Des couplages aux flots

Michel Habib,
habib@irif.fr
http://www.irif.fr/~habib

22 décembre 2017

Plan

Un autre paradigme algorithmique : amélioration par échanges

Prenons l'exemple du voyageur de commerce et son heuristique classique construite en deux étapes.

1. Construire une solution gloutonnement (par exemple on part d'un point et on prend toujours le plus proche voisin)
2. Amélioration de la solution (par exemple à l'aide de 2-échanges)
3. La solution obtenue n'est pas trop mauvaise.

Comme le problème du voyageur de commerce est NP-complet, l'heuristique précédente ne peut pas fournir l'optimum dans tous les cas. Pour s'en convaincre il suffit de construire une instance pour laquelle à partir du premier glouton il faut faire un 3-échange pour obtenir l'optimum.

La même idée s'applique pour tout procédé basé sur des k-échanges (avec k fixé).

Questions naturelles

1. Quand cette idée d'amélioration par échange donne-t-elle une solution optimale ?
2. Peut-on évaluer la distance à l'optimum ?

Couplages dans les graphes bipartis

Pour un graphe biparti $G = (X, Y, E)$,
$C \subseteq E$ est un couplage s'il est constitué d'arêtes indépendantes (i.e., $\forall e, f \in C$, $e \cap f = \emptyset$)
Le problème intéressant est celui de la recherche d'un couplage de cardinal maximum de G.

Aspects matroïdaux

$M = (E, \mathcal{C})$ avec $\mathcal{C} = \{\mathcal{C}$ couplage de $G\}$ est l'ensemble des indépendants communs à deux matroïdes de partition. Les indépendants de taille maximum étant les couplages de cardinal maximum.

- Les éléments communs à deux matroïdes ne forment pas toujours un matroïde.
- Il suffit de prendre l'exemple des couplages. Ainsi le graphe biparti P_4, admet deux couplages maximaux pour l'inclusion de taille 1 ou 2. Les indépendants maximaux n'ont donc pas tous le même cardinal !
- Cependant il existe quand même une structure d'échange entre couplages, le long de chaînes alternées.

Théorème

Pour un graphe G, $M \subseteq E(G)$ est un couplage de taille maximum ss'il n'existe pas de chaîne alternée impaire joignant deux sommets insaturés par M.

Preuve

Nous remarquons que si un couplage M admet une chaîne alternée impaire joignant deux sommets insaturés, en faisant l'échange des arêtes sur cette chaîne on augmente la taille du couplage de 1.
La condition est donc nécessaire.
Considérons deux couplage M_0 optimal et M qui vérifie la condition du théorème.

Preuve (suite)

Considérons les composantes connexes du sous-graphe partiel $G(M_0 \Delta M)$.
Ce sont nécessairement des chaînes paires ou des cycles pairs.
Donc $|M| = |M_0|$.

Ce théorème permet d'écrire un algorithme polynomial.

$M \leftarrow \emptyset$;

tant que *Existe une chaîne alternée impaire entre deux sommets insaturés* **faire**

Procéder à l'échange sur la chaîne impaire

fin

- Quel est la meilleure complexité pour un tel algorithme ?
- Cette idée se généralise au problème de l'intersection de deux matroïdes.
- Le problème reste polynomial si le graphe est valué et que l'on cherche le couplage de poids maximum.

Ajout d'une valuation dans un problème d'optimisation

- Dans certains cas le problème devient NP-complet.
- Dans le cadre du couplage, il faut maintenant considérer des échanges sur des chaînes alternées impaires mais aussi sur des chaînes et cycles alternés pairs.

Le cas biparti

Déterminer les chaînes alternées se fait par un parcours en largeur. Ce qui implique un algorithme en $O(n(n+m)) = O(nm)$.

Un très intéressant théorème Min-Max

Théorème Edmonds 1970

Soient $M_1 = (E, \mathcal{I})$ et $M_2 = (E, \mathcal{J})$ deux matroïdes sur E ayant pour fonction de rang r_1, r_2,
alors la taille maximum d'un indépendant commun aux deux matroïdes est :
$min_{A \subseteq E}(r_1(A) + r_2(E\text{-}A))$

Preuve :

Soit $I \in \mathcal{I} \cap \mathcal{J}$,
$\forall A \subseteq E$, $|I| = |A \cap I| + |I - A| \leq r_1(A) + r_2(E\text{-}A)$
L'autre inégalité se montre soit par induction, soit algorithmiquement.

Algorithme pour l'intersection de deux matroïdes

Soient $M_1 = (E, \mathcal{I})$ et $M_2 = (E, \mathcal{J})$ deux matroïdes sur E
Soit $I \in \mathcal{I} \cap \mathcal{J}$, on construit un graphe orienté $G_{M_1 M_2}(I)$ sur E tel que :
$\forall y \in I$, and $\forall x \in E$-I
yx arc ssi $I - y + x \in \mathcal{I}$
xy arc ssi $I - y + x \in \mathcal{J}$
$X_1 = \{x \in E - I | I + x \in \mathcal{I}\}$
$X_2 = \{x \in E - I | I + x \in \mathcal{J}\}$

Propriété

I est de taille maximum ss'il n'existe pas de chemin de X_1 à X_2 dans G.

