optimum** (respectivement fille-optimum) si et seulement si chaque garçon (respectivement chaque fille) est en binôme avec son binôme le mieux classé parmi tous ses binômes potentiels.*

Définition 4. *Un appariement est dit **garçon-pessimum** (respectivement fille-pessimum) si et seulement si chaque garçon (respectivement chaque fille) est en binôme avec son binôme le moins bien classé parmi tous ses binômes potentiels.*

Théorème 5. *L'appariement retourné par l'algorithme de Gale-Shapley est garçon-optimum et fille-pessimum.*

Lemme 6. *Si F rejette la proposition de G dans l'algorithme de Gale-Shapley, alors F n'est pas un binôme potentiel pour G.*

Démonstration. Supposons que, parmi tous les refus de proposition impliquant deux enfants qui sont de potentiels binômes, le rejet de la proposition de G par F soit le premier à avoir lieu. Soit $\mathscr{A}$ l'appariement stable dans lequel G et F sont binômes. Si F a rejeté G, c'est parce qu'elle avait reçu ou a reçu plus tard une proposition de G' qu'elle préfère à G. Donc, puisque $\mathscr{A}$ est stable, G' préfère son binôme F' dans $\mathscr{A}$ à F. Mais pour que F ait reçu une proposition de G', cela signifie que celui-ci avait déjà essuyé un refus de la part de F', ce qui contredit notre hypothèse. □

Lemme 7. *Si G fait une proposition à F dans l'algorithme de Gale-Shapley, alors*
▷ G ne peut pas avoir un binôme potentiel mieux classé que F,
▷ F ne peut pas avoir un binôme potentiel moins bien classé que G.

Démonstration. ▷ Si G fait une proposition à F, c'est qu'il a essuyé un refus de la part de toutes les filles mieux classées dans ses préférences, donc, d'après le lemme précédent, aucune fille mieux classée ne peut être un binôme potentiel.
▷ Si F et G' sont binômes potentiels dans l'appariement $\mathscr{A}$ et que F préfère G à G', alors, d'après le premier point, G préfère F à son propre binôme dans $\mathscr{A}$ et $\mathscr{A}$ n'est pas stable. Contradiction. □

Le théorème découle naturellement du lemme précédent et des définitions.

L'algorithme est donc fondamentalement sexiste et en choisissant qui parmi les garçons ou les filles fera les propositions, on pourra favoriser les uns ou les autres. En particulier dans l'exemple considéré précédemment, l'appariement garçon-optimum est celui en haut à droite et celui fille-optimum celui en haut à gauche.

Listes partielles Lorsque le nombre d'enfants est important, les listes de préférences peuvent être longues et, pour des questions de temps, on souhaiterait limiter la taille de celles-ci, par exemple à 10. Dans ce cas, on peut uniquement garantir que l'algorithme donnera un appariement partiel stable : si tous les garçons ont la même liste, certaines filles ne seront choisies par personne, donc ne seront jamais dans un binôme. Cependant, s'il existe un appariement stable, alors l'algorithme le trouvera. En effet, si un garçon est arrivé à la fin de sa liste, alors, d'après notre premier lemme il n'a pas de binôme potentiel et donc il n'existe aucun appariement stable.

II.2 Implémentation et complexité

À partir de la description de l'algorithme que nous avons donnée, un certain nombre de choix doivent être faits pour arriver à une implémentation, plus ou moins efficace, de cet algorithme. Par exemple, comment va-t-on choisir un garçon seul : en testant à chaque fois chacun des garçons jusqu'à en trouver un qui est seul ? Comment déterminer quel garçon, parmi les deux possibles, une fille préfère ? En cherchant à chaque fois la position de ces garçons dans la liste de la fille ? Ces deux solutions ne seraient pas très efficaces en terme de temps de calcul et cela pourrait augmenter dramatiquement le temps d'exécution de l'algorithme. Nous allons donc expliquer les choix que nous avons faits, puis nous analyserons la complexité de l'implémentation proposée.

Arguments L'algorithme aura comme arguments N le nombre de garçons (et de filles), ceux-ci étant numérotés de 1 à N, ainsi que $PrefsG$ et $PrefsF$ qui sont les listes de préférences des garçons et des filles, $PrefsG[g][i]$ étant la $i^{\text{ème}}$ fille dans la liste de préférence de g et $PrefsF[f][i]$ étant le $i^{\text{ème}}$ garçon dans la liste de préférence de f.

