

Chapitre 8

Flots

Dans ce chapitre nous définissons d'abord la notion de flot compatible et donnons la condition d'existence d'un tel flot. Nous présentons ensuite les propriétés qui interviennent de manière cruciale dans la conception des algorithmes sur les flots, à savoir la décomposition en cycles, l'absence de cycles améliorants dans un flot de coût minimum et la dominance des flots entiers. Nous définissons ensuite le problème du flot maximum d'une source vers un puits. Après avoir défini la notion de graphe d'écart et démontré le théorème de Ford et Fulkerson, nous analysons les algorithmes les plus performants pour ce problème, à savoir l'algorithme primal des distances estimées au puits et l'algorithme dual du préflot. Deux variantes efficaces de l'algorithme du préflot sont décrites, l'algorithme de Karzanov et l'algorithme des excès échelonnés. Nous considérons ensuite le problème du flot de coût minimum. Nous définissons le problème dual et donnons la condition nécessaire et suffisante d'optimalité d'un flot et d'un ensemble de potentiels, exprimée avec les coûts réduits des arcs du graphe d'écart. Nous présentons alors un algorithme primal dû à Goldberg et Tarjan fondé sur la notion d' ϵ -optimalité et sur les circuits de coût moyen minimum. Nous montrons ensuite comment transformer un problème de flot compatible de coût minimum en problème de flot maximum de coût minimum et nous présentons un algorithme dual pour résoudre ce dernier problème. Nous présentons enfin l'algorithme efficace d'Edmonds et Karp pour la recherche d'un plan de transport de coût minimum.

Introduction

La notion de flot modélise de manière naturelle une circulation discrète (voitures, électrons,...) ou continue (fluide,...), le long des arcs d'un graphe orienté. Cette circulation est soumise à des contraintes d'équilibre de charge aux sommets du graphe et de limitation de capacité sur les arcs. De nombreux problèmes d'optimisation se ramènent à la recherche de flots particuliers sur un graphe, les

plus connus sont les problèmes de transport (par exemple entre des fournisseurs et des clients), les problèmes de couplage et d'affectation (par exemple de tâches à des machines) ou encore les problèmes de chemins de coût minimum. Mais ce modèle intervient aussi souvent dans d'autres domaines comme la résolution de certains problèmes d'ordonnancement, le placement de tâches ou encore la maximisation de certaines fonctions booléennes. La théorie des flots occupe une place très intéressante en optimisation combinatoire car elle se situe à la frontière de la programmation linéaire continue et de la programmation en nombres entiers. En effet, les problèmes de flot que nous étudierons peuvent être formulés comme des programmes linéaires continus mais la matrice sous-jacente est telle que les solutions entières sont dominantes.

8.1 Préliminaires

Dans tout le chapitre, les graphes considérés sont orientés mais ne sont pas nécessairement simples, c'est-à-dire que des arcs distincts peuvent avoir la même origine et la même extrémité. Soit u un arc d'un graphe $G = (S, A)$, il sera commode de noter u^- son sommet origine et u^+ son sommet extrémité.

Soit E un ensemble fini. L'ensemble des fonctions de E dans \mathbb{Z} est noté $\mathcal{F}(E)$. Le *produit scalaire* de deux éléments f et g de $\mathcal{F}(E)$ est par définition le nombre

$$f \star g = \sum_{e \in E} f(e)g(e).$$

L'ordre partiel sur $\mathcal{F}(E)$ défini par $\forall e \in E, f(e) \leq g(e)$ est noté $f \leq g$. La relation $f < g$ est définie par $f \leq g$ et $f \neq g$. La *fonction caractéristique* d'un sous-ensemble F de E est notée χ_F .

Soient $G = (S, A)$ un graphe et T un sous-ensemble de sommets. L'ensemble $\omega^-(T)$ des arcs *entrants dans* T et l'ensemble $\omega^+(T)$ des arcs *sortants de* T sont définis par :

$$\begin{aligned}\omega^-(T) &= \{u \in A \mid u^- \in S - T, u^+ \in T\} \\ \omega^+(T) &= \{u \in A \mid u^- \in T, u^+ \in S - T\}.\end{aligned}$$

Soient f une fonction de $\mathcal{F}(A)$ et T un sous-ensemble de sommets, les nombres $f \star \chi_{\omega^-(T)}$ et $f \star \chi_{\omega^+(T)}$ sont notés respectivement $f^-(T)$ et $f^+(T)$. L'*excès* de la fonction f pour la partie T est défini par $e_f(T) = f^-(T) - f^+(T)$. Lorsque T est le singleton $\{s\}$, les notations précédentes sont simplifiées en $f^-(s)$, $f^+(s)$ et $e_f(s)$.

Remarquons que pour toute fonction f de $\mathcal{F}(A)$, l'excès de S est nul et que l'excès d'une partie non vide T de S est égal à la somme des excès des sommets de T , c'est-à-dire $e_f(T) = f^-(T) - f^+(T) = \sum_{s \in T} e_f(s)$.

Comme la notion de chaîne n'a été introduite que pour les graphes non orientés (voir section 4.1.3), nous en définissons ici une extension pour les graphes orientés.

Une *chaîne* γ du graphe $G = (S, A)$ est une suite :

$$\gamma = ((s_0, u_1, s_1), (s_1, u_2, s_2), \dots, (s_{q-1}, u_q, s_q))$$

telle que pour tout $i \in \{1, \dots, q\}$ les deux extrémités de l'arc u_i sont s_{i-1} et s_i . Une chaîne est *élémentaire* si tous les sommets s_i pour $i \in \{0, \dots, q\}$ sont distincts. Un arc u_i de la chaîne γ est dit *avant* si $u_i^- = s_{i-1}$ et $u_i^+ = s_i$, dans le cas contraire il est dit *arrière*. Un *cycle élémentaire* est une chaîne telle que tous les sommets $s_i, i \in \{0, \dots, q-1\}$ sont distincts et $s_0 = s_q$. Sur la figure 1.1 sont représentés à gauche une chaîne élémentaire $[(1, a, 2), (2, b, 3), (3, c, 4), (4, d, 5)]$ et à droite un cycle élémentaire $[(1, a, 2), (2, b, 3), (3, c, 4), (4, d, 5), (5, e, 6), (6, f, 1)]$.

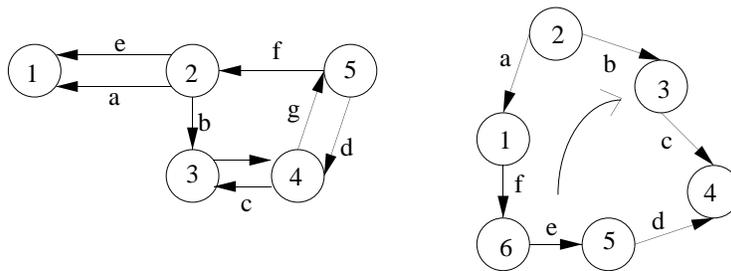


Figure 1.1: Chaîne et cycle élémentaires.

8.2 Flots d'un réseau

Un *réseau* $R = (G, a, b)$ est un graphe connexe sans boucles $G = (S, A)$, muni de deux fonctions a et b de $\mathcal{F}(A)$ vérifiant $a \leq b$. Les entiers $a(u)$ et $b(u)$ sont respectivement appelés *capacité minimale* et *capacité maximale* de l'arc u .

Un *réseau valué* $R = (G, a, b, c)$ est un réseau (G, a, b) muni d'une fonction c de $\mathcal{F}(A)$. Le nombre $c(u)$ est appelé *coût* de l'arc u .

Un *flot* f du réseau $R = (G, a, b)$ est une fonction $f : A \mapsto \mathbb{Q}$ vérifiant la *condition d'équilibre* (E) ci-dessous :

$$\forall s \in S \quad e_f(s) = 0 \tag{E}$$

Un flot *compatible* est un flot qui vérifie également les *conditions de limitation de capacité* (L) suivantes :

$$a \leq f \leq b \tag{L}$$

Remarque. Le choix de flux rationnels (nous aurions pu également considérer des flux réels) a été fait pour mieux mettre en évidence par la suite la propriété de dominance des flots entiers vis-à-vis du critère coût.

