Corrigé du TD 2

3 février 2004

1. a. Il suffit d'ajouter au graphe G une source, à laquelle on relie toutes les places par des arêtes de capacité 1, et un puits, auquel on relie toutes les issues, de capacité 1. On attribue également à chaque arête du graphe une capacité de 1. Il faut cependant ajouter, pour éviter que deux sommets aient un sommet commun, une contrainte au sommet disant que le flot entrant est au plus 1.

L'application d'un des algorithmes de flot vus dans le cours fournit un flot maximal à valeurs dans $\{0, 1\}$. À chaque arête traversée par un flot de capacité 1 on associe une arête du plan d'évacuation. La contrainte sur les sommets assure que les différents chemins sont sommet-disjoints : au plus une arête entrant dans un sommet peut être saturée, puisque le flot est à valeurs dans $\{0, 1\}$. Enfin, la conservation du flot en chaque sommet montre que chaque arête appartient à un sommet allant d'une place à une issue. Tout flot à valeurs $\{0, 1\}$ traversant le graphe fournit donc un plan d'évacuation.

Inversement, partant d'un plan d'évacuation, on sature toutes les arêtes des chemins du plan d'évacuation, ainsi que l'arête allant de la source à la place commençant le chemin et l'arête allant de l'issue au puits. Chaque chemin mène d'une place à une issue ; la conservation du flot entre la source et le puits montre qu'on construit bien un chemin allant d'une place à une issue, et les contraintes sur les sommets sont assurées par le fait que les chemins sont sommet-disjoints.

On a donc bien réduit la recherche du plan d'évacuation à la recherche d'un flot entier dans le graphe construit.

b. On construit un nouveau graphe en remplaçant tout sommet distinct de la source et du puits par deux sommets reliés par un arc de capacité 1, l'un où rentrent les arcs entrants, l'autre dont sortent les arcs sortants. Cette construction limite automatiquement le flot entrant en le sommet à une valeur de 1.

c. Il suffit alors d'appliquer l'algorithme de Ford-Fulkerson au graphe ainsi construit. Le nombre d'arêtes de ce graphe est $O(m + n)$, le nombre de sommets $O(n)$. La complexité est $O((m + n)\Phi)$, où Φ est la valeur du flot maximal, ici $O(n)$ (puisqu'une unité de flot au plus va de chaque arête vers chaque place). La complexité totale est donc $O((m + n)n)$. À noter qu'Edmonds-Karp donnerait $O(n(m + n)^2)$, qui est moins bon dans ce cas.

2. On considère (x_0) comme la source et (y_0) comme le puits d'un réseau, en ajoutant

un poids de 1 sur chaque arête. L'algorithme de Ford-Fulkerson garantit de trouver un flot à valeur entière, et donc à valeur $\{0, 1\}$. On considère alors le recouvrement par des chemins obtenu en gardant les arêtes (i, j) pour lesquelles l'arête (x_i, y_j) de A est saturée. Noter que cela ne peut être le cas que pour au plus une arête issue de x_i et pour au plus une arête issue de y_j, ce qui montre que chaque sommet a au plus une arête entrante et une arête sortante.

La valeur du flot maximal est alors égale au nombre d'arêtes du recouvrement par chemins ainsi construit (considérer le flot traversant la coupe $\{(x_i)\}, \{(y_i\}$. Comme chaque chemin a une arête de moins que son nombre de sommets (le graphe ne contient pas de cycles), le flot maximal vaut le nombre de sommets du recouvrement moins le nombre de chemins.

b. L'algorithme ne fonctionne pas pour les graphes orientés contenant des circuits car l'ensemble C peut contenir des circuits et la remarque ci-dessus ne s'applique plus. D'ailleurs si un algorithme polynomial existait on pourrait savoir en temps polynomial si un graphe admet un chemin hamiltonien, problème NP-complet.

3. On considère le graphe formé de la façon suivante : une source s, un puits t, un sommet par homme h_i, un sommet par femme f_j et un sommet par agence a_k. Les arêtes sont de la forme (s, h_i), de capacité 1, (f_j, t), de capacité 1, et (h_i, a_k), (a_k, f_j) pour les (i, k) tel que l'homme i soit dans le fichier de l'agence k, et (k, j) tels que la femme j soit dans le fichier de l'agence k.