Choix d'implémentation Afin de pouvoir traiter efficacement chacun des trois cas possibles de l'algorithme, nous avons besoin de savoir rapidement le classement relatif des garçons dans les listes de préférences des filles et nous allons donc utiliser un tableau *Rang*, tel que $Rang[f][g]$ nous donne le rang de g dans la liste de préférences de f, tableau qui sera rempli ainsi :

Algorithme 1 Initialisation du tableau *Rang*.

```
for f = 1 to N do
  for i = 1 to N do
    Rang[f][ PrefsF[f][i] ] ← i
  end for
end for
```

Afin de pouvoir rapidement ajouter ou supprimer un binôme, on utilise deux tableaux *BinomG* et *BinomF*, $BinomG[g]$ étant le binôme actuel de g et $BinomF[f]$ étant le binôme actuel de f. On utilisera également un tableau *Prochaine* indiquant pour chaque garçon le rang de la prochaine fille à laquelle il peut encore faire une proposition. Ces trois tableaux sont initialisés ainsi :

La valeur -1 signifie que le garçon ou la fille n'a pas de binôme pour le moment.

Algorithme 2 Autres initialisations.

```
for i = 1 to N do
  BinomG[i] ← −1
  BinomF[i] ← −1
  Prochaine[i] ← 1
end for
```

Algorithme On regarde donc chaque garçon l'un après l'autre et, s'il est toujours célibataire, on va lui faire faire des propositions. On commence par trouver la prochaine fille f à qui il va faire une proposition puis on regarde le binôme actuel g' (possiblement égal à -1) de f. Si f est toujours célibataire ou si elle préfère g à g', elle va donc accepter la proposition de g. On met donc à jour les tableaux *BinomG* et *BinomF* puis, pour la prochaine itération, on va s'intéresser au garçon g' que nous savons seul (ou égal à -1 mais alors le **while** s'arrêtera). Ainsi, nous ne perdons pas de temps à chercher un garçon seul et donc, quand le **while** termine, tous les garçons entre 1 et i (et uniquement eux) ont un binôme.

Algorithme 3 Calcul du mariage stable.

```
for i = 1 to N do
  g ← i
  if BinomG[g] ≠ −1 then
    Passer au prochain i
  end if
  while g ≠ −1 do
    f ← PrefsG[g][ Prochaine[g] ]
    Prochaine[g] ← Prochaine[g] + 1
    g' ← BinomF[f]
    if g' < 0 or Rang[f][g] < Rang[f][g'] then
      BinomG[g] ← f
      BinomF[f] ← g
      g ← g'
    end if
  end while
end for
```

Complexité Soit n le nombre de garçons (et de filles). On peut distinguer deux phases dans l'algorithme, l'initialisation des différents tableaux dont la complexité est $O(n^2)$ due au remplissage du tableau *Rang* et la phase des propositions dont la complexité dépend du nombre de propositions faites, chaque proposition prenant un temps constant.

A chaque itération de l'algorithme, un des garçons éliminera une fille de sa liste et fera une proposition. Donc chaque liste étant de longueur n, il y aura au plus n^2 propositions faites et donc la complexité totale de l'algorithme est $O(n^2)$.

Nous avons ici une version centralisée de l'algorithme, c'est-à-dire que l'institutrice a collecté les

préférences de chacun et va effectuer l'appariement. On pourrait imaginer une situation où chaque enfant conserverait sa liste de préférences privée, et donc même s'il leur sera relativement rapide de construire leur liste de préférences et les tableaux indiqués, chaque proposition, bien que faite en temps constant, va prendre un certain temps puisque le garçon devra se déplacer jusqu'à la fille pour faire sa proposition.

Il serait donc intéressant d'avoir une idée plus précise du nombre de propositions nécessaires en moyenne pour que l'algorithme termine, nombre qui, comme nous le verrons, est environ égal à $n \log n$.

II.3 Nombre moyen de propositions faites

Pour calculer le nombre moyen de propositions faites, nous allons analyser l'algorithme de manière probabiliste en faisant l'hypothèse que les listes des garçons sont choisies de manière uniforme et indépendantes parmi toutes les listes possibles, les listes des filles étant, elles, quelconques mais choisies à l'avance.