Soient f un flot du réseau R et T un sous-ensemble de sommets. Le nombre $f(u)$ est appelé *flux* de l'arc u , les quantités $f^-(T)$ et $f^+(T)$ sont appelées respectivement *flux entrant dans T* et *flux sortant de T* .

Remarquons que si la condition d'équilibre est satisfaite en tout sommet, elle est également vérifiée pour toute partie non vide T de sommets :

$$e_f(T) = f^-(T) - f^+(T) = 0.$$

Le *coût* du flot f d'un réseau valué est par définition le nombre $f \star c$.

Les diverses quantités introduites jusqu'ici s'expriment facilement en utilisant la matrice M d'incidence sommets-arcs du graphe G . La ligne du sommet s représente la fonction $\chi_{\omega^+(s)} - \chi_{\omega^-(s)}$. Un flot compatible d'un réseau $R = (G, a, b)$ à m arcs est donc un vecteur f de \mathbb{Q}^m satisfaisant :

$$Mf = 0 \quad \text{et} \quad a \leq f \leq b.$$

Un flot compatible de coût minimum du réseau valué $R = (G, a, b, c)$ est une solution optimale du programme linéaire :

$$\begin{aligned} f &\leq b \\ -f &\leq -a \\ Mf &= 0 \\ \text{MIN} \quad &\sum_{u \in A} f(u)c(u) \end{aligned}$$

où les variables $f(u)$, $u \in A$ sont rationnelles.

L'ensemble des flots d'un réseau constitue un *espace vectoriel* sur \mathbb{Q} . Soient f et g deux flots et α et β deux nombres rationnels; si $a \leq \alpha f + \beta g \leq b$ alors $\alpha f + \beta g$ est un flot compatible du réseau.

8.2.1 Existence d'un flot compatible de coût minimum

Les contraintes imposées par les limitations de capacité rendent la question de l'existence d'un flot compatible dans un réseau $R = (G, a, b)$ non triviale. Une condition nécessaire et suffisante d'existence due à Hoffman exprime simplement que pour tout sous-ensemble non vide T de sommets, le flux entrant minimal $a^-(T)$ doit être inférieur ou égal au flux sortant maximal $b^+(T)$. Cette condition de *consistance*, notée (C) , s'exprime par :

$$\forall T \subset S, \quad a^-(T) \leq b^+(T). \quad (C)$$

Soient $R = (G, a, b)$ un réseau et g un flot de ce réseau. La distance $\Delta_g(u)$ de $g(u)$ à l'intervalle $[a(u), b(u)]$ étant définie par :

$$\Delta_g(u) = \begin{cases} g(u) - b(u) & \text{si } g(u) > b(u) \\ a(u) - g(u) & \text{si } g(u) < a(u) \\ 0 & \text{si } a(u) \leq g(u) \leq b(u) \end{cases}$$

l'incompatibilité du flot g est mesurée par $i(g) = \Delta_g \star \chi_A$. Le lemme 2.1 montre que si la condition de compatibilité est satisfaite, il existe un flot h dont l'incompatibilité est strictement plus petite que celle de g .

Lemme 2.1. *Soit g un flot entier tel que $i(g) > 0$. Si le réseau est consistant, il existe un flot h tel que $i(h) < i(g)$.*

Preuve. Si $i(g)$ est strictement positif, il existe un arc v vérifiant par exemple $g(v) > b(v)$. On notera respectivement x et y l'origine et l'extrémité de v . Notons $H = (S, B)$ le graphe dont les arcs sont colorés en rouge ou en noir par la procédure COLORER ci-dessous :

```

procédure COLORER( $G, a, b, v$ );
  pour tout arc  $u$  de  $G$  faire
    si  $g(u) > a(u)$  alors
      créer un arc rouge  $u'$  d'origine  $u^-$  et d'extrémité  $u^+$  dans  $H$ ;
    si  $g(u) < b(u)$  alors
      créer un arc noir  $u''$  d'origine  $u^+$  et d'extrémité  $u^-$  dans  $H$ .
  
```

La figure 2.1 montre le graphe H correspondant à un flot nul et à l'arc $v = (4, 1)$. Soit T l'ensemble des sommets accessibles à partir de y dans H et supposons que T

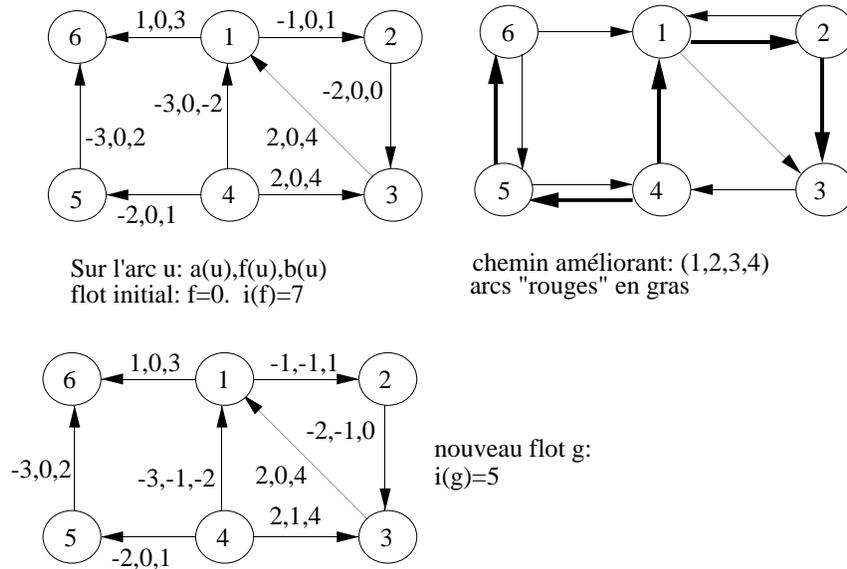


Figure 2.1: Amélioration de la compatibilité d'un flot.

ne contienne pas x . Dans le graphe G , un arc u de $\omega^-(T)$ satisfait nécessairement $g(u) \geq b(u)$ et un arc u de $\omega^+(T)$ satisfait nécessairement $g(u) \leq a(u)$. De plus l'arc v appartient à $\omega^-(T)$. Il en résulte que :

$$a^-(S - T) = a^+(T) \geq g^+(T) = g^-(T) > b^-(T) = b^+(S - T).$$

ce qui contredit la condition de compatibilité. Il existe donc un chemin rouge et noir de y à x dans H . Pour tout arc rouge u' de ce chemin posons $h(u) = g(u) - 1$, pour tout arc noir u'' de ce chemin, posons $h(u) = g(u) + 1$. Posons enfin $h(v) = g(v) - 1$ et $h(u) = g(u)$ pour tous les autres arcs de G . La fonction h est un flot du réseau R tel que $i(h) \leq i(g) - 1$ car on a $\Delta_h(v) = \Delta_g(v) - 1$ et pour tout arc u de G , $\Delta_h(u) \leq \Delta_g(u)$. Pour le flot g de la figure 2.1, le chemin $(1, 2, 3, 4)$ est améliorant et l'incompatibilité diminue de deux unités. ■

Le théorème de Hoffman résulte alors directement du lemme 2.1.

Théorème 2.2 (de Hoffman). *Un réseau possède un flot compatible si et seulement s'il est consistant.*

Preuve. La condition est suffisante car à partir d'un flot g (par exemple le flot nul), il est possible d'après le lemme 2.1 d'obtenir en au plus $i(g)$ itérations un flot compatible. La condition est nécessaire car si f est un flot compatible et T un sous-ensemble de sommets, la sommation des excès des sommets de T pour f conduit à :

$$0 = f^+(T) - f^-(T) \leq b^+(T) - a^-(T).$$

■

Si la condition de consistance est satisfaite pour un réseau valué $R = (G, a, b, c)$, nous savons d'après le théorème de Hoffman que l'ensemble de ses flots compatibles n'est pas vide. Cet ensemble formé des vecteurs de \mathbb{Q}^m solutions de $Mf = 0$ et $a \leq f \leq b$, est un *polyèdre convexe fermé et borné* de \mathbb{Q}^m . Il résulte alors d'un théorème important de la programmation linéaire continue que la fonction linéaire $f \mapsto f \star c$, où f parcourt l'ensemble des flots compatibles de R , atteint son minimum en un point extrême du polyèdre. Nous en concluons que la condition de consistance est aussi une condition nécessaire et suffisante d'existence d'un flot compatible de coût minimum.

