Techniques d'Optimisation
Chapitre 2: Problème de flôt

Dr TARI Abdelkamel

Mars 2014

Sommaire

- Modélisation du problème de flôt
- Problème de la coupe minimale
- Chaîne augmentante
- Algorithme de Ford-Fulkerson
- Exemple

Modélisation du problème (1/3)

- Soient un réseau R=(X, U, c) un graphe fini orienté, connexe et sans boucles, et c une application: U$\rightarrow$R

$$u \rightarrow c(u)=c_u$$

- Il s'agit d'envoyer un flôt (une matière) d'une source s ($\Gamma^-(s)=\varnothing$) et de récupérer son maximum au sommet puits p ($\Gamma^+(p)=\varnothing$) tout en conservant la matière en chaque sommet de ce réseau (Kirchoff)

- Un flôt f est un vecteur de R^m vérifiant:

$$\sum_{x \in w^+(x)} f(u) = \sum_{x \in w^-(x)} f(u) \qquad \forall x \in X$$

$w^+(x)$: Cocycle engendré par les arcs sortants du sommet x

$w^-(x)$: Cocycle engendré par les arcs entrants vers le sommet x

Modélisation du problème (2/3)

On crée l'arc u_r =(p, s) fictif tel que $c(u_r)=+\infty$ pour avoir une conservation totale sur le réseau.

f est dit réalisable s'il vérifie: $0 \leq f(u) \leq c(u)\ \forall\ u \in U \cup \{u_r\}$

•Un flot f est dit compatible sur R (X, U, a, b) s'il vérifie :

$$\forall\ u \in U \cup \{u_r\},\ a(u) \leq f(u) \leq b(u).$$

Le problème du flot maximum se formule comme suit:

$$\begin{cases} Trouver\ f \in R^m \\ \sum_{u \in w^+(x)} f(u) = \sum_{u \in w^-(x)} f(u)\ \forall x \in X \\ 0 \leq f(u) \leq c(u) \\ f(u_r) = Z\ (Max) \end{cases}$$

Modélisation du problème (3/3)

- Soit x_{ij} la quantité du flot sur l'arc (i, j), la contrainte de conservation de flot peut s'exprimer sur la forme suivante :

$$\sum_j x_{ji} - \sum_i x_{ij} = \begin{cases} v & si\ i = s \\ 0 & si\ i \neq p, s \\ -v & si\ i = p \end{cases}$$

- Le modèle mathématique devient:

$$\begin{cases} \sum_j x_{ji} - \sum_i x_{ij} = \begin{cases} v & si\ i = s \\ 0 & si\ i \neq p, s \\ -v & si\ i = p \end{cases} \\ 0 \leq x_{ij} \leq c_{ij} \\ f(u_r) = v\ (Max) \end{cases}$$

- Ce modèle est un programme linéaire à variables bornées.!

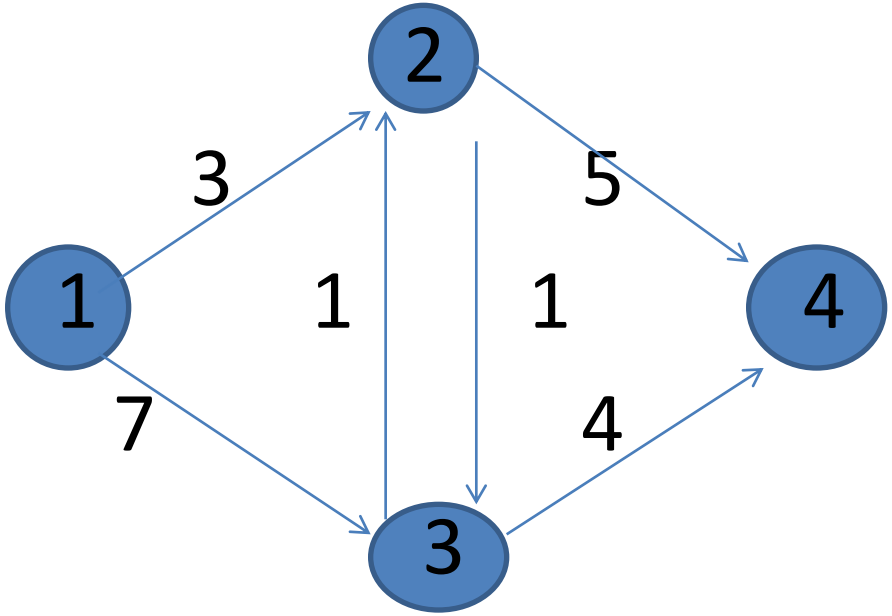
Coupe minimale

- Une coupe $\zeta = w^+(A)$ séparant p de s et engendré par $A \subset X$ est un cocircuit $w^+(A)$ où $s \in A$ et $p \notin A$.
- Sa capacité est: $c(\zeta) = \sum_{u \in \zeta} c(u)$
- Une coupe ζ^* est dite minimale si sa capacité est minimale. Le problème de la coupe minimale se formule comme suit :

$$\begin{cases} \text{Trouver une coupe } \zeta^* \\ c(\zeta^*) \leq c(\zeta) \; \forall \, \zeta \text{ coupe séparant p de s} \end{cases}$$

Exemple

- Soit le réseau suivant:



Déterminer toutes les coupes ainsi que la coupe minimale.

