Des problèms sur les réseaux exprimés en programmes linéaires 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

Ici on considère des algorithmes pour les deux derniers problèmes

Flot maximum

pourquoi le PL est (très) dégénéré

On peut traiter un problème de flot maximum par un algorithme général de PL. Mais
les seules équations sont des formes (pour un arc de capacité c) $x_i = 0$ et $x_i = c$
(pour un sommet avec m arcs entrants et n sortants) $x_{i_1} + \cdots + x_{i_m} \cdots = x_{j_1} + \cdots + x_{j_n} \cdots$ (principe de conservation de flot) et pareilles pour chaque source et puits.

Donc à un sommet où il y a flux nul, les équations $x_i = 0$ et celle de conservation sont vérifiées; mais la dernière est conséquence des autres!

L'algorithme de Floyd et Fulkerson
Flot maximum

Il suffit de considérer le cas d'une source et un puits

Basé sur l'id'ée 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!

Un exemple

Le temps de calcul

Chaque itération trouve une chaîne augmentante (ou échoue) par un parcours du graphe; en temps linéaire

Chaque itération (sauf la dernière) augmente le flot

Donc le nombre d'itérations est borné par le flot maximum trouvé ($+1$), donc par la somme des capacités des arcs sortant de la source (ou entrant dans le puits)

Donc temps total raisonnable si les capacités sont petites
Un algorithme pseudo-polynomial

Peut être pire que le simplexe quand les capacités sont élevées.

Il existe des algorithmes plus complexes qui évitent ce défaut en cherchant la bonne chaîne augmentante à chaque itération. ...

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
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 susceptible 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).

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 augment 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 (seul 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 commencait d'un flot aléatoire, il ne serait pas garanti qu'on arrive á un flot optimal;
mais, en commençant au flot nul, oui.