Remarque. Dans la suite, tout réseau $R = (G, a, b)$ sera supposé consistant.

8.2.2 Quelques problèmes particuliers

Nous présentons dans ce paragraphe quelques problèmes d'optimisation dont les solutions peuvent être interprétées comme les flots de coût minimum d'un réseau valué.

Problèmes de transport.

Un *réseau de transport* est défini à partir d'un réseau valué $R = (G, a, b, c)$ et d'une fonction d de $\mathcal{F}(S)$ vérifiant $\sum_{s \in S} d(s) = 0$ appelée fonction *d'offre et de demande*. La valeur $c(u)$ représente le coût unitaire de transport sur l'arc u . Les sommets sont répartis en trois sous-ensembles, les *fournisseurs* pour lesquels $d(s) > 0$, les *clients* pour lesquels $d(s) < 0$ et enfin les *sommets de transit* vérifiant

$d(s) = 0$. Le problème est de trouver un *plan de transport* de coût minimum, c'est-à-dire une fonction g de $\mathcal{F}(A)$ satisfaisant $a \leq g \leq b$ telle que :

$$e_g(s) = \begin{cases} -d(s) & \text{si } s \text{ est un fournisseur} \\ d(s) & \text{si } s \text{ est client} \\ 0 & \text{si } s \text{ est un sommet de transit} \end{cases}$$

et dont le coût soit minimum. Le lecteur vérifiera aisément à partir d'un exemple que les plans de transport sont en bijection avec les flots compatibles d'un réseau valué. La figure 2.2 montre un problème de transport (capacités et coûts non représentés) et le réseau valué équivalent. Les coûts d'un plan de transport et de son flot image par la bijection étant égaux, on est ramené à la recherche d'un flot compatible de coût minimum.

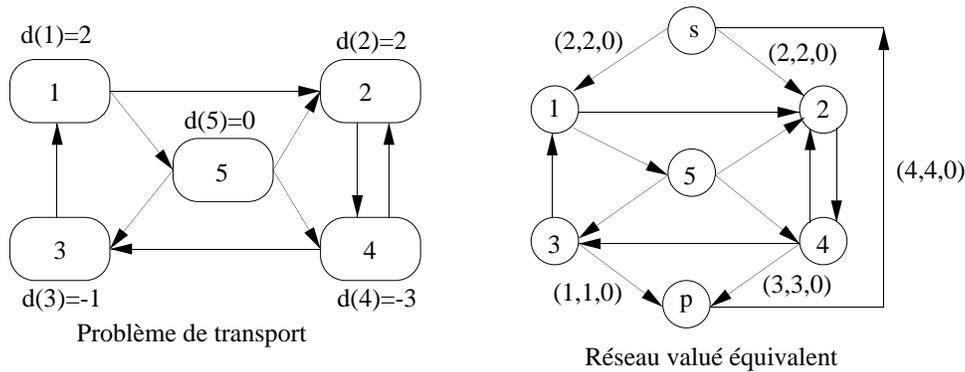


Figure 2.2: *Problème de transport et réseau valué équivalent.*

Chemins de coût minimum.

Soit G un graphe à n sommets dont les arcs sont valués par une fonction coût v et soit s une racine de G . On cherche à déterminer pour chaque sommet x de G un chemin de coût minimum de s à x . Posons $d(x) = -1$ si le sommet x est distinct de s , $d(s) = n - 1$ et pour tout arc u de G , $a(u) = 0$, $b(u) = n$, $c(u) = v(u)$. Un plan de transport entier de coût minimum du réseau de transport ainsi constitué correspond à l'envoi par le fournisseur s d'une unité vers chaque client le long d'un chemin de coût minimum.

Flot maximum.

Soit G un graphe dont deux sommets sont distingués, une source s et un puits p , et dont chaque arc u possède une capacité minimale nulle et une capacité maximale $b(u)$. Le problème du flot maximum de s à p est la recherche d'une fonction f , vérifiant $0 \leq f \leq b$ et $e_f(x) = 0$ pour tout sommet x distinct de la source et du puits, telle que la quantité transportée $e_f(p)$ soit maximum. On alloue alors un coût nul à chaque arc et on crée un *arc de retour* d'origine p , d'extrémité s , de coût -1 et de capacité maximale $b^-(p)$. Maximiser la quantité transportée de s à p revient alors à trouver un flot compatible de coût minimum.

Problème d'affectation.

Soient E et F deux ensembles de p éléments et $c(e, f)$ le coût attaché au couple (e, f) . Le problème d'affectation de E sur F est la recherche d'une bijection $\varphi : E \rightarrow F$ dont le coût total $\sum_{e \in E} c(e, \varphi(e))$ soit minimum. Soit G le graphe complet biparti $(E \cup F, E \times F)$. Munissons chaque arc (e, f) d'une capacité minimale nulle, d'une capacité maximale unité et du coût $c(e, f)$. Posons pour tout $e \in E$, $d(e) = 1$ et pour tout $f \in F$, $d(f) = -1$. Un plan de transport entier de coût minimum correspond à une affectation optimale.

8.2.3 Trois propriétés fondamentales

L'ensemble des flots d'un réseau possède de nombreuses propriétés algébriques. Nous en présentons trois qui interviennent de manière directe dans les algorithmes de recherche de flots optimaux : la décomposition d'un flot entier en cycles élémentaires, une caractérisation d'un flot compatible de coût minimum et l'existence d'un flot compatible de coût minimum dont tous les flux sont entiers.

Décomposition en cycles

Ce paragraphe concerne une propriété générale des flots d'un réseau. Les fonctions capacité minimale et capacité maximale ne jouent aucun rôle.

Flot entier positif ou nul

Soient $R = (G, a, b)$ un réseau et f un flot entier *strictement positif* de ce réseau. Nous montrons l'existence d'un ensemble $C(f)$ de couples $(\mu, \lambda(\mu))$ où μ est un circuit élémentaire de G et $\lambda(\mu)$ un entier strictement positif, tel que :

$$f = \sum_{\mu \in C(f)} \lambda(\mu) \chi_{\mu}.$$

Un tel ensemble $C(f)$ est appelé *décomposition du flot f* sur les circuits de G . La fonction χ_{μ} est appelée par convention *flot canonique* du circuit μ .

Soit $G(f)$ le graphe partiel de G , appelé *graphe support* de f , composé des arcs de G de flux *strictement positif*. Le lemme 2.3 fournit une condition nécessaire et suffisante sur le graphe support pour que le flot f soit nul.

Lemme 2.3. *Un flot entier $f \geq 0$ est nul si et seulement si son graphe support $G(f)$ est sans circuits.*

Preuve. La condition est évidemment nécessaire. Pour la réciproque, nous montrons que si le graphe support est sans circuit, il n'a pas d'arcs. Dans le cas

contraire, il existe au moins un arc u de $G(f)$ dont l'extrémité s n'a pas de successeur. Par définition de $G(f)$, on a $f^-(s) > 0$ et $f^+(s) = 0$. La condition d'équilibre n'est donc pas satisfaite pour le sommet s . ■

Nous pouvons maintenant énoncer le théorème de décomposition.