Preuve

S'il existe $a \in X_1 \cap X_2$, alors $I \cup \{a\} \in \mathcal{I} \cap \mathcal{J}$. On peut donc supposer $X_1 \cap X_2 = \emptyset$.
S'il existe un chemin $[x, y, z]$ de longueur 2, allant de X_1 à X_2, nous avons en suivant ce chemin :
$I + x \in \mathcal{I}$ et $I + x - y \in \mathcal{J}$ puis $I + z - y \in \mathcal{I}$ et enfin $I + z \in \mathcal{J}$.
D'après $I + x \in \mathcal{I}$ et $I + z - y \in \mathcal{I}$ par l'axiome des indépendants nous avons $I + x + z - y \in \mathcal{I}$.
D'après $I + z \in \mathcal{J}$ et $I + x - y \in \mathcal{J}$ par l'axiome des indépendants nous avons $I + x + z - y \in \mathcal{J}$.
Donc I n'était pas de taille maximum, car $I + x + z - y \in \mathcal{I} \cap \mathcal{J}$

Si le chemin n'est pas de longueur 2, il est nécessairement de longueur paire (vu la définition des arcs du graphe) et le raisonnement ci-dessus se généralise simplement.
Dans l'autre sens, s'il n'existe pas de chemin de X_1 à X_2 dans G, on considère un ensemble A contenant X_2 vérifiant $A \cap X_1 = \emptyset$ et tel qu'il existe aucun arc entrant dans A (il existe nécessairement un tel ensemble A).
Montrons $r_1(A) \leq |I \cap A|$. Sinon, il existe (cf. axiome sur les indépendants) $x \in A - I$ tel que $I \cap A + x \in \mathcal{I}$.
Cependant $I + x \notin \mathcal{I}$, car $A \cap X_1 = \emptyset$. Comme $I \cap A + x$ est nécessairement inclus dans un indépendant de taille max sur l'ensemble $I + x$, il existe donc $y \in I - A$ tel que $I - y + x \in \mathcal{I}$, ce qui est impossible car alors il existerait un arc entrant en A.

De même on obtient $r_2(E - A) \leq |I - A|$.
Ainsi $r_1(A) + r_2(E - A) \leq |I|$ et I est donc optimal.
En fait cela fournit aussi une preuve algorithmique de l'autre inégalité de théorème d'Edmonds.

Limites sur les couplages

Couplage de poids maximum

Chaque arête est pondérée par un réel positif, il existe un algorithme polynomial pour la recherche d'un couplage de poids maximum.

Couplage Maximum pour l'inclusion et de cardinal minimum

La recherche d'un tel couplage est un problème NP-difficile.

Dénombrement

Trouver le nombre de couplages parfaits d'un graphe biparti est #P-complet.

- Un couplage M est maximal pour l'inclusion si pour $\forall e = (u, v) \in E(G)$, soit u ou v est adjacent à une arête de M.
- Cette condition peut se construire et vérifier localement en $O(|E(G)|)$.
- D'ailleurs un couplage maximal est une 2-approximation d'un couplage maximum. En effet le pire cas est celui dans lequel chaque arête de M est au centre d'une chaîne alternée de longueur 3. Dans ce cas il existe un couplage optimal ayant un cardinal $2.|M|$.
- Cette borne est atteinte il suffit de considérer un biparti constitué de k N disjoints.

Si le graphe n'est pas biparti, les notions de couplages (resp couplages de cardinal maximum) se définissent naturellement. Mais les algorithmes sont plus compliqués à mettre en œuvre, mais restent polynomiaux.

Définitions

On considère un graphe orienté $G = (X, U)$, deux sommets $s, t \in X$. Les arcs sont munis de capacités.
$c : U \rightarrow R+$
un flot sur G est un vecteur indexé par U (donc de taille m) qui vérifie :
$\forall xy \in U$, $0 \leq \phi(x, y) \leq c(x, y)$.
$\forall x \in X$, $\Sigma_{yx\ entrant\ en\ x}\phi(x, y) = \Sigma_{xz\ sortant\ de\ x}\phi(x, z)$
Lois de Kirchoff de conservation aux sommets.
Etant donné deux sommets particuliers s, t de G, on cherche un flot qui maximise le flot sur l'arc ts.

On cherche un flot ϕ qui vérifie :

1. $\forall xy \in U$, $0 \leq \phi(x,y) \leq c(x,y)$.
2. $\forall x \in X$, $\Sigma_{yx \text{ entrant en } x}\phi(x,y) = \Sigma_{xz \text{ sortant de } x}\phi(x,z)$
3. Et maximise $\phi(t,s)$

Soit A la matrice d'incidence sommets-arcs, i.e. ; pour un arc $u = ij$ $A[i, u] = 1$, $A[j, u] = -1$ et $A[k, u] = 0$ pour $k \neq i, j$.
La loi de Kirchoff de conservation peut s'écrire pour un flot $\phi \in R^m$ $A.\phi = 0$.