Afin de simplifier l'analyse de l'algorithme, nous allons présenter une variante (dite « amnésique ») dans laquelle il est possible à un garçon de faire plusieurs fois une proposition à la même fille, y compris à celles qui ont déjà refusé une précédente proposition : à chaque fois qu'il doit faire une nouvelle proposition, il choisit une fille aléatoirement parmi les n possibles. Ainsi, les listes de propositions des garçons sont construites au fur et à mesure que l'algorithme progresse et ces listes peuvent contenir des répétitions. Pour calculer à partir de celles-ci les listes de préférences de l'algorithme classique, on ne gardera simplement que la première occurrence de chaque fille dans chaque liste.

Soit T_P et T_A le nombre de propositions faites dans l'algorithme classique et dans la version amnésique. Alors, pour tout $m \in \mathbb{N}$,

$$\mathbb{P}[T_P > m] \leqslant \mathbb{P}[T_A > m]$$

puisque, dans la version classique, un garçon ne fera jamais plusieurs propositions à la même fille.

Théorème 8.

$$\mathbb{E}[T_A] = n\ln n + n\gamma + O(1)$$

et

$$\lim_{n\to\infty} \mathbb{P}[T_A > n\ln n + cn] = 1 - e^{-e^{-c}}$$

avec γ la constante d'Euler et $c \in \mathbb{R}$.

Ainsi, même si, dans le pire cas, n^2 propositions peuvent être faites dans l'algorithme du mariage stable, au plus $n\ln n$ le seront en moyenne et on s'écarte peu de cette moyenne.

Formalisons cette modélisation. Soit X une variable aléatoire définie comme le nombre de propositions nécessaires pour que chaque fille soit en binôme, et $C_1, C_2, \ldots, C_X$ la suite de ces propositions où $C_i \in \{1, 2, \ldots, n\}$ est le numéro de la fille à qui la $i^{\text{ème}}$ proposition a été faite.

On dit qu'une proposition est un succès lorsqu'elle est faite à une fille encore seule (en particulier C_1 et C_X seront toujours des succès), et on divise maintenant la suite des propositions en phases, la $i^{\text{ème}}$ phase commençant juste après le $i^{\text{ème}}$ succès et finissant avec le $(i+1)^{\text{ème}}$ succès.

Pour tout $i \in \{0, 1, \ldots, n-1\}$, on définit la variable aléatoire X_i comme le nombre de propositions faites pendant la $i^{\text{ème}}$ phase.

Espérance de X Soit P_i la probabilité de succès d'une proposition de la $i^{\text{ème}}$ phase, c'est-à-dire la probabilité de faire une proposition à une fille seule sachant qu'il y a déjà i filles en binôme. Puisque

$$P_i = \frac{n-i}{n}$$

et puisque la variable X_i est distribuée géométriquement alors, par définition

$$\mathbb{E}[X_i] = \frac{1}{P_i}.$$

Étant donné que

$$X = \sum_{i=0}^{n-1} X_i,$$

et que les variables aléatoires X_i sont indépendantes,

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=0}^{n-1} \mathbb{E}[X_i] = \sum_{i=0}^{n-1} \frac{1}{P_i} \\
&= \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=1}^{n} \frac{1}{i} \\
&= n\left(\ln n + \gamma + O\left(\frac{1}{n}\right)\right) \\
&= n\ln n + n\gamma + O(1).
\end{aligned}$$

Déviation de X Dans cette partie, nous aurons besoin de ce petit lemme dont nous laisserons la démonstration aux soins du lecteur

Lemme 9. *Pour $p \in [0,1]$*

$$|(1-p)^r - e^{-rp}| = O\left(\frac{1}{r}\right)$$

donc, pour r assez grand,

$$(1-p)^r \approx e^{-rp}.$$

Notons E_i^r l'événement selon lequel la $i^{\text{ème}}$ fille n'est l'objet d'aucune des r premières propositions. Alors

$$\mathbb{P}[E_i^r] = \left(1-\frac{1}{n}\right)^r \approx e^{-\frac{r}{n}}.$$

Lemme 10. *Les événements $(E_i^r)_{i=1\ldots n}$ sont presque indépendants, c'est-à-dire que pour $i \in \{1,2,\ldots,n\}$ et tout ensemble d'indices $\{j_1,\ldots,j_k\}$ ne contenant pas i,*

$$\mathbb{P}\left[E_i^r \middle| \bigcap_{l=1}^{k} E_{j_l}^r\right] \approx \mathbb{P}[E_i^r].$$