Théorème 2.4. *Soit $R = (G, a, b)$ un réseau à m arcs. Tout flot entier strictement positif de R est une combinaison linéaire à coefficients entiers strictement positifs d'au plus m flots canoniques de G .*

Preuve. La preuve est constructive. Soit f un flot entier strictement positif et $G(f)$ le graphe support de f . Si $G(f)$ ne possède pas de circuits élémentaires, alors $f = 0$. Dans le cas contraire, nous choisissons un circuit élémentaire μ . La fonction χ_μ est un flot strictement positif de G . Notons ϵ la valeur minimum du flux d'un arc de μ ($\epsilon > 0$). La fonction $h = f - \epsilon\chi_\mu$ est un flot positif ou nul de G et $G(f)$ possède au moins un arc de plus que $G(h)$. En répétant au plus m fois l'opération précédente, on obtient un graphe support qui n'a pas de circuits. Le flot correspondant est donc nul. ■

Le flot positif de la figure 2.3 se décompose sur les circuits élémentaires (S, A, D, F, P, S) , (S, A, D, F, G, P, S) , (S, A, D, G, P, S) , (S, B, E, H, P, S) , (S, C, H, P, S) et (A, D, F, A) .

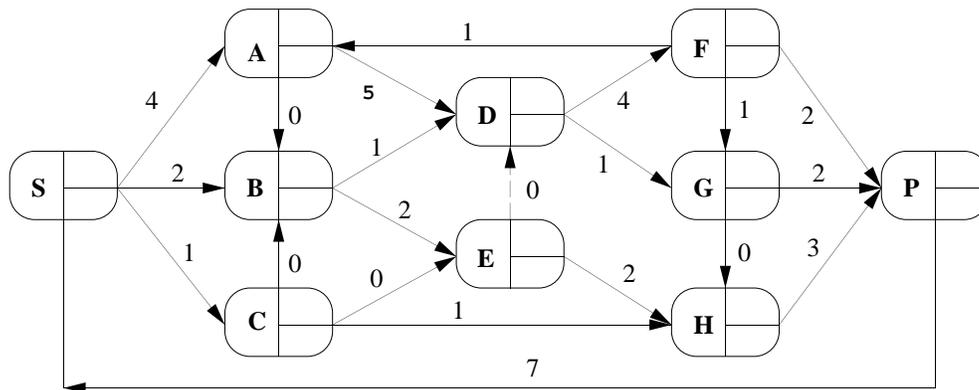


Figure 2.3: Un flot entier positif.

Flot entier quelconque

La propriété précédente se généralise au cas d'un flot entier f dont les flux sont de signe quelconque. Le flot f se décompose alors sur les *cycles élémentaires* de G .

Soit γ un cycle élémentaire. Nous notons respectivement γ^- et γ^+ les sous-ensembles des arcs «arrière» et «avant» de γ . La fonction ϕ_γ définie par $\phi(\gamma) = \chi_{\gamma^+} - \chi_{\gamma^-}$, qui satisfait les conditions d'équilibre en chaque sommet, est appelée *flot canonique* du cycle γ .

Théorème 2.5. Soit f un flot entier d'un réseau $R = (G, a, b)$. Il existe un ensemble $C(f)$ d'au plus m cycles élémentaires de G tel que :

- le flot f est une combinaison linéaire à coefficients entiers strictement positifs des flots canoniques des cycles de $C(f)$,
- les flux (non nuls) alloués à un arc u de G par les flots canoniques de la décomposition ont tous le signe de $f(u)$.

Preuve. Soient $G = (S, A)$ un graphe et soit B une partie de A . Nous notons alors $R' = (G', a', b')$ le réseau obtenu à partir de R en remplaçant chaque arc u de B par un arc opposé noté u' de capacité minimale $a'(u') = -b(u)$ et de capacité maximale $b'(u') = -a(u)$.

On note A' l'ensemble des arcs de G' et B' la partie des arcs de A' provenant de l'inversion d'un arc de B . La fonction f' de $\mathcal{F}(A')$ définie par :

$$f'(v') = \begin{cases} f(v) & \text{si } v' \in A' \setminus B' \\ -f(v) & \text{si } v' \in B' \end{cases}$$

est par construction un flot du réseau R' . De plus l'application définie par $\psi_{A,B} : f \in \mathcal{F}(A) \mapsto f' \in \mathcal{F}(A')$ est linéaire et satisfait :

$$\psi_{A',B'} \circ \psi_{A,B} = id_{\mathcal{F}(A)}.$$

Nous supposons qu'au moins un flux de f est strictement négatif et nous appliquons la transformation précédente à l'ensemble B des arcs de G dont le flux est strictement négatif. D'après le théorème 2.4, le flot f' se décompose sur les flots canoniques d'une famille $C'(f')$ d'au plus m circuits élémentaires de G' . A tout circuit μ' de G' correspond un cycle μ de G et l'on a $\psi_{A',B'}(\Phi_{\mu'}) = \Phi_{\mu}$.

La propriété de décomposition de f résulte alors de la linéarité de $\psi_{A',B'}$. Si $f(u)$

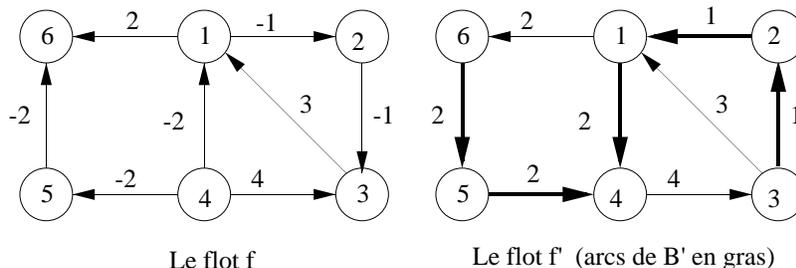


Figure 2.4: L'application $\psi_{A,B}$.

est strictement négatif (respectivement strictement positif), pour tout circuit de G' passant par l'arc u' , on a $\psi_{A',B'}(\Phi_{\mu'})(u) = -1$ (respectivement $+1$). Il en résulte la condition b) du théorème.

Sur l'exemple de la figure 2.4, le flot positif f' est égal à $2\chi_{\mu_1} + \chi_{\mu_2} + \chi_{\mu_3}$ où $\mu_1 = (1, 4, 3, 1)$, $\mu_2 = (1, 6, 5, 4, 3, 1)$ et $\mu_3 = (1, 6, 5, 4, 3, 2, 1)$ sont trois circuits de G' . Il en résulte que le flot f est égal à $2\phi_{\gamma_1} + \phi_{\gamma_2} + \phi_{\gamma_3}$ où γ_1 , γ_2 et γ_3 sont les trois cycles de G associés aux circuits μ_1 , μ_2 et μ_3 de G' . ■

Conditions d'optimalité d'un flot compatible

Soit $R = (G, a, b, c)$ un réseau valué consistant. La notion de *cycle augmentant* est utile pour comparer les coûts de deux flots compatibles de ce réseau. Un cycle élémentaire μ est dit *augmentant* pour le flot compatible f s'il existe un entier positif k tel que $a \leq f + k\phi_\mu \leq b$. Si de plus le *coût unitaire* $c \star \phi_\mu$ est strictement négatif, le cycle μ est dit *améliorant*.

Proposition 2.6. *Soit R un réseau valué et soient f et g deux flots compatibles de ce réseau. Le flot $f - g$ se décompose sur un sous-ensemble d'au plus m cycles augmentant de g .*

Preuve. Soit $h = f - g$. D'après le théorème 2.5, il existe r ($r \leq m$) cycles élémentaires μ_1, \dots, μ_r de G et r nombres entiers strictement positifs k_1, \dots, k_r tels que : $h = \sum_{i=1}^r k_i \phi_{\mu_i}$. Comme $f = g + h$, nous avons :

$$a \leq g + \sum_{i=1}^r k_i \phi_{\mu_i} \leq b.$$

Soient μ_j l'un de ces cycles et u un arc de μ_j . D'après le théorème 2.5, tous les nombres $\phi_{\mu_i}(u)$, $i \in \{1, \dots, r\}$ sont de même signe. S'ils sont tous positifs ou nuls on a :

$$a(u) \leq g(u) + k_j \phi_{\mu_j}(u) \leq b(u)$$

car $g(u) \geq a(u)$ et $k_j \phi_{\mu_j}(u) \leq \sum_{i=1}^r k_i \phi_{\mu_i}$. S'ils sont tous négatifs ou nuls on a la même inégalité car $g(u) \leq b(u)$ et $k_j \phi_{\mu_j}(u) \geq \sum_{i=1}^r k_i \phi_{\mu_i}$. ■

La proposition 2.6 nous permet maintenant de caractériser un flot compatible de coût minimum.