On cherche un flot $\phi \in R^m$ qui vérifie :

1. $0 \leq \phi \leq c$.
2. $A.\phi = 0$.
3. Et maximise $\phi(t, s)$

Autrement dit :

Le problème du flot maximum dans un graphe peut se modéliser comme un problème de programmation linéaire en nombres réels.

On considère un graphe orienté $G = (X, U)$, deux sommets $s, t \in X$. Les arcs sont munis de capacités.
$c : U \rightarrow R+$

Prototype de théorème Min-Max

La valeur minimale d'une coupe séparant s et t est égale à la valeur du flot maximum entre s et t dans G

Recherche d'une chaîne améliorante

RechercheChaîne(G, ϕ);
Données: un graphe orienté $G = (X, U)$, une fonction de capacité $c : U \to R+$, deux sommets $s, t \in X$, un flot $\phi \geq 0$
Résultat: soit une chaîne améliorante de s à t, soit la maximalité de ϕ
$OUVERTS \leftarrow \{s\}$; $FERMES \leftarrow \emptyset$;
tant que $OUVERTS \neq \emptyset$ **faire**
  $x \leftarrow Choix(OUVERTS)$;
  Explorer(x);
  $Ajout(x, FERMES)$; $Retrait(z, OUVERTS)$;
fin
"Le flot ϕ est maximum" ;

```
Explorer(x);
pour Tous les successeurs y de x faire
    si y ∉ FERMES ∪ OUVERTS et φ(xy) < c(yx) alors
        si y = t alors
            STOP : "On a trouvé une chaîne améliorante de s à t"
        sinon
            Ajout(y, OUVERTS)
        fin
    fin
fin
pour Tous les prédécesseurs y de x faire
    si y ∉ FERMES ∪ OUVERTS et φ(yx) > 0 alors
        si y = t alors
            STOP : "On a trouvé une chaîne améliorante de s à t"
        sinon
            Ajout(y, OUVERTS)
        fin
    fin
```

Algorithme de Ford-Fulkerson

Afin de calculer un flot maximum dans un graphe, on peut appliquer la procédure suivante :

$\phi \rightarrow 0$;
tant que *RechercheChaîne*(G, ϕ)=*Vrai* **faire**
    Améliorer le flot sur la chaîne ;
    mise à jour de ϕ ;
fin

1. La recherche d'une chaîne améliorante peut se calculer en $O(n+m)$ à l'aide d'un parcours de graphe.
2. Lorsque les capacités sont entières l'algorithme précédent augmente à chaque itération $\phi(ts)$ d'au moins une unité de flot.
3. Soit $C = \Sigma_{u \in U} c(u)$. Alors $\phi(ts) \leq C$.

Une coupe S est un sous-ensemble de sommets du graphe tel que $s \in S$ et $t \notin S$. La capacité $c(S)$ d'une coupe est la somme des capacités des arcs sortants de S.

Lemme

Pour tout flot ϕ et toute coupe S :
$\phi(t,s) + \Sigma_{xy \neq ts \ entrant \ en \ S} \phi(x,y) = \Sigma_{ab \ sortant \ de \ S} \phi(a,b) \leq c(S)$
d'où $\phi(t,s) \leq c(S)$.

corollaire

Si l'on trouve un flot ϕ et une coupe S tels que $\phi(t,s) = c(S)$, ils sont optimaux.

Fin de la preuve du Théorème FF

Preuve

Il est facile de vérifier que chaque chaîne améliorante permet de construire à partir de ϕ un nouveau flot ϕ' : qui vérifie $\phi(t,s) < \phi'(t,s)$.
Lorsqu'il n'existe plus de chaîne améliorante pour un flot ϕ, soit S l'ensemble des sommets explorés à partir de s. Nécessairement (vu la procédure d'exploration) les arcs rentrant en S on un flot nul et les arcs sortants sont saturés (valeur de flot $=$ capacité de l'arc). L'inégalité du lemme est donc une égalité et le flot est donc bien optimum.

1. En conclusion lorsque les capacités sont entières, l'algorithme de Ford - Fulkerson converge vers un flot maximal entier et se termine en au plus $O(C(n+m))$.
2. Nous verrons sur un exemple que cette borne de complexité est atteignable.
3. Ford et Fulkerson avaient remarqué qu'avec des capacités réelles, l'algorithme pouvait converger vers un flot non maximal.

corollaire

A partir d'un flot maximum, on peut calculer en $O(m)$ une coupe minimale.

Remarque

Sans information supplémentaire sur le parcours de graphe utilisé, (par exemple un parcours générique) l'algorithme de FF n'est pas polynomial.

