

Des problèmes sur les réseaux

Classement

- **Affectation** Réseau trivial? ensemble de sources (disponibilité = 1) et de puits (demande = 1) et où chaque arc est d'une source à un puits
- **Transport** Demandes, disponibilités (et capacités des arcs) peuvent être supérieures à 1 mais toujours chaque arc est d'une source à un puits
- **Transbordement** Réseau général avec capacités infinies
- **Flot maximum** Réseau général avec capacités finies
- **Flot de coût minimum** Réseau général avec capacités finies et coûts différents sur les arcs

L'algorithme de Ford et Fulker- son

Calcul du flot maximum

Il suffit de considérer le cas d'une source et un puits
Basé sur l'idée d'une **chaîne augmentante**: une suite de changements possibles $\pm\alpha$ dans une chaîne d'arcs dont le résultat est de transférer quantité α de la source à un autre sommet.

NB: l'augmentation dans une chaîne augmentante peut diminuer le flux dans certains arcs!

Facile de trouver des chaînes augmentantes dans un arbre enraciné à la source

Si on trouve une chaîne augmentante au puits, on peut augmenter le flot actuel (initialisé à 0)

par le minimum des changements possibles sur les arcs de la chaîne

Sinon on a trouvé un flot maximum.

La coupe minimum

Quand l'algorithme ne trouve plus de chaîne augmentante au puits, il a trouvé un ensemble E de sommets autour de la source (ceux pour lesquels il existe une chaîne augmentante) tel que tous les arcs partant de E sont saturés (et tous ceux entrant en E ont flux nul);

dans le flot maximum trouvé le flux total sur ces arcs partant de E est le même que le flot total trouvé; donc l'algorithme a trouvé une **coupe** avec cette valeur.

Théorème: cette coupe est minimum parmi les coupes séparant la source et le puits.

Du point de vue de la PL, la recherche d'une coupe minimum est le problème dual du celui du flot maximum!

Les problèmes sur les réseaux

sont des programmes linéaires

- Une variable pour le flot dans chaque arête orientée
- Disponibilités, capacités de transfert, capacités d'absorption sont toutes exprimées par des contraintes linéaires
- Fonction objective linéaire : flot total arrivant à la destination, coût total, ...
- Propriété fondamentale d'un réseau : ce qui part doit arriver
facilement exprimé par une contrainte linéaire
- (et inversement?)

La coupe minimum

Pourquoi a-t-on un flot maximum quand il n'existe pas de chaîne augmentante de la source au puits?

- Les sommets vers lesquels on peut trouver une chaîne (faire passer un flot) constituent un ensemble X autour de la source
- Les autres sommets constituent un ensemble Y autour du puits
- Chaque arête de X à Y est saturée
- Chaque arête de Y à X a flot nul
- Donc, le flot de X à Y ne peut pas être augmenté mais tout le flot source \rightarrow puits doit passer de X à Y
- Les arêtes de X à Y sont une **coupe** du graphe.

Théorème

La coupe minimum est égal au flot maximum.

Encore une méthode

- Construire un flot partiel où chaque sommet sauf la source transmet au maximum ce qu'il reçoit
- si possible d'étendre ce flot, faites-le
- sinon faire reculer ce qui est arrivé à un sommet et ne peut pas être transmis
- basé sur l'idée de hauteur : le flot est toujours du haut vers le bas; pour modifier le flot, il faut incrémenter la hauteur (au minimum des hauteurs des voisins utiles +1)
- au début hauteur nulle sauf pour la source (hauteur = nombre de sommets); les hauteurs de la source et du puits ne changent pas.
- complexité $O(n^3)$ (si la recherche du prochain sommet à traiter est bien gérée).

Pourquoi cette méthode marche-t-elle?

- un petit exemple
- elle doit trouver un flot (les surplus sont forcément rejetés vers la source; en fait sans dépasser hauteur de $2 \times$ nombre de sommets)
- ce flot est-il maximum?
- quand le flot commence à reculer vers la source (première fois une hauteur dépasse le nombre de sommets), s'il y avait un chemin faisable pour passer un surplus vers le puits, ce chemin aurait une différence de hauteur entre deux sommets consécutifs de au moins 2, ce qui est impossible.

Rapport aux méthodes de parcours des solutions de base

L'algorithme de F et F, est-il une méthode de parcours des solutions de base comme le simplexe?

...?

Non. Il peut trouver un flot maximum qui n'est pas une solution de base du PL.

(Considérer un réseau ne possédant que deux chemins source – X de capacités 1 chacun et deux chemins X – puits de capacité 2.)

L'algorithme de Dinic ou Edmonds-Karp

Un algorithme qui trouve le flot maximum en temps polynomial

A l'itération d , il trouve le flot maximum parmi ceux qui n'utilisent que les chemins de longueur $\leq d$

Pour l'itération d , il construit un nouveau graphe qui est une copie partielle de celui du réseau mais qui ne contient que les arcs susceptibles d'être inclus dans une chaîne augmentante de longueur d

Il utilise ce graphe pour chercher une chaîne augmentante de longueur d (et termine cette itération s'il n'y en a pas)

Après chaque augmentation, il découpe du graphe tout arc dont le flux vient d'être réduit à nul ou augmenté à sa capacité

(et aussi tout autre arc qui n'est plus utile pour construire des chaînes augmentantes).

Equivaut, *grosso modo* à Ford et Fulkerson, mais avec recherche en largeur pour une chaîne augmentante au lieu de recherche en profondeur.

Complexité $O(m^2n)$ (ou $O(mn^2)$ avec des idées de Dinic) pour un graphe de n sommets et m arêtes.

Flot maximum avec coût minimum

L'algorithme de Busacker et Gowen

Itérativement améliorer un flot (initialisé à 0)

A chaque itération construire un graphe R qui dépend du flot actuel f et le graphe du réseau G : si G a un arc (x, y) avec coût c ,

- si (x, y) n'est pas saturé, R a un arc (x, y) valué à c
- si (x, y) a un flux non nul, R a un arc (y, x) valué à $-c$

On cherche un chemin de la source au puits avec une valeur minimum ; s'il existe, on augmente le flot selon la chaîne augmentante correspondant sinon l'algorithme termine.

Cette recherche est effectuée par un algorithme bien connu de recherche du plus court chemin entre deux sommets (seule complication ici, arcs de "distance" négative mais c'est pas grave)

La raison pour laquelle cet algorithme réussit est plus compliquée, en particulier, si on commençait d'un flot aléatoire, il ne serait pas garanti qu'on arrive à un flot optimal;

mais, en commençant au flot nul, oui.

Equivalut à Ford-Fulkerson avec choix de la chaîne augmentante de coût minimum.