Théorème 2.7. *Un flot compatible f est de coût minimum si et seulement s'il ne possède pas de cycle améliorant.*

Preuve. La condition nécessaire est immédiate. Soit f un flot compatible et soit g un flot compatible de coût strictement inférieur. Posons $h = g - f$. D'après la proposition 2.6, il existe des cycles μ_1, \dots, μ_r augmentant pour f et des nombres entiers positifs k_1, \dots, k_r tels que :

$$c \star g = c \star f + \sum_{i=1}^r k_i (c \star \phi_{\mu_i}).$$

Il existe donc un j tel que $c \star \phi_{\mu_j}$ soit strictement négatif et le cycle μ_j est améliorant pour f . Contradiction. ■

Soient f un flot compatible et μ un cycle de coût unitaire non nul. Si le flux de chaque arc du cycle n'est pas égal à l'une de ses deux bornes, le flot f n'est pas de coût minimum. Il est possible en effet soit d'augmenter soit de diminuer d'une

même quantité strictement positive tous les flux des arcs de μ et l'une de ces deux transformations conduit à un flot compatible strictement meilleur.

Un arc u de A est dit *libre* vis-à-vis du flot f si $a(u) < f(u) < b(u)$, dans le cas contraire il est dit *bloqué*. Un cycle élémentaire est *libre* si tous ses arcs sont libres.

Proposition 2.8. *Un réseau valué consistant possède un flot de coût minimum sans cycles libres. Le graphe partiel des arcs libres peut être complété par des arcs bloqués pour former un arbre (appelé arbre-base).*

Preuve. Soit R un réseau valué consistant et f un flot compatible de coût minimum. S'il existe un cycle γ libre, ce cycle est de coût nul. Posons alors $\alpha = \min\{f(u) - a(u) \mid u \in \gamma^+\}$, $\beta = \min\{b(u) - f(u) \mid u \in \gamma^-\}$, $\epsilon = \min\{\alpha, \beta\}$ et $g = f - \epsilon\Phi_\gamma$. La fonction g est un flot compatible de coût minimum de R qui possède au moins un arc libre de moins que f . En répétant cette transformation, on aboutit à un flot compatible de coût minimum sans cycle libre.

Dominance des flots entiers

Soit $R = (G, a, b, c)$ un réseau valué consistant. Nous montrons qu'il existe un flot compatible de coût minimum dont tous les flux sont entiers. On dit encore que les flots entiers sont *dominants*. La preuve s'appuie sur une propriété de la matrice d'incidence sommets-arcs d'un *arbre orienté*, c'est-à-dire un graphe obtenu par orientation des arêtes d'un arbre (voir figure 2.5), et sur la proposition 2.8.

Proposition 2.9. *Soit $H = (T, B)$ un arbre orienté. Le graphe H possède une matrice d'incidence triangulaire inférieure dont tous les éléments diagonaux sont non nuls.*

Preuve. Nous raisonnons par récurrence sur $n = \text{Card}(T)$. La propriété est vraie pour $n = 2$. Soit H un arbre orienté à n sommets ($n > 2$), t une feuille de H , u l'arc incident à cette feuille et H_t le sous-graphe induit par $T - \{t\}$. Le graphe H_t est un arbre orienté à $n - 1$ sommets qui possède (par induction) une matrice d'incidence triangulaire inférieure M_t satisfaisant la proposition. La matrice d'incidence M de H obtenue en ajoutant à M_t la colonne u et le sommet t comme première ligne et première colonne satisfait également la proposition. ■

La figure 2.5 illustre la construction d'une telle matrice d'incidence.

Proposition 2.10. *Un réseau valué consistant possède un flot de coût minimum dont tous les flux sont entiers.*

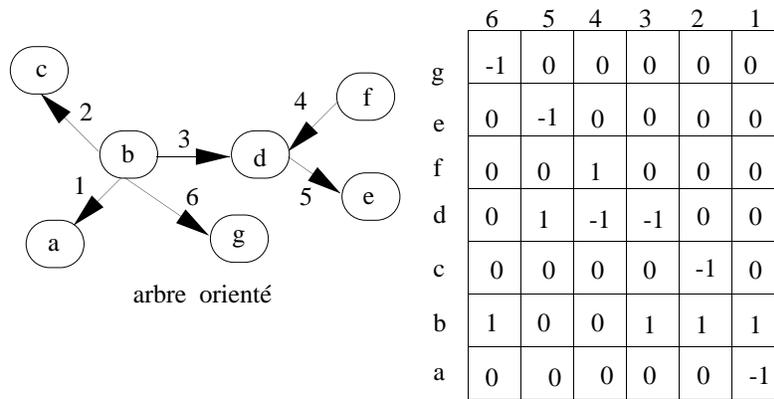


Figure 2.5: Matrice d'incidence d'un arbre orienté.

Preuve. Soient R un réseau valué consistant. D'après la proposition 2.8, il existe un flot compatible f de coût minimum sans cycle libre et un arbre-base pour f dont les arcs constituent l'ensemble H . Nous notons alors K l'ensemble des arcs du coarbre. D'après la proposition 2.9, il existe une matrice d'incidence M de G composée d'une matrice d'incidence de H notée M_H vérifiant les propriétés de la proposition 2.9 et d'une matrice d'incidence de K notée M_K . La condition d'équilibre satisfaite par le flot compatible f s'écrit (en utilisant les notations habituelles) :

$$M_H f_H = -M_K f_K.$$

Le vecteur f_K est entier puisque les arcs de K sont bloqués pour le flot compatible f . Le vecteur f_H , solution unique de l'équation d'équilibre, est donc également entier. ■

Il résulte de la proposition 2.10 que la recherche d'un flot compatible de coût minimum peut être réalisée dans le sous ensemble des flots compatibles entiers du réseau.

8.3 Problème du flot maximum

Un problème de *flot maximum* noté (G, b, s, p) est spécifié par la donnée d'un graphe orienté $G = (S, A)$ connexe sans boucles, de deux sommets distincts s et p appelés respectivement *source* et *puits* et d'une fonction capacité maximale b telle que pour tout arc u de A , la capacité $b(u)$ soit un entier strictement positif. Un *flot de s à p* est une fonction g de $\mathcal{F}(A)$ satisfaisant les contraintes (D) ci-dessous :

$$\begin{aligned}
 e_g(s) &\leq 0 \\
 e_g(p) &\geq 0 \\
 0 &\leq g \leq b \\
 \forall x \in S - \{s, p\}, \quad e_g(x) &= 0.
 \end{aligned}
 \tag{D}$$

La *valeur* de g est définie par $\hat{g}=e_g(p)$ et le problème est la recherche d'un flot g de s à p de valeur maximum.

En munissant le graphe G d'un arc *de retour* noté u_0 d'origine p , d'extrémité s et de capacité maximale $b^-(p)$, on obtient un réseau associé au problème (G, b, s, p) . Les flots compatibles de ce réseau sont en bijection avec les fonctions g solutions de (D) . De plus si f est le flot du réseau associé à g dans la bijection, on a $e_g(p) = f(u_0)$. Le problème revient donc à déterminer un flot compatible du réseau associé dont le flux de l'arc de retour est maximum.

Nous emploierons dans la suite le terme *flot* pour désigner un flot compatible du réseau associé.

Le problème du flot maximum a été très étudié et a donné lieu à des algorithmes de plus en plus efficaces depuis les travaux de Ford et Fulkerson. On peut regrouper ces algorithmes en deux classes, les méthodes primales qui construisent une suite de flots de valeur croissante et les méthodes duales qui construisent une suite de flots approchés, appelés *préflots*, dont le dernier est un flot maximum. Nous présentons d'abord l'algorithme «générique» de Ford et Fulkerson, puis un algorithme efficace représentatif de chacune des classes, l'algorithme primal des distances estimées au puits et l'algorithme dual du préflot. Nous terminons par deux variantes très efficaces de l'algorithme du préflot, l'algorithme de Karzanov et l'algorithme des excès échelonnés.

