Chapitre 4

Flots et réseaux de transport

I Cycles, flux et flots

Soit $G = (X = \{x_1, \dots x_n\}, U = \{u_1, \dots u_m\})$ un graphe orienté où X est un ensemble de sommets et U un ensemble d'arcs (un arc représentant un lien orienté entre deux sommets) où $\forall i \in [\![1, m]\!], u_i \in X \times X$.

Définition 5 (Chemin, Chaîne)**.** *Un chemin est une suite d'au moins deux sommets $(s_1, \dots s_p)$ telle que $\forall i \in [\![1, p-1]\!]$, (s_i, s_{i+1}) est un arc ($\in U$). Pour supprimer toute ambigüité, un chemin est représenté par la séquence de ses arcs.*

Une chaîne est une suite de au moins deux sommets $(s_1, \dots s_p)$ telle que $\forall i \in [\![1, p-1]\!]$, (s_i, s_{i+1}) ou (s_{i+1}, s_i) est un arc ($\in U$).

Un chemin ou une chaîne sont dits simples si leurs arcs sont tous différents, il sont élémentaires si leurs sommets sont tous différents. La longueur d'un chemin ou d'une chaîne est son nombre d'arcs. (Par définition, un chemin ou une chaîne de longueur O n'existe pas.)

Définition 6 (Vecteur cycle)**.** *Un cycle μ est une chaîne **simple** dont les extrémités coincident. Le vecteur cycle $\mu = (\mu^1, \dots, \mu^m)$ de $\mathbb{Z}^m$ associé au cycle μ est défini par*

$$\forall i \in [\![1, m]\!], \mu^i = \begin{cases} 0 & \text{si l'arc } u_i \text{ n'apparaît pas dans le cycle} \\ 1 & \text{si l'arc } u_i \text{ est utilisé dans le sens de parcours du cycle} \\ -1 & \text{si l'arc } u_i \text{ est utilisé dans le sens opposé} \end{cases}$$

Étant donné un ensemble de sommets A, on désigne par $\omega^+(A)$ l'ensemble des arcs incidents à A vers l'extérieur (sortants), et par $\omega^-(A)$ l'ensemble des arcs incidents à A vers l'intérieur (entrants) et l'on pose $\omega(A) = \omega^+(A) \cup \omega^-(A)$. Si $\omega(A)$ est non vide il est appelé cocycle de A.

Définition 7 (Flot)**.** *Un flot sur un graphe est un vecteur $\varphi = (\varphi^1, \dots, \varphi^m)$ de $\mathbb{Z}^m$ tel que le nombre $\varphi^i = \varphi(u_i)$ est appelé flux dans l'arc u_i et qui vérifie la loi de Kirchhoff ou loi de conservation du flux en chaque sommet du graphe :*

$$\forall x \in X, \sum_{u_i \in \omega^-(\{x\})} \varphi(u_i) = \sum_{u_i \in \omega^+(\{x\})} \varphi(u_i) \qquad \textit{(loi de Kirchhoff)}$$

Propriété 5.

1. *Toute combinaison linéaire de flot sur G définit un flot sur G*
2. *Le vecteur nul de $\mathbb{Z}^m$ est un flot sur tout graphe G (dit "flot nul")*
3. *Tout vecteur cycle de G est un flot sur G*

Propriété 6. *φ est un flot sur G ssi*

$$\forall \emptyset \subset A \subset X, \sum_{u \in \omega^-(A)} \varphi(u) = \sum_{u \in \omega^+(A)} \varphi(u) \qquad \textit{(loi de Kirchhoff généralisée)}$$

II Flots compatibles dans un Réseau de transport

Définition 8 (réseau de transport)**.** *Un réseau de transport est un graphe orienté connexe $R = (X, U = \{u_1, \dots u_m\})$ avec*

- *un sommet sans prédecesseur appelé entrée (ou source) noté s ($\Gamma^-(s) = \emptyset$)*
- *un sommet sans suivant appelé sortie (ou puits) noté t ($\Gamma^+(t) = \emptyset$)*
- *une application capa : $U \to \mathbb{Z}^+ \cup \{\infty\}$ qui à chaque arc u associe sa capacité $capa(u) \geq 0$.*

On y adjoint un arc u_0 (fictif) $u_0 = (t, s)$ de capacité infinie qui sera appelé arc de retour.

