Magistère d'Informatique et Modélisation

Devoir d'Algorithmique : *Mariages*

Pour le 24 Octobre 2000

Laure Danthony

1 Mariage maximum

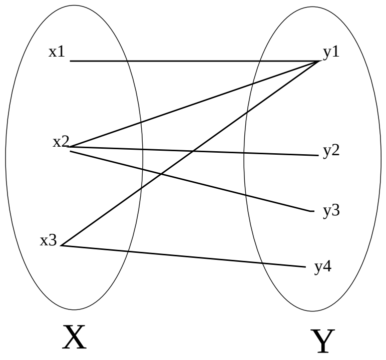

FIG. 1: *Pour se mettre en situation*

Voici une configuration, c'est à dire un graphe G. L'arête $\{x, y\}$, notée xy signifie "x aime y". Si une arête est surlignée en rouge (désolée, je n'ai pas d'imprimante couleur sous la main), celà signifie qu'elle fait partie du mariage considéré.

1. Algorithme de construction d'un mariage maximal

   $M = \emptyset$

   Pour chaque sommet x de X faire

   vérifier qu'il existe y de Y, $(x, y) \in E$ avec y non encore marié.

   si oui, ajouter (x, y) à M

   sinon, ne rien faire. ■

2. Tout mariage maximal n'est pas maximum, par exemple sur le mariage "rouge" de la Figure 2.

   Ce mariage M donné est maximal car on ne peut pas marier x_1 mais il n'est pas maximum car le mariage M' "vert" de la Figure 3 a pour cardinal 3, et il est donc strictement plus grand que M !

   ■

3. Supposons qu'il existe un chemin M-augmentant, c'est à dire un chemin de la forme : $x_1, y_1, x_2, \ldots, x_n, y_n$ avec $y_i x_{i+1} \in M$ (M est de cardinal $n-1$). Alors le mariage $(\forall i, x_i yi)$ est de cardinal n. Donc M n'était pas un mariage maximum. ■

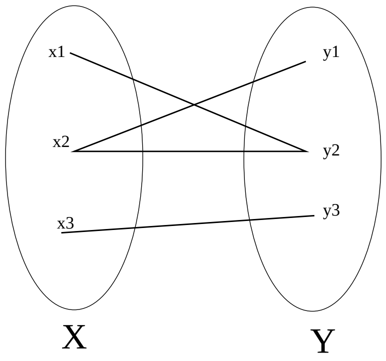

FIG. 2: *Un contre exemple...*

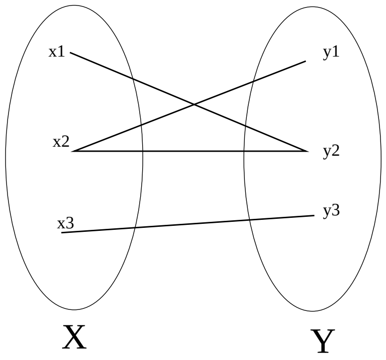

FIG. 3: *Un contre-exemple, suite*

4. Considérons un mariage M et C une couverture de G. Alors chaque élément $e \in E$ a au moins une extrémité $\varphi(e)$ dans C et alors, vu que les mariages ne sont pas polygames : $\forall e \neq e', \varphi(e) \neq \varphi(e')$ donc $\varphi : M \mapsto C$ est injective donc finalement le résultat sur les cardinaux :

$$|M| \leq |C|$$

$\blacksquare$

5.
   - Prouvons tout d'abord la terminaison de l'algorithme. Soit n_s le nombre de sommets de S marqués et n'_s le nombre de sommets de S non marqués. Alors, à chaque tour où l'on ne sort pas, $n'_s \leftarrow n'_s + \{0, -1\}$ En effet :
     - ou bien n'_s ne bouge pas ce qui veut dire que l'on a rajouté un certain w à S (alors $X - S$ a strictement diminué)
     - ou bien n'_s diminue de 1 (par exemple lorsque x n'a pas de voisin).

     Donc, $\big(n' - s, |X| - |S|\big)$ est strictement décroissante pour $(\mathbb{N}, \leq_{lex})$ et comme $\mathbb{N}$ est nœthérien pour $\leq_{lex}$, on en déduit la terminaison. $\blacksquare$

REMARQUE : Plus simplement, on aurait pu dire que la quantité $n'_s + |X| - |S|$ diminue strictement et donc cela termine forcément.