Démonstration. Par définition des probabilités conditionnelles,

$$\mathscr{P} \overset{\text{def}}{=} \mathbb{P}\left[E_i^r \middle| \bigcap_{l=1}^{k} E_{j_l}^r\right] = \frac{\mathbb{P}\left[E_i^r \cap \left(\bigcap_{l=1}^{k} E_{j_l}^r\right)\right]}{\mathbb{P}\left[\bigcap_{l=1}^{k} E_{j_l}^r\right]},$$

le numérateur (respectivement le dénominateur) signifiant que $k+1$ (respectivement k) filles n'ont été l'objet d'aucune des r premières propositions.

On a donc

$$\mathscr{P} = \frac{\left(1-\frac{k+1}{n}\right)^r}{\left(1-\frac{k}{n}\right)^r} \approx \frac{e^{-\frac{r(k+1)}{n}}}{e^{-\frac{rk}{n}}} = e^{-\frac{r}{n}} = \mathbb{P}[E_i^r].$$

□

On peut maintenant calculer la probabilité que le nombre total de propositions soit plus grand que r

$$\begin{aligned} P[X>r] &= \mathbb{P}\left[\bigcup_{i=1}^{n} E_i^r\right] \\ &= 1-\mathbb{P}\left[\bigcap_{i=1}^{n}(\neg E_i^r)\right] \\ &\approx 1-\left(1-e^{-\frac{r}{n}}\right)^n \\ &\approx 1-e^{-ne^{-\frac{r}{n}}}. \end{aligned}$$

En choisissant $r = n(\ln n + c)$ avec $c \in \mathbb{R}$ alors

$$P[X > r = n(\ln n + c)] = 1 - e^{-e^{-c}}$$

ce qui démontre le théorème.

III Généralisations

III.1 Polygamie, polyandrie

Il est possible de généraliser le problème des mariages stables en autorisant les filles (respectivement garçons) à être en binôme avec plusieurs garçons (respectivement filles). Le principe reste le même, sauf que chaque fille indique alors le nombre de garçons qu'elle souhaite avoir et le nombre total de garçons est supposé égal au nombre total de garçons souhaités par les filles. Ce problème est connu sous le nom *Hôpitaux-Résidents* car appliqué pour la première fois aux États-Unis par le *National Resident Matching Program* (NRMP) pour affecter les internes aux hôpitaux en fonction des préférences de chacun.

Afin de résoudre ce problème, nous allons nous ramener au problème des mariages stables et utiliser l'algorithme de Gale-Shapley. Chaque fille va se dédoubler autant de fois qu'elle souhaite avoir de garçons en conservant la même liste de préférences et chaque garçon va intégrer les doubles de chaque fille dans sa propre liste de préférences à la place de l'original. Par exemple si Betty souhaite avoir 3 garçons et que la liste de préférence d'Arthur était $[\ldots,Betty,\ldots]$ alors elle devient $[\ldots,Betty_1,Betty_2,Betty_3,\ldots]$. On a donc maintenant un problème de mariages stables, que nous savons résoudre.

Le problème avec la solution précédente est que nous multiplions les filles et que nous devons à chaque fois recopier les listes de préférences. De plus, si $Betty_1$ est déjà en binôme avec un garçon, il ne sert à rien que $Betty_2$ fasse une proposition à ce garçon, puisque $Betty_1$ et $Betty_2$ sont en réalité les mêmes. Une telle solution ne serait donc pas très efficace en pratique. On va donc autoriser les filles à être dans plusieurs binômes à la fois et une fille, si elle n'a pas atteint le nombre de binômes qu'elle souhaite, pourra faire une proposition au garçon qu'elle préfère parmi ceux à qui elle n'a encore rien demandé.

Si ce sont les filles qui reçoivent les propositions, alors elles peuvent retenir simultanément autant de propositions que le nombre de garçons souhaité, et si ce nombre est dépassé, alors elles devront refuser une des propositions : celle du garçon le moins bien classé dans leur liste de préférences.