Définition 9 (flot compatible)**.** *Un flot compatible ou encore réalisable sur un réseau $R = (X, U)$ avec $\mid U \mid= m$ est un vecteur φ de $\mathbb{Z}^{m+1}$ tel que :*

- *φ est un flot sur $R \cup \{u_0\}$: $\forall x \in X, \sum_{u \in \omega^+(\{x\})} \varphi(u) = \sum_{u \in \omega^-(\{x\})} \varphi(u)$*
- *φ est compatible avec les capacités : $\forall u \in U, \quad 0 \leq \varphi(u) \leq capa(u)$*

La valeur $v(\varphi)$ du flot φ est le flux qui traverse l'arc de retour ou encore le flux sortant de la source ou encore le flux arrivant à la sortie : $v(\varphi) = \varphi(u_0) = \sum_{u \in \omega^+(\{s\})} \varphi(u) = \sum_{u \in \omega^-(\{t\})} \varphi(u)$

Si $\varphi(u) = c(u)$ on dit que l'arc u est saturé.

Un flot est dit de valeur maximale s'il maximise cette valeur $v(\varphi)$ dans l'ensemble de tous les flots compatibles (un tel flot n'est pas nécessairement unique).

1 Théorème de la coupe

Une coupe correspond à une partition des sommets en deux sous-ensembles tels que la source soit dans l'un et la destination dans l'autre. Plus formellement,

Étant donné un réseau de transport $G = (X, U)$ ayant pour source s et puits t, muni de capacités *capa* sur les arcs, telles que $\forall u \in U, \quad capa(u) \in \mathbb{N}^+ \cup \{+\infty\}$:

Définition 10. *Une coupe cp séparant s et t est un couple*

$$cp = (A, \quad X \setminus A) \text{ tel que } A \subset X, s \in A \text{ et } t \notin A$$

Les arcs de la coupe $cp = (A, X \setminus A)$ sont les arcs de $\omega^+(A)$.

La capacité d'une coupe $cp = (A, X \setminus A)$ est la somme des capacités de ses arcs (i.e. des arcs sortants de A),

$$capa(cp) = \sum_{u \in \omega^+(A)} capa(u)$$

Théorème 1 (de la coupe)**.** *Pour tout flot φ compatible sur R et pour toute coupe cp séparant s et t la valeur du flot est inférieure à la capacité de cette coupe :*

$$v(\varphi) \leq capa(cp)$$

2 Principe de Ford-Fulkerson

Le principe de Ford-Fulkerson [9] utilise un principe de marquage relatif à un flot compatible φ. On peut prendre comme flot compatible le flot nul si on n'a pas mieux.

Définition 11 (Principe de marquage de Ford-Fulkerson)**.** *On marque l'entrée s du réseau, puis*
x étant un sommet marqué , y est marquable à partir de x ssi
y n'est pas marqué et $\begin{cases} \exists u = (x, y) \in R \text{ et } \varphi(u) < capa(u) & \text{marquage direct} \\ \exists u = (y, x) \in R \text{ et } \varphi(u) > 0 & \text{marquage indirect}(u \neq u_0) \end{cases}$

Propriété 7. *Si à la fin de la procédure de marquage*

1. *on parvient à marquer la sortie t du réseau alors on peut augmenter la valeur du flot de k calculée comme suit : soit ch la chaîne d'origine s ayant permis de marquer t, cette chaîne est appelée chaîne augmentante, et soit ch^+ les arcs de ch participant aux marquages directs et ch^- les arcs de ch participant aux marquages indirects, $k = \min(\min_{u \in ch^+} capa(u) - \varphi(u), \min_{u \in ch^-(u)} \varphi(u))$. Le nouveau flot est obtenu en augmentant de k le flux sur les arcs de ch^+ et en diminuant de k sur les arcs de ch^-.*
2. *on ne parvient pas à marquer la sortie alors le flot est maximum*

L'algorithme de Ford-Fulkerson consiste donc à appliquer le marquage de Ford-Fulkerson tant qu'on peut marquer la sortie : dans ce cas-là, on augmente le flot, on efface le marquage et on recommence un nouveau marquage à partir du nouveau flot, quand on ne peut plus marquer la sortie alors on s'arrête car le flot est maximum.

Théorème 2 (Théorème du flot maximum (Ford Fulkerson 1957))**.** *Soit F l'ensemble des flots compatibles et K l'ensemble des coupes dans un réseau de transport,*

$$\max_{\varphi \in F} v(\varphi) = \min_{cp \in K} capa(cp)$$

La valeur maximum du flot est égale à la capacité minimum d'une coupe.