- Prouvons maintenant la correction
  - Montrons que l'algorithme est correct lorsqu'il sort en répondant que M n'est pas maximum. Soit (x_0, y_0) les deux personnes "responsables" de la sortie de l'algorithme. Alors montrons qu'il existe un chemin M-augmentant. En effet :
    - ou bien $x_0 \in U$ et (x_0, y_0) est M-augmentant (voir Figure 4).
    - ou bien $x_0 \notin U$ et x_0 "joue le rôle d'un w", c'est à dire qu'il existe $y_1 \in Y$ non saturé, tel que $x_1 y_1 \in M$ et alors, soit $x_1 \in U$ et alors $y_0 x_0 y_1 x_1$ est M augmentant, soit $x_1 \notin U$ et on continue.

    Comme le nombre de femmes n'est pas infini, on récupère l'existence de $k \in \mathbb{N}$, $x_k \in U$ tel que $y_0 x_0 y_1 \ldots y_k x_k$ est M-augmentant (voir Figure 5), et donc M n'est pas maximum (d'après la question 3).

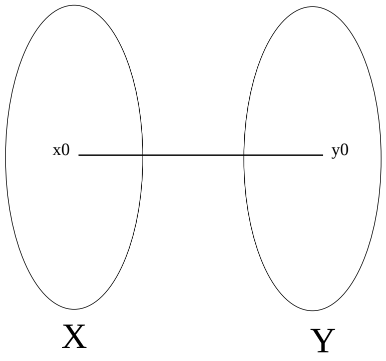

FIG. 4: *Un cas facile ...*

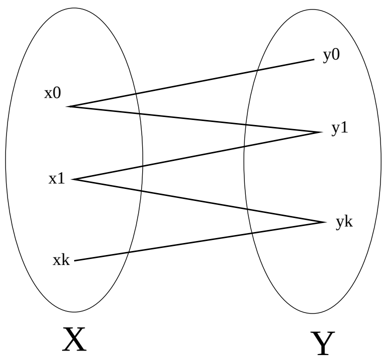

FIG. 5: *Un cas un peu plus difficile*

  - Montrons que si l'algorithme sort en rendant $T \cup (X - S)$, alors c'est bien une couverture. (par l'absurde) Supposons qu'il existe

(x_0, y_0) dans G, tel que ni x_0, ni y_0 n'est dans $T \cup (X - S)$. On a donc à la fois $x_0 \in S$ et $y_0 \notin T$. Notons d'ailleurs que y_0 n'a jamais appartenu à T qui est croissante.
Le sommet x_0 n'appartenait pas à U au début car il n'est pas insaturé. Donc x_0 a été rajouté à S au cours du déroulement de l'algorithme. Lors de cet ajout :

- x_0 était un "w" donc il existe y dans Y tel que $(x_0, y) \in M$ avec $y \in T$ par la suite.
- x_0 a été marqué (après avoir fait la boucle). A l'intérieur de cette boucle, y_0 n'a pas été rajouté à T, donc, puisque y_0 n'appartenait pas à T avant, $(x_0, y_0) \in M$.

Comme trivialement, $y \neq y_0$ (l'un est dans T, l'autre non!), il y a comme un petit problème car x_0 est mariée à y_0 et à y en même temps !
Donc, à la sortie de l'algorithme, $T \cup (X - S)$ est bien une couverture. ■

6. Dans le cas où l'algorithme renvoie une couverture, comparons la taille de cette couverture avec celle de M :

   Considérons la quantité $|T|+|X - S|$. C'est un invariant de boucle car, soit ni T ni S ne varient, soit T et S augmentent tous les deux d'un élément. Donc:
$$|T| + |X - S| = |X - U| = |M|$$
   (ensemble des sommets non insaturés)

   D'où
$$|C| = |M|.$$
■

   Alors, M est maximum car, supposons qu'il existe un autre mariage M_1 tel que $|M| < |M_1|$ alors on récupère, d'après la question 4, $|M| < |M_1| \leq |C|$, soit encore $|M| < |C|$, ce qui est contradictoire avec $|C| = |M|$.

   Donc, si l'algorithme rend une couverture, M est maximum. ■

7. Pour construire un mariage maximum, on effectue le même algorithme, sauf qu'à la place de sortir avec "M non maximum", on construit M' tel que $|M| < |M'|$ avec l'aide de la question 3, et on relance l'algorithme **entier** sur M'. ■

8. Dans l'exemple de la Figure 6, M n'est pas maximum et il n'existe pas de chemin augmentant.

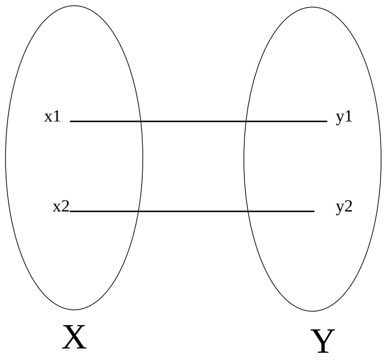

FIG. 6: *Encore un contre-exemple !*

2 Mariages stables

1. Montrons que dans le cas donné, il n'y a pas d'internat stable :

   Dans le cas $(a, b) \parallel (c, d)$, b et c sont perturbateurs ;

   Dans le cas $(a, c) \parallel (b, d)$, a et b sont perturbateurs ;

   Dans le cas $(a, d) \parallel (b, c)$, a et c sont perturbateurs.