Graphe des écarts ou graphe résiduel

Pour un flot ϕ donné, on construit un graphe $G_\phi = (X, V)$ muni d'une valuation $\omega : V \to R+$ comme suit :
si $xy \in U$ alors :

- si $\phi(x, y) < c(x, y)$ alors $xy \in V$ et $w(x, y) = c(x, y) - \phi(x, y)$
- si $\phi(x, y) > 0$ alors $yx \in V$ et $w(y, x) = \phi(x, y)$

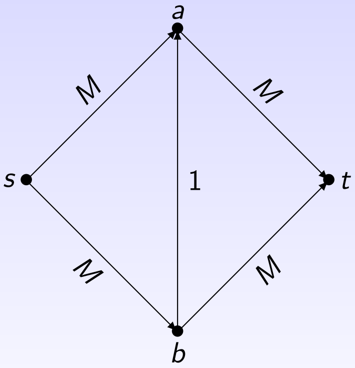
Propriété

Il existe un chemin de s à t dans G_ϕ ss'il existe une chaîne améliorante dans G pour ϕ

Nous sommes donc ramenés à un problème d'existence de chemin dans un graphe.
On peut calculer :

1. Un plus court en nombre d'arcs
2. Celui qui augmente le plus le flot
3. ...

Un mauvais cas



Si l'on utilise les chaînes améliorantes s, b, a, t puis s, a, b, t ...

Il faut alors $2M$ chaînes améliorantes pour obtenir le flot maximum

Non polynomial

M n'est pas polynomial en la taille d'un codage de la donnée.
Donc tel quel l'algorithme de Ford-Fulkerson vu précédemment n'est pas polynomial !
En outre sur cet exemple à quatre sommets, nous avons vu que la recherche de la chaîne améliorante la plus longue ne donne pas un algorithme polynomial.

On peut étendre l'exemple

$\forall u,\ c(u) = 1$

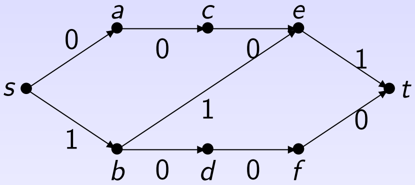


FIGURE: Ce flot de valeur 1 a été obtenu à l'aide de la plus courte chaîne améliorante :$[s, b, e, t]$. Pour l'améliorer par l'algorithme FF il faut utiliser la chaîne $[s, a, c, e, b, d, f, t]$. C'est la seule chaîne améliorante à cette étape.

Cependant il n'est pas possible d'étendre cet exemple en un exemple non polynomial lorsqu'on utilise à chaque étape une plus courte chaîne améliorante.
En effet si $\forall u \neq be, c(u) = M$ et $c(be) = 1$, une deuxième plus courte chaîne améliorante est $[s, b, d, f, t]$ (valeur $M - 1$), la troisième est $[s, a, c, e, t]$ (valeur $M - 1$).
L'algorithme se terminant par la chaîne améliorante $[s, a, c, e, b, d, f, t]$ (valeur 1).

Ces exemples ont justifié la conception d'autres algorithmes de flots, en particulier ceux basés sur la notion de préflot.

Un algorithme polynomial simple

Soit Δ la puissance de 2 maximale telle que $\Delta \leq max_{sx \in U}(c(s,x))$.
On notera $G_\phi(\Delta)$ le graphe partiel résiduel, constitué des arcs de capacités $\geq \Delta$.

$\phi \leftarrow 0$;
tant que $\Delta \geq 1$ **faire**
  tant que *il existe un chemin de s à t dans* $G_\phi(\Delta)$ **faire**
    améliorer ϕ;
    Mise à jour de $G_\phi(\Delta)$
  fin
  $\Delta \leftarrow \Delta/2$
fin

Théorème

L'algorithme précédent calcul un flot maximum entre s et t en $O(m^2 log_2 C)$, avec C la capacité maximale d'un arc sortant de s.

Preuve

1. Le nombre de passages dans la première boucle Tant que est borné par $log_2 C$.
2. Chaque chemin trouvé dans $G_\phi(\Delta)$ permet d'augmenter le flot d'au moins Δ.
3. A la fin de la 2ème boucle Tant que pour une valeur de Δ.
   Reprenons l'égalité du lemme sur la dernière coupe :
   $\phi(t,s) = \Sigma_{ab \text{ sortant de } S}\phi(a,b) - \Sigma_{xy \neq ts \text{ entrant en } S}\phi(x,y)$
   Vu le marquage :
   Pour un arc ab sortant de S, $c(ab) < \phi(ab) + \Delta$ et pour un arc entrant xy, $\phi(xy) < \Delta$.
   d'où $\phi(t,s) \geq c(S) - m\Delta$ (majoration grossière), ou encore $c(S) \leq \phi(t,s) + m\Delta$.