Nous commençons par une simplification naturelle du réseau. Soit (G, b, s, p) un problème de flot maximum de s à p . Comme nous l'avons vu dans la section 8.2.2, le problème du flot de valeur maximum de s à p est un cas particulier du problème du flot compatible de coût minimum. Il existe donc d'après la proposition 2.10 un flot f^* de valeur maximum dont tous les flux sont entiers. D'après le théorème 2.4, le flot f^* est une combinaison linéaire à coefficients entiers strictement positifs des flots canoniques de circuits de G . Or seuls les circuits de la décomposition passant par l'arc de retour u_0 contribuent à la valeur de f^* . Il en résulte que le flot obtenu en ne retenant dans la décomposition de f^* que les circuits passant par u_0 est aussi de valeur maximum. Il existe donc un flot de valeur maximum qui est une combinaison linéaire à coefficients entiers strictement positifs de circuits passant par l'arc de retour u_0 .

Soit x un sommet qui n'appartient pas à un chemin de s à p . Le graphe obtenu en supprimant x et tous les arcs adjacents à x contient encore tous les circuits de G passant par u_0 . La valeur maximum d'un flot de s à p dans le nouveau graphe est donc la même que celle associée au graphe G . Nous pouvons donc supposer sans perte de généralité que tous les sommets du graphe G sont accessibles de s et co-accessibles de p .

La figure 3.1 présente le réseau associé à un problème de flot maximum et un flot f de ce réseau. À côté de chaque arc u sont inscrites sa capacité maximale $b(u)$ (entre parenthèses) et le flux $f(u)$.

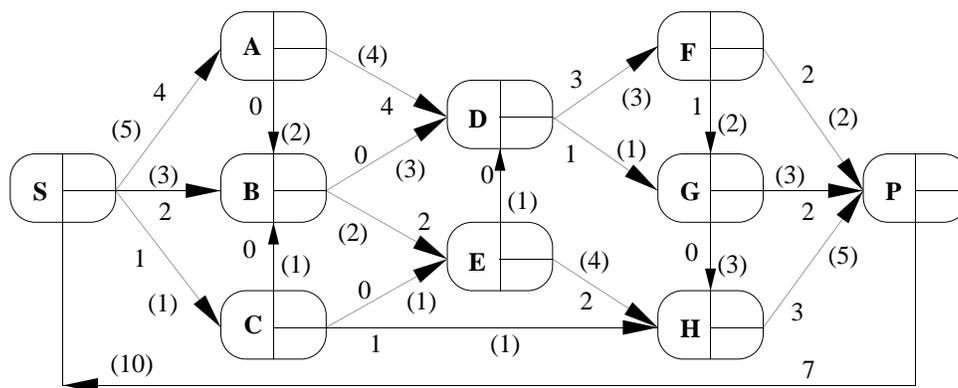


Figure 3.1: Un problème de flot maximum et l'un de ses flots.

Graphe d'écart d'un flot f

Les algorithmes de résolution du problème du flot maximum sont fondés sur la recherche de chaînes dites *améliorantes* le long desquelles sont réalisées des augmentations de flux. Comme nous le verrons, l'efficacité de certains algorithmes augmente si les améliorations de flot sont réalisées sur des chaînes de longueur minimum en nombre d'arcs. Pour ramener la recherche d'une chaîne améliorante à celle d'un chemin, on définit un outil très utile : le graphe d'écart. Soit R le réseau d'un problème de flot maximum (G, b, s, p) et soit f un flot de ce réseau. Le *graphe d'écart* d' $G_f = (S, A_f)$ du flot f est un graphe *valué* ayant les mêmes sommets que G et dont les arcs sont définis à partir des arcs de G par la procédure GRAPHE-D'ÉCART suivante :

```

procédure GRAPHE-D'ÉCART( $G, f$ );
   $A_f := \emptyset$ ;
  pour tout arc  $u$  de  $G$  faire
    si  $f(u) < b(u)$ 
      alors insérer dans  $A_f$  l'arc  $u'$  valué par  $r(u') = b(u) - f(u)$ 
        d'origine  $u^-$  et d'extrémité  $u^+$ 
      finsi;
    si  $f(u) > 0$ 
      alors insérer dans  $A_f$  l'arc  $u''$  valué par  $r(u'') = f(u)$ 
        d'origine  $u^+$  et d'extrémité  $u^-$ ;
      finsi;
  finpour;
  retourner( $A_f$ ).

```

Un arc u libre pour f produira donc les deux arcs u' et u'' dans G_f , un arc u saturé produira seulement l'arc u'' et un arc u vide produira seulement l'arc u' .

Un arc u' est appelé *représentant conforme* de l'arc u , un arc u'' est appelé *représentant non conforme* de l'arc u . Les arcs u' et u'' sont dits *conjointes*. La figure 3.2 montre le graphe d'écart du flot de la figure 3.1.

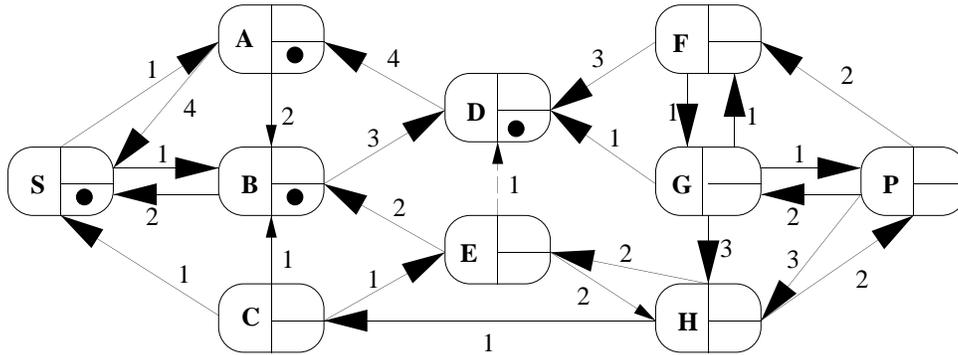


Figure 3.2: Un graphe d'écart.

Donnons quelques propriétés simples du graphe d'écart. Tout arc v du graphe d'écart est le représentant conforme ou non conforme d'un arc unique du graphe G appelé son *père*. La notation condensée $v = u'$ (respectivement $v = u''$) signifiera que l'arc v de G_f est le représentant conforme (respectivement non conforme) de l'arc u de G . Un arc u de G a donc pour le flot f un ou deux *représentants* dans G_f . Chacun de ces représentants a une valuation strictement positive et la somme des valuations des représentants de u est toujours égale à $b(u)$.

Soit u un arc de G . Faisons varier son flux $f(u)$ de 0 à $b(u)$. Si $f(u) = 0$, l'arc u' existe et l'arc u'' n'existe pas dans G_f . Si $f(u)$ augmente mais n'atteint pas la valeur $b(u)$, l'arc u'' apparaît dans G_f , enfin lorsque $f(u) = b(u)$, l'arc u' disparaît de G_f . Une augmentation de flux est donc susceptible de faire apparaître et/ou disparaître un arc du graphe d'écart. Il en est bien sûr de même d'une diminution de flux.

La fonction fondamentale du graphe d'écart G_f est de ramener la recherche d'un flot meilleur que f à celle d'un chemin de s à p dans G_f . Un tel chemin est appelé *chemin améliorant*. Soit μ un chemin améliorant et α la plus petite valuation des arcs de ce chemin, le lemme 3.1 montre que la nouvelle fonction f calculée par la procédure AUGMENTER-FLOT ci-dessous est un flot strictement meilleur.

```

procédure AUGMENTER-FLOT( $f, \mu$ );
   $\alpha := \min\{r(v) \mid v \text{ arc de } \mu\}$ ;
  pour chaque arc  $v$  de  $\mu$  faire
    si  $v = u'$  alors  $f(u) := f(u) + \alpha$  finfaire
    si  $v = u''$  alors  $f(u) := f(u) - \alpha$  finfaire
  finpour
   $f(u_0) := f(u_0) + \alpha$ .