   Donc, dans tous les cas, les internats sont instables. ■

   REMARQUE : Heureusement, il n'y a que 4 personnes dans cet internat, parce que la méthode d'énumération, c'est long !

2. Illustration de l'algorithme sur l'exemple donné :

   - Boucle 1 :
     - x demande a en mariage ;
     - y demande a en mariage ;
     - z demande c en mariage ;
     - w demande c en mariage ;
     - a $(k = 2)$ rejette w, $a : z < x < y$;
     - c $(k = 2)$ rejette z, $a : w < x < y$.
   - Boucle 2 :
     - x demande a en mariage ;
     - y demande a en mariage ;
     - z demande d en mariage ;
     - w demande c en mariage ;
     - a $(k = 2)$ rejette y, $a : z < x$.
   - Boucle 3 :
     - x demande a en mariage ;
     - y demande c en mariage ;
     - z demande d en mariage ;

- w demande c en mariage ;
  - c ($k=2$) rejette y, $c : w < x$.
- Boucle 4 :
  - x demande a en mariage ;
  - y demande b en mariage ;
  - z demande d en mariage ;
  - w demande c en mariage.

  Ainsi l'algorithme répond le mariage suivant :

$$\{(x,a),(y,b),(z,d),(w,c)\}$$

■

3. • Prouvons la terminaison de l'algorithme. Pour celà remarquons :
   - Si, à une étape, il n'y a pas de rejet, chaque homme a reçu exactement une demande, et donc on sort.
   - Le nombre de rejets effectués, borné par $n(n-1)$, augmente donc strictement à chaque tour où l'on ne sort pas.

   D'où la terminaison. ■

   • Prouvons la correction de l'algorithme. Supposons que le mariage suivant "créé" par l'algorithme : $M = \{(f,h),(f',h'),\ldots\}$ soit instable, c'est à dire que l'on a à la fois $f : \ldots > h' > \ldots > h > \ldots$ et $h' : \ldots > f > \ldots > f' > \ldots$. Alors
   - f a été mariée à h par l'algorithme alors qu'elle préfère h' : elle a donc à la fin demandé h en mariage, ce qui implique qu'**elle a été rejetée par h' (plus haut dans sa liste) à un moment de l'algorithme**.
   - h' a été marié à f' alors qu'il préfère f, il a donc reçu une demande de f' qu'il n'a pas rejetée. Dès lors, h' **n'a jamais été demandé en mariage par** f (car sinon, il aurait rejeté au moins une par une les filles les plus basses sur sa liste, et donc aurait fini par rejeter f').

   Il y a donc comme un petit problème, et donc M est stable. ■

4. • Montrons par l'absurde que M est optimal du point de vue des femmes. Pour celà, notons M le résultat de l'algorithme sur une certaine distribution de femmes et d'hommes. Notons M' un autre mariage stable, quelconque. Supposons que f soit mariée dans M' à h_1 qu'elle préfère à son mari h dans M : $f : h_1 > h$.

   Dans la Figure 7, le mariage M est surligné en rouge, le mariage M' en vert. Dans l'algorithme, f a forcément demandé h_1 avant de demander h qu'elle aime moins. En conséquence, h_1 a repoussé f dans l'algorithme. En notant h_2 l'époux de f_1 dans M',
   - ou bien $f_1 : h_1 > h_2$ (f_1 préfère h_1 à h_2), ce qui est impossible car M' est stable par hypothèse,

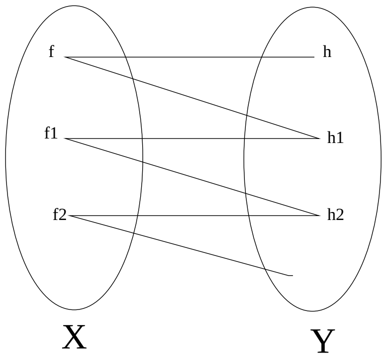

FIG. 7: *M, mariage optimal pour les femmes*

- ou bien $f_1 : h_2 > h_1$ et alors h_2 a rejeté f_1 dans l'algorithme. Dans ce cas, en notant f_2 la femme de h_2 dans M' :
  - ou bien f_2 préfère h_2 à h_1 ce qui est impossible car M' est stable,
  - ou bien f_2 préfère h_1 et c'est reparti pour un tour

Dès lors, soit on récupère une absurdité tout de suite avec M' stable, soit on récupère une suite telle que, quel que soit i f_i préfère h_{i+1} à h_i et h_i préfère f_i à f_{i-1}.