Preuve (suite)

Soit $\phi_p(t, s)$ la valeur du flot obtenu à l'étape Δ.
$\phi^*(t, s) \leq c(S) \leq \phi(t, s) + m\Delta$. (Où $\phi^*(t, s)$ représente l'optimum)
Donc à la phase suivante avec la valeur $\Delta/2$, le flot ne peut donc augmenter que de $m\Delta$, et il y a au plus $2m$ améliorations du flot, à l'étape $\Delta/2$.
Donc au total pour tour tout l'algorithme, on calcule au plus $m\Delta$ chaînes améliorantes.
La recherche d'une chaîne améliorante nécessite un parcours en $O(m)$, d'où le résultat :
$O(m^2 log_2 C)$.

Notons que cet algorithme est polynomial car les capacités des arcs se codent en binaire, et donc $O(m^2 log_2 C)$ est polynomial en fonction de la donnée.

On appelle flot normal un flot dont tous les circuit positifs utilisent l'arc ts.

1. Montrer qu'à tout flot ϕ on peut associer un flot ψ normal tel que $\psi(t,s) = \phi(t,s)$.
2. Montrer que l'algorithme de Ford-Fulkerson ne manipule que des flots normaux.
3. En déduire un théorème de décomposition des flots obtenus par l'algorithme de Ford-Fulkerson.

Lemme de Minty 1960

On considère un graphe orienté G et xy un arc de G. Les arcs de G sont colorés à l'aide de 3 couleurs Bleu, Rouge et Noir (exactement une couleur par arc). On supposera xy coloré en Noir. Une et une seule des deux propriétés suivantes est vraie :

- soit il existe un cycle Rouge - Noir passant par xy, dont tous les arcs noirs sont dans le même sens que xy.
- soit il existe un cocyle Bleu - Noir contenant xy, dont tous les arcs noirs sont dans le même sens que xy.

Preuve

On considère un marquage (un parcours de graphe à partir de y)
Lors de l'exploration d'un sommet z
si l'arc zt est Noir on ouvre t
si l'arc zt ou tz est Rouge on ouvre t

1. Soit on arrive à marquer le sommet x, alors on a trouvé un tel cycle qui ne contient que des arcs Noirs dans le même sens que xy et éventuellement des arcs Rouges dans les deux sens.
2. Soit le marquage s'arrête sans avoir rencontré x, mais alors le cocycle de l'ensemble des sommets marqués à partir de y ne contient que des arcs Bleus et tous les arcs Noirs sont entrants comme l'arc xy.

Cela ressemble à la dualité cycle-cocycle déjà vue lors de l'étude des arbres recouvrants de poids mininaux.

Exercices

- En déduire que dans un graphe orienté, tout arc appartient soit à un circuit élémentaire, soit à un cocircuit élémentaire.
- Il suffit de ne considérer que les arcs noirs.
- Peut-on en déduire le théorème de Ford-Fulkerson (égalité entre coupe min et flot max) ?
- l'arc *xy*= l'arc *ts*
  colorer u en noir (resp. rouge, bleu) si $\phi(u) = 0$ (resp. si $0 < \phi(u) < c(u)$, si $\phi(u) = c(u)$).

Le lemme de Minty contient la combinatoire de l'algorithme de Ford - Fulkerson,
il a servi de base à la définition des matroïdes orientés.

Quelques applications des flots

1. Calculs de Couplages de cardinal maximaux
2. Capacités sur les noeuds (on se ramène à un problème de flot en dédoublant les sommets).
3. Théorèmes de Menger (conséquences directes du théorème de Ford-Fulkerson)
4. Flots compatibles (on peut se ramener à un problème de flot maximum, dans un graphe construit à partir du premier).

Les théorèmes de Menger

Les deux versions orientées

Pour $a, b \in G$, avec $ab \notin U(G)$, le nombre maximum de chemins arc-disjoints (resp sommet-disjoints) joignant a et b est égal au nombre minimum d'arcs (resp. sommets) qui permettent de disconnecter a de b.

Les deux versions non-orientées

Pour $a, b \in G$, avec $ab \notin U(G)$, le nombre maximum de chaînes arêtes-disjointes (resp sommet-disjointes) joignant a et b est égal au nombre minimum d'arêtes (resp. sommets) qui permettent de disconnecter a de b.

Une application récente et importante

Les calculs de flots max sont une étape importante du calcul en imagerie
Afin de calculer une coupe min
Il s'agit de flots sur le graphe d'adjacence des pixels de l'image
Très nombreuses publications en imagerie à ce sujet.
Algorithmes adaptés aux matrices de pixels.

Propriété

La capacité d'une coupe est une fonction submodulaire :
$\forall A, B \subseteq X - \{t\}$ avec $s \in A \cap B$
$c(A \cap B) + c(A \cup B) \leq c(A) + c(B)$

Preuve

Une étude exhaustive de tous les cas possibles montre que la seule différence provient des capacités des arcs de $B - A$ vers $A - B$ ainsi que celles des arcs allant $A - B$ vers $B - A$.
Ces capacités sont présentes dans le terme de gauche, les capacités étant positives ou nulles, d'où la conclusion.