Chaîne augmentante

- Une chaine C est dite augmentante par rapport à un flot f si elle vérifie :

$$\forall u \in C^{+} \quad f(u) < c(u).$$

$$\forall u \in C^{-} \quad f(u) > 0$$

C^{+} : Ensemble des arcs allant de s vers p et

C^{-}: Ensemble des arcs dans le sens inverse.

Dans l'exemple précédent:

C = { (1,3) ; (3,4)} est une Chaîne augmentante pour le flot f = (2, 0, 1, 0, 1, 1) de valeur 2.

avec: u_1=(1,2), u_2=(1,3) u_3=(2,3) u_4=(3, 2) u_5=(2,4) u_6=(3,4)

Résultats théoriques (1/2)

Lemme: Soient R=(X, U, c) un réseau

$\forall \zeta = w^+(A)$ coupe séparant p de s

$\forall$ v=f(ur) flôt réalisable

alors on a: **$v \leq c(\zeta)$**

En effet: Soient flot f réalisable et A une partie de X vérifiant $s \in A$ et $p \notin A$

f est un flot $\Leftrightarrow \sum_{u \in w^+(A)} f(u) = \sum_{u \in w^-(A)} f(u) = \sum_{u \in w^-(A) \cup \{u_r\}} f(u)$ + f(ur)

$\Rightarrow f(u_r) = \sum_{u \in w^+(A)} f(u) - \sum_{u \in w^-(A) \cup \{u_r\}} f(u) \leq \sum_{u \in w^+(A)} f(u)$

Comme f réalisable $f(u) \leq c(u) \; \forall u \in U \cup \{u_r\}$

D'où le résultat

Résultats théoriques (2/2)

Théorème: Un flôt est maximum si et seulement s'il n'existe aucune chaine augmentante

CN: Evidente

CS: Considérons un flot réalisable f* n'ayant pas de chaîne augmentante alors il existerait un cocycle $\Omega = \omega(A)$ vérifiant :

$\forall u \in \Omega^+ \quad f^*(u) = c(u)$

$\forall u \in \Omega^- \quad f^*(u) = 0$

$$f^*(u_r) = \sum_{u \in w^+(A)} f^*(u) - \sum_{u \in w^-(A)\cdot\{u_r\}} f^*(u) = \sum_{u \in w^+(A)} f^*(u) = \sum_{u \in w^+(A)} c(u) = c(\zeta^*)$$

$\zeta^* = \omega^+(A)$ est une coupe séparant p de s et f* est maximum (Lemme précédent.)

Remarque : Si toutes les capacités des arcs sont entières alors il existe un flot entier maximum.

Algorithme de Ford-Fulkerson (1/2)

- ***Principe:*** Démarrer d'un flot réalisable (par exemple, nul) et l'améliorer d'itération en itération. Utiliser une procédure de marquage pour exhiber la chaîne augmentante. Dans la cas où elle n'existe pas, le flôt courant est optimal (on aurait construit une coupe minimale séparant p de s).
- ***Procédure de marquage***: On marque le sommet s d'un (+) et on pose d(s)=+$\propto$ (flôt infini). On pose $C^+=C^-=\varnothing$ et Y={s}

  Marquage direct: S'il existe un arc u=(x, y) avec x marqué et y non marqué vérifiant: f(u) < c(u) alors on marque y d'un +x et on pose: d(y)=Min{d(x); c(u)-f(u)}, $C^+=C^+ \cup \{u\}$ et $Y=Y \cup \{y\}$

  Marquage indirect: S'il existe un arc u=(y, x) avec x marqué et y non marqué vérifiant: f(u) >0 alors on marque y d'un -x et on pose: d(y)=Min{d(x); f(u)}, $C^-=C^- \cup \{u\}$ et $Y=Y \cup \{y\}$.

Algorithme de Ford-Fulkerson (2/2)

- Deux cas se présentent à la fin de la procédure de marquage:

- ***p marqué***: C= C+ $\cup$ C- est une chaîne augmentante. Le flôt est amélioré comme suit:

$$f^{k+1}(u) = \begin{cases} f^k(u) + \varepsilon & \forall\, u \in C^+ \\ f^k(u) - \varepsilon & \forall\, u \in C^- \\ f^k(u) & \forall\, u \notin C \end{cases}$$

où: f^k(u) est la valeur du flot sur l'arc u à l'itération k et ε = d (p) = min{d(x)/ x$\in$ Y}.