On ajoute de plus une contrainte sur chacun des sommets a_k : le flux entrant doit être au plus égal à b_k. On a vu plus haut que cela pouvait s'obtenir en terme de contraintes sur les arêtes en dédoublant le sommet a_k en $a_k^{(1)}$ et $a_k^{(2)}$, en faisant entrer les arêtes en $a_k^{(1)}$ et sortir les arêtes en $a_k^{(2)}$, les deux sommets étant reliés par une arête de capacité b_k.

Montrons qu'il y a correspondance entre un ensemble de mariages valides et un flot entier dans le graphe ainsi construit. Soit un ensemble de mariages valides. On construit un flot en faisant passer une unité de flot par chaque arête (h_i, a_k) et (a_k, f_j) telle que l'agence k ait arrangé le mariage entre l'homme i et la femme j. Il est facile de voir que le flot est bien conservé, et que la contrainte sur le sommet a_k est manifestement vérifiée.

Inversement, à un flot dans le graphe, on peut associer une affectation d'hommes et de femmes à chaque agence, de sorte que le même nombre d'hommes et de femmes soit associé à chaque agence. Il suffit alors, pour chaque agence, de choisir les mariages de façon arbitraire.

On peut donc résoudre le problème initial par une technique type Edmonds-Karp, qui conduit à une complexité $O(SA^2)$ et garantit que le flot final est entier, les capacités d'arcs étant entières. Le nombre de sommets du graphe est $O(n+m)$; le nombre d'arêtes est $O(nm)$. La complexité dans le cas le pire est donc $O(n^2m^2(n+m))$.

4. Fixons un sommet s du graphe, et pour un sommet $t \neq s$ définissons $R(s, t)$ comme le plus petit nombre d'arêtes à enlever à G pour qu'il ne soit plus possible d'aller de s à t. Le nombre $R(s, t)$ s'évalue par un calcul de flot : on construit un nouveau graphe orienté dans lequel on dédouble chaque arête pour orienter le graphe, que l'on attribue un poids 1 à chaque arête ; s est la source et t le puits. Le nombre $R(s, t)$ vaut alors exactement la coupure minimale (c'est la définition), donc le flot maximal.

Montrons maintenant que la résistance aux pannes $R(G)$ vaut $\min_{t \in G-\{s\}} R(s, t)$. Il est clair, par définition, que comme chaque $R(s, t)$ est associé à un ensemble d'arêtes déconnectant G, on a $R(G) \leq \min_{t \in G-\{s\}} R(s, t)$.

Si A est un ensemble minimal d'arêtes déconnectant G, le graphe résultant de la suppression des arêtes de A a deux composantes connexes, dont l'une contient s ; soit t un sommet de l'autre composante. On a alors $R(s, t) \leq R(G)$, puisque supprimer A sépare s de t. Par suite, $R(G) = \min_{t \in G-\{s\}} R(s, t)$, et on a montré comment déterminer $R(G)$ en n appels à un algorithme de flot, soit un coût total de $O(nm\Phi) = O(n^2m)$, le flot maximal étant manifestement $O(n)$.

3. Flots.

Modélisation : réseau à $n+m+2$ sommets : le source s, n sommets $p_1, \ldots, p_n$, m sommets $s_1 \ldots s_m$, et le puits t.

Pour tout $j \in \{1, \ldots, n\}$ et tout $i \in \{1, \ldots, m\}$ tels que $p_j \in S_i$, on a un arc (p_j, s_i) de capacité 1. De plus, pour tout $j \in \{1, \ldots, n\}$, on a un arc (s, p_j) de capacité $\lceil \sum_{i:p_j \in S_i} 1/|S_i| \rceil$, et pour tout $i \in \{1, \ldots m\}$, on a un arc (s_i, t) de capacité 1.

On résoud un problème de flot maximal sur le réseau ainsi défini.

Analyse.

Puisque la coupe $\{s, p_1, \ldots, p_n, s_1, \ldots, s_m\}, \{t\}$ a capacité m, le flot maximal aura valeur au plus m. S'il a valeur exactement m, ce flot, qui par l'algorithme du cours a valeur entière sur chaque arc, associera exactement un conducteur à chaque S_i, et la contrainte de capacité sur les arcs (s, p_j) garantit que la répartition obtenue sera équitable.

Il existe évidemment un flot de valeur m, à savoir le flot *fractionnaire* qui fait passer exactement $1/|S_i|$ quantité de flot sur chaque arc (p_j, s_i) ; donc la flot maximal aura valeur exactement m, et le flot entier de valeur m obtenu par l'algorithme du cours donnera donc toujours une répartition équitable.