3 Graphe d'écart (voir TD)

Définition 12 (graphe d'écart)**.** *Le graphe d'écart $G_R(\varphi) = (X, U_\varphi, r_\varphi)$ associé à un réseau de transport $R = (X, U, c)$ et à un flot compatible φ de R est défini par :*
tout arc $(x, y) \in U$ induit au moins un des 2 arcs suivants :

$$\begin{cases} \text{un arc dit “direct” : } (x, y) \in U_\varphi \text{ si } \varphi(x, y) < capa(x, y) & \text{avec } r_\varphi = r^+(x, y) \\ \text{un arc dit “indirect” : } (y, x) \in U_\varphi \text{ si } \varphi(y, x) > 0 & \text{avec } r_\varphi = r^-(x, y) \end{cases}$$

Propriété 8. *Il existe une chaîne élémentaire de s à t FF-marquable relativement à φ dans R ssi il existe un chemin élémentaire de s à t dans $G_R(\varphi)$.*

Conséquence : on peut utiliser les graphes d'écart au lieu de la procédure de marquage.

4 Complexité

Dinic en 1970 [6], puis Edmond et Karp en 1972 [7] ont montré que l'algo de Ford Fulkerson devient polynomial si la recherche des chemins de s à t se fait en largeur d'abord (plus courts chemins en nombre d'arcs).

III Flots maximum de cout minimum (voir TD)

1 Flots de coûts minimum

$R = (X, U, c, \gamma)$ réseau de transport avec

— $capa(u) \geq 0$ capacité de l'arc u et
— $\gamma(u) \geq 0$ coût de passage d'une unité de flux dans l'arc u
— avec $capa(u_0) = \infty$ et $\gamma(u_0) = 0$.
Le coût d'un flot φ :$\gamma(\varphi) = \sum_{u \in U} \gamma(u)\varphi(u)$
Trouver un flot compatible de coût minimum. Voire un flot max de coût min.

2 Algorithme de Roy, Busacker et Gowen

Algorithme de Roy (1960) [19], Busacker et Gowen (1961) [4]
— Initialisation : $\varphi \leftarrow (0, \ldots, 0)$ (flot nul)
— Étape courante : Construire le graphe d'écart $G(R, \varphi) = (X, U_\varphi, capa_\varphi, \gamma_\varphi)$ avec $\gamma_\varphi(u) = \begin{cases} \gamma(u) & \text{si } u \text{ est un arc direct} \\ -\gamma(u) & \text{si } u \text{ est un arc indirect} \end{cases}$
— Si $\nexists$ chemin $[s, t]$ dans $G(R, \varphi)$
  alors FIN : φ est un flot maximum de coût minimum
  Sinon
  — déterminer un chemin ν_{st} élémentaire st-minimal de $(X, U_\varphi, \gamma_\varphi)$
  — $k \leftarrow \min_{u \in \nu_{st}} capa_\varphi(u)$
  — $\mu_{st} \leftarrow$ vecteur cycle $= \nu_{st} \cup \{u_0\}$
  — $\varphi \leftarrow \varphi + k\mu_{st}$ nouveau flot compatible $\gamma(\varphi) \leftarrow \gamma(\varphi) + k.\gamma_\varphi(\nu_{st})$ coût du nouveau flot
— réitérer pour obtenir un flot de valeur supérieure

3 Autre méthode : suppression des circuits négatifs

Théorème 3 (Théorème d'optimalité de Berges)**.** *Soit φ un flot compatible, φ est de coût minimum ssi $G(R, \varphi)$ n'admet aucun circuit de coût négatif.*

Algorithme de Klein 1967 [13] :
— construire un flot φ compatible sur $R \cup \{u_0\}$
— tant que $G(R, \varphi)$ admet un circuit négatif μ (φ n'est pas de coût minimum) faire
  — soit k la capacité résiduelle minimale de ce circuit,
  — $\varphi \leftarrow \varphi + k\mu$. (flot de même valeur mais de coût inférieur)

Variante algo polynomial : choisir de supprimer le circuit de coût moyen minimum (= coût du circuit/ nombre d'arcs du circuit).

Comment détecter les circuits négatifs ? On peut utiliser un algorithme de plus court chemin qui le fait (Belman-Kalaba ou Moore) ou utiliser un parcours en profondeur d'abord (DFS) où à chaque fois qu'on trouve un arc de retour on calcule le poids du circuit.