```

Lemme 3.1. *S'il existe un chemin améliorant dans G_f , le flot f n'est pas de valeur maximum.*

Preuve. Soit μ un chemin améliorant du graphe d'écart G_f . Notons α ($\alpha > 0$) la plus petite valuation des arcs de ce chemin. Appelons μ^+ (respectivement μ^-) les arcs conformes (respectivement non conformes) du chemin μ . Si nous notons g la nouvelle fonction f à l'issue de la procédure AUGMENTER-FLOT, g est un flot par définition des arcs conformes et non conformes et nous avons $\hat{g} = \hat{f} + \alpha > \hat{f}$ puisque $\alpha > 0$. ■

Il est important de remarquer que la procédure AUGMENTER-FLOT supprime au moins un arc du graphe d'écart G_f . En effet, si α est la valuation d'un arc $v = u'$ de μ^+ , on a $\alpha = b(u) - f(u)$ et $g(u) = b(u)$. L'arc v de G_f n'appartient donc pas à G_g . Si α est la valuation d'un arc $v = u''$ de μ^- , on a $\alpha = f(u)$ et $g(u) = 0$. L'arc v de G_f n'appartient donc pas à G_g .

8.3.1 L'algorithme générique de Ford et Fulkerson

Tous les algorithmes de résolution du problème de flot maximum utilisent comme critère de terminaison un théorème important, dû à Ford et Fulkerson, que l'on peut rapprocher du théorème de la dualité en programmation linéaire continue.

Les objet duaux des flots sont appelés *coupes*. Une coupe est un sous-ensemble C d'arcs de G tel que $C = \omega^+(T)$ où T est un sous-ensemble de sommets contenant s et ne contenant pas p . La valeur d'une coupe est par définition $b(C)$. La figure 3.3 montre une coupe du réseau de la figure 3.1. Cette coupe, constituée des arcs de $\omega^+\{S, A, B, C\}$, a la valeur 10. Au problème de recherche d'un flot de valeur

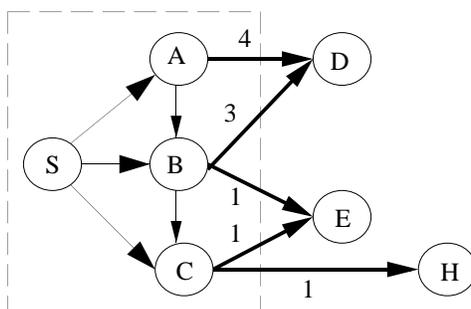


Figure 3.3: Une coupe.

maximum correspond par dualité le problème de recherche d'une coupe de valeur minimum. Le théorème de Ford et Fulkerson montre que ces deux valeurs sont égales.

Théorème 3.2 (de Ford et Fulkerson). *La valeur maximum d'un flot est égale à la valeur minimum d'une coupe.*

Preuve. Nous montrons d'abord que la valeur $b(C)$ d'une coupe $C = \omega^+(T)$ ($s \in T, t \notin T$) est supérieure ou égale à la valeur \hat{f} d'un flot f . En effet, par sommation des équations d'équilibre sur les sommets de T , il vient :

$$f^-(T) = f(u_0) + \sum_{u \in \omega^-(T) - \{u_0\}} f(u) = f^+(T) \leq b(C).$$

Il en résulte : $\hat{f} \leq b(C)$.

Supposons maintenant que f soit un flot maximum. D'après le lemme 3.1, il n'existe pas de chemin de s à p dans le graphe d'écart G_f . Notons alors T l'ensemble des sommets accessibles à partir de s dans G_f . Un arc u de G sortant de T est saturé car dans le cas contraire, son représentant conforme serait un arc sortant de T dans G_f . Un arc u de G entrant dans T est vide car dans le cas contraire, son représentant non conforme serait, lui, un arc sortant de T dans G_f . Il en résulte que :

$$f^-(T) = f(u_0) = f^+(T) = b(C).$$

La coupe C est donc de valeur minimum. ■

On déduit directement du lemme 3.1 et du théorème 3.2 la condition nécessaire et suffisante suivante :

Proposition 3.3. *Un flot f est maximum si et seulement s'il n'existe pas de chemin améliorant dans G_f .*

Le théorème précédent conduit très naturellement à un algorithme générique, dû à Ford et Fulkerson.

```

procédure FORD-FULKERSON ( $G, b, s, p$ );
   $f := 0$ ;
  tantque  $p$  est accessible à partir de  $s$  dans  $G_f$  faire
    déterminer un chemin améliorant  $\mu$  de  $G_f$ ;
    AUGMENTER-FLOT ( $f, \mu$ )
  fintantque.

```

Cet algorithme se termine puisqu'à chaque itération la valeur du flot, majorée par $b^-(p)$, augmente au moins d'une unité. D'après le théorème de Ford et Fulkerson, le dernier flot obtenu est de valeur maximum. Le graphe d'écart de la figure 3.2 montre que le flot est maximum puisque l'ensemble des sommets accessibles à partir de S , pointés d'un rond noir, ne contient pas P .

En laissant totalement libre le choix du chemin améliorant, l'algorithme précédent peut réaliser un très grand nombre d'itérations. Sur l'exemple de la figure 3.4,

l'algorithme peut, à partir du flot nul, augmenter la valeur du flot alternativement sur le chemin $(1, 2, 4, 3)$ et sur la chaîne $(1, 4, 2, 3)$. Le flot maximum est alors obtenu après M itérations alors que deux itérations suffisent. Pour transformer cet algorithme «générique» en un algorithme efficace, il faut contraindre le choix du chemin améliorant. Un critère judicieux, dû à Edmonds et Karp, consiste à choisir un chemin améliorant de longueur minimum (en nombre d'arcs). Nous montrerons en effet que la longueur d'un *plus court chemin améliorant* ne peut pas croître d'une itération sur l'autre pourvu que l'augmentation de flot soit réalisée le long d'un tel chemin. De plus, après au plus m itérations (où m est le nombre d'arcs de G), cette diminution est stricte.

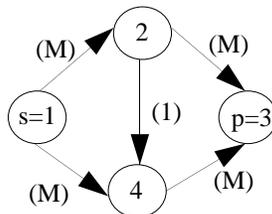


Figure 3.4: Un problème de flot maximum.

8.3.2 L'algorithme des distances estimées au puits

La faiblesse essentielle de l'algorithme générique est de ne retenir aucune information sur les chemins améliorants du graphe d'écart d'une itération sur l'autre alors que deux graphes d'écart successifs sont souvent très voisins. Le principe de l'algorithme des distances estimées au puits est de calculer, en chaque sommet du graphe d'écart, une évaluation par défaut de sa distance au puits. On peut alors calculer efficacement un *plus court chemin améliorant*.

Distance estimée et graphe d'admissibilité

Soient $G_f = (S, A_f)$ le graphe d'écart d'un flot f et n le nombre de sommets de G_f . Nous appelons *distance estimée* une application $\Delta : S \rightarrow \mathbb{N}$ telle que :

- a) $\Delta(p) = 0$;
- b) pour tout arc v de G_f , $\Delta(v^-) \leq \Delta(v^+) + 1$.

Si le sommet p est accessible à partir du sommet x dans G_f , la distance estimée du sommet x est une évaluation par défaut de la longueur minimum (en nombre d'arcs) d'un chemin de x à p dans G_f . Il en résulte que si $\Delta(y) \geq n$ pour $y \in S$, il n'existe pas de chemin de y à s dans G_f .

Un arc v de G_f est dit *admissible* si $\Delta(v^-) = \Delta(v^+) + 1$. Un *chemin admissible* est un chemin améliorant dont les arcs sont admissibles. Un chemin admissible est donc un plus court chemin améliorant. Nous appellerons *graphe d'admissibilité*

le graphe partiel de G_f formé des arcs admissibles. Le graphe d'admissibilité ne possède pas de circuits et aucun arc admissible n'a le sommet p comme origine.