- Les fonctions submodulaires sont très importantes en optimisation combinatoire, elles sont souvent associées à des algorithmes polynomiaux.
- Les fonctions de rang de matroïdes sont des fonctions submodulaires.
- La submodularité est aussi appelée convexité discrete (i.e. sur des ensembles).

Considérons une fonction $\theta : X \rightarrow R+$.
Ceci nous permet de définir pour un vecteur ψ_θ de R^m comme suit :
$\forall xy \in U$, $\psi_\theta(xy) = \theta(y) - \theta(x)$

Théorème

$\forall\theta$ and $\forall\phi$ flot sur G, nous avons : ${}^t\psi_\theta.\phi = 0$

Preuve

$\forall\theta$ and $\forall\phi$ flot sur G, nous avons :
${}^t\psi_\theta\phi = \Sigma_{xy \in U}\ \phi(xy).(\theta(y) - \theta(x))$
Si A est la matrice d'incidence sommets-arcs, il suffit de remarquer que :
${}^t\psi_\theta.\phi = \Sigma_{x \in V(G)}\theta(x)(\Sigma_{yx\ entrant\ en\ x}\phi(x,y) - \Sigma_{xz\ sortant\ de\ x}\phi(x,z)) = T.A.\phi$
où T est le vecteur à n composantes tel que $T_i = \theta(x_i)$
Et comme $A\phi = 0$ (loi de Kirchoff) ceci donne le résultat.

Application

Soit G le réseau routier de France, et ϕ le flot des véhicules circulant à un instant ϵ.
On considère θ la fonction de température au même instant en chaque sommet du réseau.
$\forall \epsilon \ {}^t\psi_\theta.\phi = 0$.

Certificats pour un flot ϕ

- Vérification des conservations aux nœuds : $O(n+m)$
- Vérification de la maximalité de $\phi(s,t)$:
  À l'aide du calcul d'une coupe minimale.
  On peut procéder à un parcours sur (G,ϕ) en $O(n+m)$

Certificats pour une coupe minimale S

Il faudrait calculer un flot maximum, ce qui ne se fait pas en un simple parcours !

Malgré le théorème de Ford-Fulkerson flot max et coupe min ne sont pas algorithmiquement équivalents.

Pour le calcul d'un flot max :

1. Algorithme avec seuil décroissants en $O(m^2 log_2 C)$.
2. Ford-Fulkerson + plus courtes chaînes améliorantes en $O(m^2 n)$.

Lemme 1

Soit $D = (X, U)$ un multigraphe fini orienté vérifiant :

1. $\forall x \neq s, t \; d^-(x) = d^+(x) > 0$
2. $d^+(s) - d^-(s) = k > 0$

Alors D contient k chemins arc-disjoints de s à t.

Remarque

Pas de condition sur les degrés de t, ni sur la connexité !

- Remarquons que par effet de bord si le graphe est fini, les conditions impliquent : $d^-(t) - d^+(t) \geq k$.
- Il suffit d'utiliser l'égalité : $\Sigma_{x \in G} d^+(x) = \Sigma_{x \in G} d^-(x)$
- qui donne après simplification $d^+(s) + d^+(t) = d^-(s) + d^-(t)$
- d'où : $d^-(t) - d^+(t) = d^+(s) - d^-(s) \geq k$

Preuve

On construit un chemin en partant de s en suivant les arcs de D aléatoirement, mais en ne prenant un arc qu'une fois. Ce chemin peut passer au plus $d^{-}(s)$ fois en s, mais à chaque fois il peut ressortir en utilisant un arc non déjà parcouru.
Comme le graphe est fini et que l'on est jamais bloqué en un sommet, nécessairement le chemin va arriver en t.
On vire ce chemin et l'on procède par induction sur k.

Remarque

Le lemme est faux pour un graphe infini, il suffit de considérer un graphe constitué de 2 chemins infinis démarrant en s, auquel on ajoute un arc ts arrivant en s. L'exemple vérifie bien les conditions du lemme mais il n'existe pas de chemin de s à t.

Lemme 2

Considérons une suite de flots $f_1, \ldots$ telle que f_{i+1} s'obtient à partir de f_i en augmentant sur une chaîne améliorante de longueur minimale P_i, alors :

1. $\forall k, |P_k| \leq |P_{k+1}|$
2. $|P_k| + 2 \leq |P_l|$, $\forall l$ tel que $P_k \cup P_l$ contienne une paire d'arcs inverses

Preuve

On considère le graphe D construit sur $P_k \cup P_{k+1}$ en orientant les arcs de s vers t et en ôtant les arcs utilisés dans les deux sens (par contre D peut contenir des arcs parallèles).
D est construit à partir d'arcs du graphe G initial. On peut appliquer le lemme 1 précédent sur D avec $k = 2$. Il existe donc deux chemins arc-disjoints Q_1 et Q_2 allant de s à t, qui sont aussi des f_k chaînes améliorantes.
Par définition : $|P_k| \leq |Q_1|$ et $|P_k| \leq |Q_2|$.
Donc $2|P_k| \leq |Q_1| + |Q_2| \leq |P_k| + |P_{k+1}|$
D'où $|P_k| \leq |P_{k+1}|$.

