

TD n°8

Flot et usage de flot

1 Comment trouver du flot

On rappelle qu'un *flot* sur un graphe G orienté pondéré par une pondération c est le fait d'attribuer à chaque arc $(u; v)$ une valeur $f(u; v)$ telle que :

1. Pour tout arc $(u; v)$, $0 \leq f(u; v) \leq c(u; v)$ (où $c(u; v)$ est la capacité de l'arc) ;
2. Pour tout sommet u (tel que $u \neq s$ et $u \neq t$), on a $\sum_v f(v; u) = \sum_w f(u; w)$ (loi des nœuds) ;

Les sommet s et t sont la *source* et le *puits*. La *charge* du flot est $\sum_v f(s; v)$ (égal à $\sum_w f(w; t)$).

Un flot maximum est un flot de charge maximum. L'algorithme de Ford-Fulkerson fonctionne ainsi :

Soit F un flot vide (pour tout arc $(u; v)$, $f(u; v) = 0$)

tant que *Il existe un chemin améliorant* **faire**

Augmenter le flot F le long de C de la charge de C

fin

Renvoyer F

Exercice 1 Une définition naturelle de *chemin améliorant* serait :

C est un chemin de s à t tel que le long de tout arc $(u; v)$ de ce chemin, $c(u; v) - f(u; v) > 0$.

La charge de C est alors le minimum de $c(u; v) - f(u; v)$ le long du chemin.

Or, ça ne marche pas ! Donner un petit exemple où ce pseudo-Ford-Fulkerson échoue avant de trouver un flot maximum.

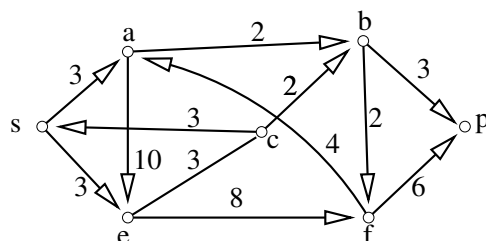
On s'autorise une définition plus large de "chemin améliorant" : c'est un chemin dans le **graphe des augmentations** G_ϕ de G .

- Il a les mêmes sommets que le graphe de départ
- Il a une fonction de capacité a
- Pour tout arc $(u; v)$ de G , si cet arc n'est **pas saturé** (càd $c(u; v) \neq f(u; v)$) alors on crée un arc $(u; v)$, dit **arc à l'endroit**, de capacité $a(u; v) = c(u; v) - f(u; v)$
- Pour tout arc $(u; v)$ de G , si cet arc est **utilisé** (càd $f(u; v) \neq 0$) alors on crée un arc $(v; u)$, dit **arc à l'envers**, de capacité $a(v; u) = f(u; v)$

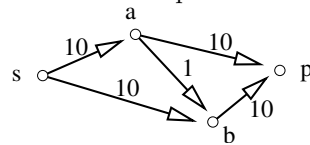
Noter que comme le flot f de G ne peut utiliser à la fois un arc $(u; v)$ et un arc $(v; u)$ il n'y a jamais plus d'un arc à l'endroit et un arc à l'envers entre deux sommets u et v de G_ϕ .

Exercice 2 Quelle est la capacité d'un chemin dans le graphe des augmentations ? Comment un chemin dans le graphe des augmentations modifie-t-il le flot dans le graphe initial ?

Exercice 3 Appliquer l'algorithme sur le graphe suivant. À chaque étape, donner le graphe des augmentations.



Exercice 4 Montrer que le temps d'exécution de Ford-Fulkerson peut être *tres long*, c'est-à-dire qu'il n'existe pas de fonction f tel que le temps d'exécution de Ford-Fulkerson est $O(f(n;m))$ où n est le nombre de sommets et m est le nombre d'arcs. Vous pourrez vous inspirer du graphe suivant :



2 Couplage de graphes bipartis

On considère un graphe *biparti* $G = (A \cup B; E)$. L'ensemble des sommets est coupé entre deux sous-ensembles A et B . Toute arête du graphe est entre un sommet de l'ensemble A et un sommet de l'ensemble B .

Un *couplage* d'un graphe biparti est un ensemble d'arêtes tel que deux arêtes différentes n'ont pas de sommet en commun. Un couplage de G est *maximum* s'il n'existe pas de couplage de G ayant plus d'arêtes.

Exercice 5 Comment peut-on résoudre le problème du couplage maximum grâce à Ford-Fulkerson ?

indication : il faut rajouter une source et un puits quelque part ! Car Ford-Fulkerson en a besoin, et on ne voit pas quels sommets de G pourraient jouer ce rôle

Exercice 6 Appliquer votre algorithme pour résoudre les problèmes de cœur des amoureux ci-dessous : ils voudraient bien se marier, mais afin de changer un peu un couple "traditionnel", signalé par un (\heartsuit), ne convolera que si aucun des deux n'a moyen de faire autrement.