D'après le déroulement de l'algorithme, quel que soit l'indice i, il existe forcément un tour de boucle où f_i demande h_{i-1} en mariage, et ce, quel que soit i . Soit I l'ensemble des ces indices de boucles ainsi récupérés, et n_0 son plus petit élément. Alors : au tour n_0, quel que soit i, f_i demande h_{i-1} . J'affirme qu'alors l'algorithme marie f_i et h_{i+1}, en effet : f_i demande h_{i+1} et h_{i+1} ne le rejette pas car il n'a pas reçu la demande de f_{i+1} et vu que dans la suite du déroulement de l'algorithme f_{i+1} ne demandera jamais h_{i+1}, l'algorithme a marié, pour chaque i chaque f_i à h_{i+1}.

Alors, en appliquant l'algorithme M n'est plus le mariage M ! Et donc, finalement, M est donc optimal du point de vue des femmes[1].

■

- Montrons que l'algorithme est défavorable du point de vue des hommes, c'est à dire que si M' est un autre mariage stable, tout homme est marié à une femme qu'il préfère à sa femme dans M. Montrons ce résultat par l'absurde, en utilisant l'optimalité de l'algorithme du point de vue des femmes. Supposons donc qu'il existe h' marié dans M' à f' et dans M à f, tel que $h' : f > f'$.Alors, en notant h le mari de f dans M', comme M' est stable, $f : h > h'$ et f est mariée dans M' à h qu'elle préfère à son époux dans M. Or ceci est impossible d'après ce qu'on a démontré avant. L'algorithme est donc défavorable du point de vue des hommes[2]. ■

1. Heureusement, car c'est Natacha qui a fait l'énoncé du D.M.
2. Les femmes proposent, les hommes disposent

5. On définit la stabilité avec célibataires de la façon suivante : il n'y a pas de paire de couples instables, une femme ne préfère pas un homme célibataire à son mari, un homme ne préfère pas une femme célibataire à sa femme. De plus, il n'existe pas un homme et une femme qui sont célibataires en même temps (c'est à dire dans le même mariage).

   On se place dans le cas où les femmes sont plus nombreuses que les hommes (cas habituel!) et donc les célibataires de mariages stables sont uniquement des femmes.

   Un algorithme de mariage consiste à reprendre l'algorithme précédent en remplaçant "jusqu'à ce que chaque homme reçoive **exactement** une demande" par "**au plus** une demande". On montre de même que précédemment (en complétant les hommes par des "hommes spéciaux" qui sont en fait des ensembles vides, que chaque femme place en queue de sa liste de préférence), que cet algorithme construit un mariage stable, favorable aux femmes et défavorable aux hommes. ■

6. Trivialement, le nombre de personnes qui restent célibataires dans un mariage est constant : $||Y| - |X||$ Montrons que l'ensemble des célibataires reste le même. Soient M et M' 2 mariages stables. Supposons que f_1 soit mariée à h_1 dans M et f_1 célibataire dans M'. Alors soit f_2 la femme de h_1 dans M' (on remarque que h_1 n'est pas célibataire dans M' car sinon on aurait pu marier h_1 et f_1 dans M', et trivialement $f_1 \neq f_2$). On a $h_2 : f_2 > f_1$ par stabilité de M'. Soit alors h_2 le mari de f_2 dans M (f_2 ne peut être célibataire par stabilité de M). De plus $h_1 \neq h_2$ car $f_1 \neq f_2$ (deux personnes ayant des maris différents, par monogamie sont forcément deux personnes différentes). En faisant de même, on récupère deux suites : f_i et y_i tq :

   - f_i marié à h_i dans M, pour tout i,
   - f_{i+1} marié à h_i dans M',
   - f_0 célibataire dans M',
   - tous les f_i et h_i sont distincts (on le montre facilement par l'absurde).

   Ces suites étant infinies, il y a comme un petit problème car on dispose d'un nombre fini de protagonistes. En conséquence, l'ensemble des célibataires reste le même dans tous les mariages stables. ■

7. D'après la question précédente, les pauvres célibataires restent les mêmes, quel que soit le mariage stable. On peut alors raisonnablement se demander s'il en est de même s'ils changent leurs listes de préférences. Après avoir essayé (en vain) de trouver un cas où un célibataire malin change sa liste et trouve à se marier, j'en suis arrivée à la conclusion que ce n'est pas possible (mais ce n'est pas une démonstration !). Ceci dit, la littérature sur le sujet trouve des cas où il faut être malin pour se marier, mais en se plaçant dans le cas où toutes les personnes ne sont pas classées dans une liste de préférence (c'est à dire que certaines personnes préfèrent être célibataires que mariées à leur ennemi personnel). ■