Preuve (suite)

Pour la deuxième partie du lemme, il suffit de considérer deux chaînes ayant deux arcs inverses et d'appliquer le même raisonnement en remarquant que l'on gagne 2 dans les inégalités.

Théorème

L'algorithme de Ford-Fulkerson (version plus courtes chaînes) calcule un flot maximum en $O(m^2 n)$

Preuve

Chaque chaîne améliorante sature (resp. met à 0) au moins un arc. Appelons cet arc goulot d'étranglement. Un arc de G est goulot d'étranglement au plus $n/2$ fois (cf. lemme 2.2).
Il y a donc au plus $m.n/2$ chaînes améliorantes.
Si on utilise un BFS pour les calculer, la complexité de l'algorithme est $O(m^2 n)$.

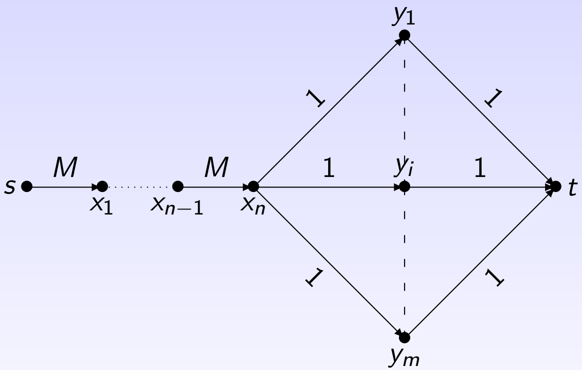
Le théorème précédent n'utilise pas d'hypothèses particulières sur les capacités.

Corollaire

Si le graphe est fini et les capacités positives alors l'algorithme de Ford-Fulkerson (version plus courtes chaînes) calcule un flot maximum.

La majoration du résultat précédent est assez grossière, et en pratique l'algorithme converge très rapidement.
Il est facile de construire une famille infinie d'exemples sur lequel Ford-Fulkerson nécessite $O(m.n)$ opérations.

Un autre mauvais cas



Tous les algorithmes à base de chaînes améliorantes vont utiliser au moins m chaînes, soit $\Omega(n(n+m))$.

Il est facile de généraliser cet exemple en un graphe dénombrable sur lequel Ford-Fulkerson ne trouve jamais le flot maximum.

Un peu d'histoire

- Une première version a été proposée par Y. Dinitz en union soviétique en 1969.
- Elle a été introduite en occident par Shimon Even et Alon Itai qui l'ont un peu modifiée et attribuée à un certain Dinic.
- La version présentée ci-après est une variante facile à comprendre de l'algorithme original.

Notion de préflot

On définit pour chaque $f \in R^m$:

- $\forall x \in X,\ f_{in}(x) = \sum_{yx \in E} f(yx)$
- $\forall x \in X,\ f_{out}(x) = \sum_{\{xz\} \in E} f(xz)$
- $exces(x) = f_{in}(x) - f_{out}(x)$

Définition

Un préflot sur $G = (X, U)$ avec deux sommets s, t.
$f \in R^m$ tel que :
$\forall u \in U$, $0 \leq f(u) \leq c(u)$
$\forall x \neq s,\ exces(x) \geq 0$.
Remarque : si pour un préflot f, $\forall x \neq s, t,\ exces(x) = 0$ alors f est un flot.

Sommets Actifs, étiquetage de distance

$\delta^+(x) = \{xy \in U\} =$ l'ensemble des arcs sortant de x.
Un sommet x tel que $exces(x) > 0$ est appelé actif.
Un étiquetage de distance pour un flot f est une fonction :
$\psi : X \rightarrow N$, telle que $\psi(t) = 0$, $\psi(s) = n$, et $\forall xy \in G_f$,
$\psi(x) \leq \psi(y) + 1$.

Data: un graphe $G = (X, U, s, t)$;
Result: Un flot de valeur max f compatible avec G;
"Initialisations :"
$a(s) = n$ **foreach** $x \in X \backslash \{s\}$ **do**
    $a(x) = 0$
end
foreach $u \in \delta^+(s)$ **do**
    $f(u) = c(u)$
end
foreach $u \notin \delta^+(s)$ **do**
    $f(u) = 0$
end
Calcul du graphe des écarts G_f et des sommets actifs.
%On a saturé tous les arcs sortants de s.%

%Maintenant le corps de la procédure %
while $\exists x \in X$ *avec* $exces(x) > 0$ **do**
    Choisir $x \in X - \{s, t\}$ tq $exces(x) > 0$ et $a(x)$ est maximale parmi les sommets actifs
    if $\exists y \in X$, *tq* $xy \in U_f$ *avec* $a(y) < a(x)$ **then**
        push(x,y)
    end
    else
        relabel(x)
    end
end

% Les deux fonctions push et relabel %

push(x, y) :

if *xy est un arc retour de G_f;*

then

  $\delta = min(f(xy), exces(x))$;

  $f(xy) \leftarrow f(xy) - \delta$;

end

else

  $\delta = min(c(xy) - f(x, y), exces(x))$;

  $f(xy) \leftarrow f(xy) + \delta$;

end

Mettre à jour : $exces(x)$, $exces(y)$, le graphe des écarts $G_f = (X, U_f)$;

relabel (x) : $a(x) \leftarrow a(x) + 1$

Lemme

Si a est un étiquetage de distance pour f, il n'existe pas de chemin de s à t dans le graphe résiduel G_f.