	Roméo	Paul	Abélard	Charden	les Musclés
Juliette	(\heartsuit)	\heartsuit		\heartsuit	
Virginie	\heartsuit	(\heartsuit)		\heartsuit	
Héloïse	\heartsuit		(\heartsuit)		
Stone	\heartsuit			(\heartsuit)	\heartsuit
Dorothée				\heartsuit	(\heartsuit)

3 Plan d'évacuation.

On considère un graphe non orienté $G = (V; E)$ ayant n sommets et deux sous-ensembles Y et Z de V (non nécessairement disjoints) ; les éléments de Y sont des places et les éléments de Z des issues. Un *plan d'évacuation* est un ensemble de chemins sommet-disjoints dans le graphe, menant d'une place à une issue. Déterminer un plan d'évacuation optimal consiste à trouver un plan d'évacuation contenant le maximum de chemins.

Un *probleme du flot avec contraintes sur les sommets* est un problème de flot classique auquel on ajoute des contraintes supplémentaires : à chaque sommet i est associé un nombre S_i exprimant que la somme des flots entrants en i doit être inférieure ou égale à S_i .

Exercice 7

1. Réduire le problème du plan d'évacuation à un problème du flot avec contraintes sur les sommets.
2. Réduire le problème du flot avec contraintes sur les sommets à un problème de flot.
3. En déduire un algorithme polynomial de recherche d'un plan d'évacuation optimal.

4 Covoiturage

Un groupe de n personnes $P = \{p_1; \dots; p_n\}$ partagent leurs véhicules pour aller travailler pendant m jours. Le jour i , un sous-ensemble S_i de P utilisent un véhicule commun pour aller travailler. Le conducteur ce jour-là sera un élément de S_i . Etant donné P et les sous-ensembles $S_1; \dots; S_m$, le problème est de choisir pour chaque i le conducteur du jour i de façon que la répartition soit équitable, c'est-à-dire qu'une personne j ne conduise pas plus de $\lceil \sum_{i:j \in S_i} \frac{1}{|S_i|} \rceil$ jours au total. Par exemple, s'il y a quatre personnes et trois jours avec $S_1 = \{p_1; p_2\}$, $S_2 = \{p_1; p_3; p_4\}$ et $S_3 = \{p_1; p_4\}$ alors la personne p_1 doit conduire au maximum $\lceil 1/2 + 1/3 + 1/2 \rceil = 2$ de ces trois jours alors que les autres conducteurs doivent conduire au maximum 1 de ces trois jours.

Modéliser ce problème comme un problème de flot maximal.

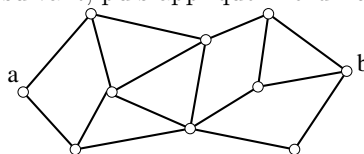
Monter qu'il existe toujours une répartition équitable.

5 Connectivité

Étant donné deux sommets a et b , on peut construire un réseau $R_{a,b}(G)$ tel que :

- Tout sommet x du graphe, sauf a et b , est dédoublé en x_d et x_a
- a est remplacé par a_d (départ) et b est remplacé par b_a (arrivée)
- Entre x_a et x_d on met un arc $(x_a; x_d)$ de capacité 1
- Pour toute arête $(x; y)$ du graphe on crée un arc $(x_d; y_a)$ de capacité infinie et un arc $(y_d; x_a)$ de capacité infinie.

Exercice 8 Transformez le graphe suivant, puis appliquez Ford-Fulkerson :



Exercice 9

1. Qu'est-ce qu'un chemin améliorant dans ce graphe (sans arc pris à l'envers) ?
2. Quelles propriétés ont deux chemins améliorants (sans arc pris à l'envers) ?
3. Une *coupe d'arcs entre a et b* est un ensemble A tel que dans $G - A$ (en enlevant ces arcs) b n'est plus accessible depuis a . Qu'est-ce qu'une coupe d'arcs entre a et b de capacité finie dans ce graphe ?
4. Et qu'est-ce que la coupe d'arcs minimale ?
5. Donc, que représente le flot maximal entre a et b ?

La *connectivité* d'un graphe (non orienté quelconque) G , notée $c(G)$, est le plus petit nombre de sommets qu'il faut enlever pour que le graphe soit déconnecté ou trivial (un seul sommet).

Formellement, c'est la taille minimum d'un ensemble $S \subset V$ tel que $G[V - S]$ ait plusieurs composantes connexes ou soit réduit à un seul sommet.

On admettra sans preuve le *theoreme de Menger* : $c(G)$ est le minimum, pour tous couples $(a; b)$ de sommets, du nombre de chemins disjoints (n'ayant aucun sommet en commun, sauf a et b) de G . On admettra aussi que si les charges sont des nombres entiers, alors il existe une version de l'algorithme de Ford-Fulkerson (dite de Edmond-Karps) qui s'exécute au plus en temps $O(mn^2)$.

Exercice 10 Proposer un algorithme qui calcule la connectivité d'un graphe. Quelle est sa complexité ?