III.2 Application aux classes préparatoires

L'algorithme présenté est celui utilisé pour l'affectation des bacheliers aux formations post-bac et pour celle des étudiants des classes préparatoires aux écoles d'ingénieurs. Il permet aux étudiants d'exprimer leurs souhaits réels sans devoir craindre qu'une place ne leur échappe puisque placer une école en fin de liste ne les désavantagera jamais pour intégrer cette école. Notons que la version de l'algorithme utilisée est celle qui avantage les écoles mais, que les étudiants se rassurent, les différences sont en général minimes comme l'a montré une étude [2] dans le cas du NRMP.

On comprend alors l'intérêt pour les écoles de pouvoir utiliser des listes partielles puisque cela leur évite d'avoir à classer tous les candidats, l'expérience leur

permettant d'estimer la taille de la liste complémentaire nécessaire.

III.3 Le problème des colocataires

Il s'agit d'une variation des mariages stables dans laquelle on a tout simplement un nombre pair de personnes (peu importe leur sexe) que l'on souhaite répartir en binômes pour former un appariement stable. L'intérêt de ce problème est qu'il est possible de trouver un cas, avec quatre personnes, dans lequel aucun appariement stable n'existe puisque tout binôme auquel D appartiendrait serait instable :

A	$B > C > D$
B	$C > A > D$
C	$A > B > D$
D	quelconque

En 1985, Irving [3] proposa un algorithme de complexité $O(n^2)$ dont nous allons expliquer brièvement le principe. Le lecteur intéressé pourra se référer à l'article d'Irving [3] pour une analyse détaillée avec toutes les preuves (elles ne sont pas difficiles), ou consulter son livre [4] consacré au problème des mariages stables et ses variations.

Première phase Dans cette phase, de manière très similaire à l'algorithme de Gale-Shapley, chaque personne va faire des propositions qui peuvent être acceptées ou refusées, la principale différence étant la non-symétrie : Arthur peut avoir proposé à Battista qui a accepté mais Battista n'a pas proposé à Arthur.

Cette phase termine soit avec quelqu'un qui n'a plus personne à qui faire de propositions (aucun appariement stable n'existe), soit quand chacun a retenu une proposition et une de ses propositions a été retenue par quelqu'un d'autre. En adaptant les lemmes utilisés pour démontrer l'optimalité pour les garçons de l'appariement de Gale-Shapley, on peut alors éliminer des listes de préférences les personnes qui ne forment pas un binôme potentiel et les listes de préférences obtenues, dites réduites, sont telles que, pour chaque individu, la première personne de sa liste est celle qui a retenu sa proposition et la dernière est celle dont il a retenu la proposition.

Deuxième phase Nous allons maintenant continuer à réduire ces listes jusqu'à ce qu'une d'entre elles soit vide (aucun appariement stable n'existe) ou qu'elles soient toutes réduites à une seule personne, ce qui donnerait un appariement stable.

Pour réduire les listes, il suffit d'en trouver un sous-ensemble (il en existera toujours un) ayant la forme des listes de gauche, c'est-à-dire qu'il existe une sorte de « rotation » entre les deux premiers éléments pour une partie des listes.

X	$A > B > \dots$	$\rightarrow$	X	$\cancel{A} > B > \dots$
Y	$B > C > \dots$		Y	$\cancel{B} > C > \dots$
Z	$C > A > \dots$		Z	$\cancel{C} > A > \dots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$

Alors, s'il existe un appariement stable pour les listes de gauche, il en existe un pour les listes de droite. De plus, par application des lemmes pré-cités, on peut éliminer de la liste de B (respectivement C et A) toutes les personnes moins bien classées que X (respectivement Y et Z).

On peut alors continuer la réduction des listes jusqu'à ce qu'il soit possible de conclure à l'existence ou non d'un appariement stable.

Références

[1] GALE D., SHAPLEY L. S., *College Admissions and the Stability of Marriage*, American Mathematical Monthly 69, 9–14, 1962.

[2] ROTH, A.E., PERANSON E., *The Redesign of the Matching Market for American Physicians : Some Engineering Aspects of Economic Design*, The American Economic Review 89 (4) : 748–780, 1999.

[3] IRVING R. W., *An efficient algorithm for the "stable roommates" problem*, Journal of Algorithms 6 (4) : 577–595, 1985.

[4] GUSFIELD D., IRVING R. W., *The Stable Marriage Problem : Structure and Algorithms*, MIT Press, 1989.