Preuve

Considérons un chemin $\mu = [s, x_1, \ldots, x_{k-1}, t]$ de longueur $k < n$.
En considérant la fonction a le long du chemin on obtient :
$n = a(s) \leq a(x_1) + 1 \leq a(x_2) + 2 \cdots \leq a(t) + k = k$ d'où la contradiction : $n \leq k < n$.

Invariant : f est un préflot sur G et a est un étiquetage de distance pour f.
La propriété de préflot est maintenue par l'opération push (la seule où l'on change la valeur du flot).
L'opération relabel, quand elle est réalisée ne peut pas violer la condition sur les étiquetage de distances.
En outre, en utilisant les opérations push et relabel on réduit nécessairement les sommets en excès de flot. Et donc à la fin l'algorithme s'arrête, il n'y a plus de sommet strictement en excès et nous avons donc un flot et vu le lemme précédent, nous avons un flot maximum.

Améliorations possibles

relabel on peut augmenter plus que de $+1$
et aller directement à la première valeur qui permettra de réduire l'excès en x.
Nombreuses études sur l'optimisation de cet algorithme (en particulier sur les structures de données associées).

Complexité

1. Si $exces(x) > 0$, il existe un chemin de s à x.
2. $\forall x \in X$, $a(x) \leq 2n - 1$
3. Chaque sommet est renuméroté (par la fonction label) au plus $2n - 1$ fois. Donc pour l'ensemble des sommets au plus $2n^2$ appels à la fonction label.
4. Le nombre d'opérations push saturantes est au plus de 2nm. En effet pour un arc donné xy, s'il est saturé à une étape nécessairement nous avions $a(y) < a(x)$, dont pour le saturer une autre fois il faut d'abord le désaturer mais dans ce cas il faut $a(x) < a(y)$ ce qui implique au moins deux opérations de renumérotage de y. Or un sommet est au plus renuméroté $2n - 1$ fois.
   Donc un arc donné peut être saturé au plus $2n$ fois.

lemme

Les autres opérations (i.e., non saturantes) push sont au plus en nombre $2n^2m$.

Preuve :

C'est le point difficile, car on peut faire plusieurs augmentations successives (push) le long d'un arc xy sans jamais le saturer.
Pour évaluer ces opérations on utilise une fonction de potentiel :
$\phi(f, a) = \Sigma_{x \neq s, excess(x) > 0} a(x)$.
A la fin de l'algorithme, il n'y a plus de sommet actif et donc $\phi(f, a) = 0$. Au démarrage, les sommets y en excès (voisins de s vérifient $a(y) = 0$). Donc la fonction vaut aussi 0 au démarrage.
Elle est va donc de 0 à 0, elle est par définition toujours positive au cours de l'algorithme (car les $a(x)$ sont positifs).

Preuve suite :

Cette fonction diminue d'au moins 1 à chaque opération push non saturante. Car à la fin de l'opération un sommet x a perdu son excès et le sommet y qui peut avoir obtenu un excès dans l'opération, vérifie $a(y) \leq a(x) - 1$. Dans le pire cas on a remplacé dans la formule $a(x)$ par $a(y)$ d'où le résultat.
Un renumérotage augmente la fonction potentiel d'exactement 1. Il y a au plus $2n^2$ renumérotages, ainsi la fonction potentielle augmente au plus de $2n^2$ par les renumérotages.
Un push saturant ne change pas les $a(x)$, mais peut mettre en excès un sommet y. Ce qui ajoute au plus $a(y) \leq 2n - 1$ à la fonction potentielle.
Comme il y a au plus $2nm$ push saturant la fonction potentielle peut augmenter en tout d'au plus $2nm(2n-1) + 2n^2 = 4mn^2$.
D'où la preuve du lemme.

- L'algorithme des préflots de Dinitz fonctionne en $O(n^2m)$ dans le plus mauvais cas.
- Cet algorithme ressemble à un parcours de graphe et de nombreuses variantes ont été étudiées.
- On peut améliorer la complexité avec la variante de : Golberg, Tarjan (1988) en $O(n^2m^{1/2})$

Ces estimations du temps de calcul sont peut-être un peu grossières mais toutefois calculer un flot maximal ne se fait pas en temps linéaire.
Ainsi il n'existe pas d'algorithme capable de traiter des problèmes de flots sur des très grands graphes.
Il serait donc intéressant de disposer d'algorithmes d'approximation pour les flots maximaux . . .