- ***p n'est pas marqué***: Le flot courant est optimal et w^+(Y) est la coupe minimale

Amélioration de l'algorithme

- Un ***flôt*** est dit ***comple***t si tout chemin de s à p contient au moins un arc saturé (f(u=c(u) ou f(u)=0)

Procédure:

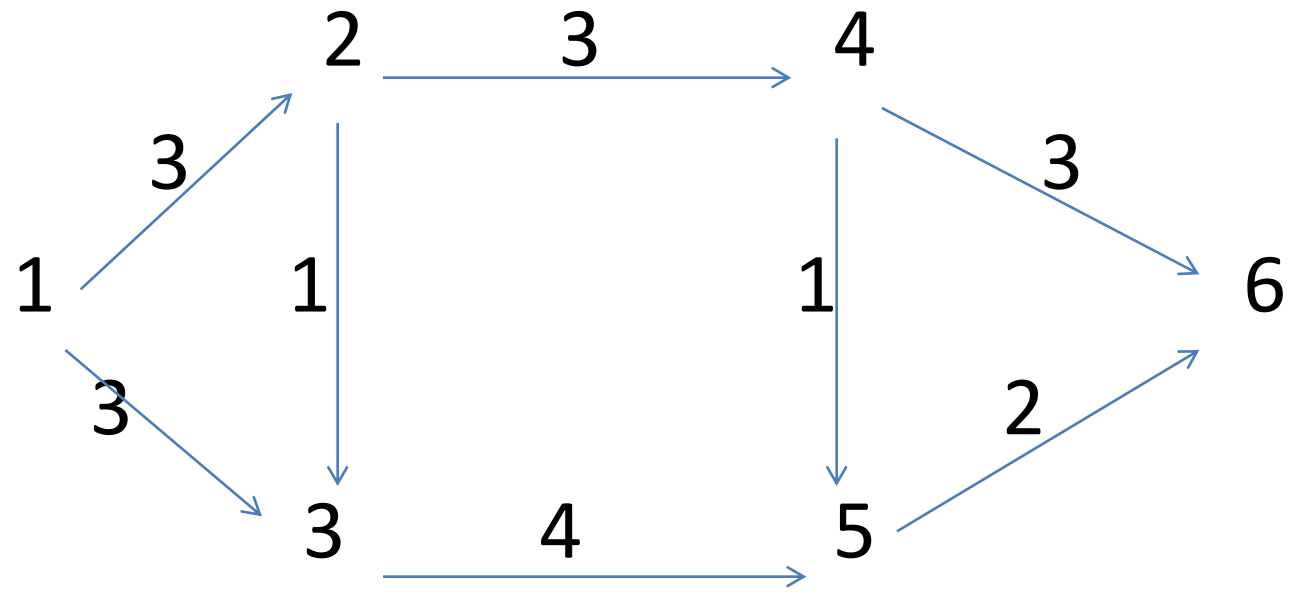
a) Recherche d'un flot « au jugé » en distribuant de sommet en sommet le flot en respectant la conservation de la matière

b) Considérer un chemin de s à p n'ayant aucun arc saturé et augmenter la valeur du flot sur ce chemin jusqu'à saturation d'un arc (en respectant la conservation de la matière sur les sommets de ce chemin)

c) Répéter l'opération jusqu'à saturer tous les chemins du réseau

Exemple

Soit le réseau suivant:



Considérer le flot au jugé suivant:

	(1,2)	(1,3)	(2,3)	(2,4)	(3,5)	(4,5)	(4,6)	(5,6)	
f = (	2,	0,	1,	1,	1,	0,	1,	1	) de valeur 2

Complexité de Ford-Fulkerson

Soit G=(X, U) avec |X|= n , U|= m et c*=Max {c(u), u$\in$U}

On suppose que les capacités des arcs sont entières

- O(m) opérations pour la recherche d'une chaîne améliorante et l'amélioration du flot.
- La capacité d'une coupe est au plus en O(n c*). En effet, dans le pire des cas le flot augmente d'une seule unité à chaque fois. Donc au plus O(n c*) améliorations.

- Complexité Ford-Fulkerson est en O(n m c*)$\approx$ O(n^4)

Quelques variantes de l’algorithme

- Variante : Algorithme d'Edmonds-Karp (1972). C'est une implémentation particulière de l'algorithme de Ford-Fulkerson en parcours largeur (BFS) et qui consiste à toujours choisir une chaine améliorante de plus court chemin de s à t, c'est-à-dire celle avec le moins d'arêtes possibles.
- Cet algorithme se termine toujours (même pour des capacités non entières, contrairement à Ford-Fulkerson...) avec une complexité en $O(nm^2)$ (indép. des capacités).
- Algorithme de Push-Relabel