Initialisation

Le flot initial est le flot nul et la distance estimée initiale d'un sommet x est la longueur minimum d'un chemin de x à p dans G_0 . Notons que cette longueur est définie pour tout sommet x puisque tout sommet est par hypothèse co-accessible de p . Le *chemin courant* initial est réduit au sommet s . La procédure INITIALISER-DEP réalise cette initialisation.

```
procédure INITIALISER-DEP( $G, b, s, p$ );
   $\mu := (s)$ ;  $f := 0$ ;  $z := s$ ;
  pour tout sommet  $x$  de  $G$  faire  $\Delta(x) := l(x, p, G_0)$ .
```

On note $l(x, p, G_0)$ la longueur minimum d'un chemin de x à p dans G_0 , μ le chemin courant, f le flot courant, et z l'extrémité de μ .

Itération courante

Soit μ le chemin courant et z son extrémité. Une itération de l'algorithme consiste à prolonger μ à partir de son extrémité dans le graphe d'admissibilité jusqu'à ce que l'extrémité z du chemin soit une sortie du graphe d'admissibilité.

Si $z = p$, le chemin μ est un plus court chemin améliorant. On améliore alors le flot le long de ce chemin, on ne modifie pas la distance estimée Δ et l'on réinitialise le chemin courant μ à (s) .

Si $z \neq p$, on augmente *strictement* la distance estimée du sommet z et on supprime le dernier arc de μ qui n'est plus admissible.

La procédure ITÉRER-DEP ci-dessous réalise l'itération courante.

```
procédure ITÉRER-DEP;
  PROLONGER( $\mu$ );
  soit  $z$  l'extrémité de  $\mu$ ;
  si  $z = p$ 
    alors AUGMENTER-FLOT( $f, \mu$ );  $\mu := (s)$ 
    sinon AUGMENTER-DISTANCE( $z$ ); SUPPRIMER-DERNIER-ARC( $\mu$ )
  fin.
```

La procédure $\text{PROLONGER}(\mu)$ transforme le chemin μ en un chemin μ' du graphe d'admissibilité qui admet μ comme sous-chemin initial et dont l'extrémité z n'est pas l'origine d'un arc admissible. Sa complexité est de l'ordre du nombre d'arcs ajoutés.

Lorsque μ est un chemin améliorant, la procédure AUGMENTER-FLOT améliore le flot le long du chemin μ et met à jour le graphe d'écart. Comme le chemin μ est élémentaire, il comporte au plus $n - 1$ arcs. La complexité de la procédure AUGMENTER-FLOT est donc $O(n)$.

Lorsque μ n'est pas un chemin améliorant, la procédure $\text{AUGMENTER-DISTANCE}$ augmente la valeur de $\Delta(z)$ comme suit :

$$\Delta(z) := \begin{cases} n & \text{si } S_f(z) \text{ est vide} \\ \min_{y \in S_f(z)} \{1 + \Delta(y)\} & \text{sinon} \end{cases}$$

($S_f(z)$ est l'ensemble des successeurs de z dans G_f d'origine z) et met à jour le graphe d'admissibilité. La figure 3.5, où les distances estimées sont inscrites à côté de chaque sommet et où les arcs de μ sont épais, montre cette mise à jour. Si le sommet z a k successeurs dans G , la complexité de $\text{AUGMENTER-DISTANCE}$ est $O(k)$.

Après l'exécution de $\text{AUGMENTER-DISTANCE}$, le dernier arc de μ n'est plus admissible. La procédure $\text{SUPPRIMER-DERNIER-ARC}$ supprime le dernier arc de μ si la longueur de μ est strictement positive et laisse μ inchangé sinon. Le nouveau chemin μ est donc admissible. La complexité de la procédure $\text{SUPPRIMER-DERNIER-ARC}$ est $O(1)$.

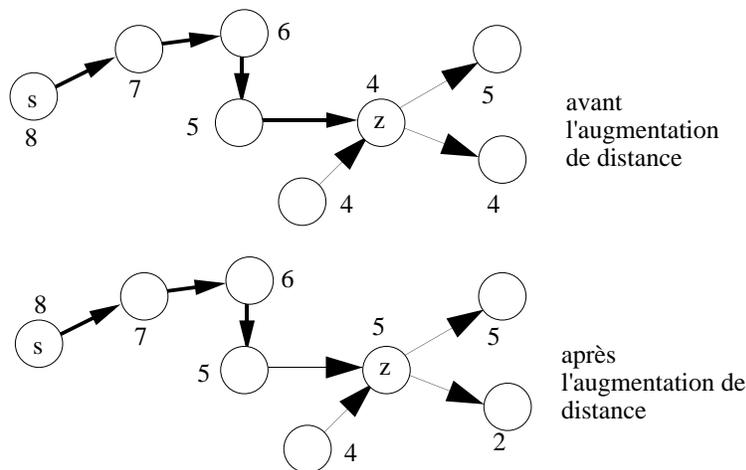


Figure 3.5: Mise à jour de la distance estimée

Terminaison

L'algorithme se termine lorsque la distance estimée du sommet s est supérieure ou égale à n . Par définition de la distance estimée, il n'existe plus alors de chemin

améliorant et le flot obtenu est maximum. Il en résulte la procédure DEP ci-dessous :

```

procédure DEP ;
  INITIALISER-DEP
  tantque  $\Delta(s) < n$  faire
    ITÉRER-DEP
  fintantque.

```

Les figures 3.6, 3.7 et 3.8 illustrent le déroulement de l'algorithme des distances estimées au puits sur le réseau de la figure 3.1. La figure 3.6 montre en (A) le graphe d'écart du flot nul initial, les distances estimées (coin nord-est du sommet) et le graphe d'admissibilité (arcs épais). L'algorithme réalise ensuite une augmentation de flot pour le chemin admissible (S, C, H, P) puis augmente la distance estimée du sommet S . Le graphe d'écart à l'issue de cette première augmentation de la distance estimée est représenté en (B). Suivent ensuite trois augmentations de flot pour les chemins (S, A, D, F, P) , (S, B, E, H, P) et (S, A, D, G, P) à l'issue desquelles la valeur du flot est 6 et le graphe d'écart celui de la figure 3.7. L'algorithme augmente alors la distance du sommet F et réalise ensuite une augmentation de flot sur le chemin (S, A, D, F, G, P) . Le flot est alors optimal comme le montre le graphe d'écart de la figure 3.8 où les sommets accessibles à partir de S sont pointés. L'algorithme n'est cependant pas terminé car il reste encore des augmentations de distance à exécuter pour que la distance estimée de S devienne supérieure ou égale à $n = 10$.

Convergence et complexité

Après avoir décrit l'algorithme des distances estimées, nous montrons maintenant que cet algorithme se termine, qu'il calcule un flot de valeur maximum et que sa complexité est $O(n^2m)$.

Quelques propriétés seront utiles pour parvenir à ces résultats. Considérons la suite des fonctions Δ calculées par la procédure DEP. On note Δ_0 la fonction Δ initiale et Δ_k la fonction Δ à l'issue de l'itération k .

Proposition 3.4. *La suite Δ_k est une suite croissante de fonctions « distance estimée ». Si l'itération k réalise une augmentation de flot, alors $\Delta_k = \Delta_{k-1}$; si elle réalise une augmentation de distance, alors $\Delta_k > \Delta_{k-1}$.*

Preuve. Considérons le cas d'une augmentation de flot et notons g le nouveau flot. Si l'ensemble des arcs du nouveau graphe d'écart G_g est inclus dans l'ensemble des arcs de G_f , la fonction Δ est toujours une distance estimée pour G_g .

Supposons que le changement de flux du père u d'un arc $v = u'$ de μ crée l'arc u'' dans G_g . L'arc v étant admissible, on a $\Delta(v^-) = \Delta(v^+) + 1$ et donc $\Delta(v^+) =$

$\Delta(v^-) - 1 \leq \Delta(v^-) + 1$. Il en est de même si le changement de flux du père u d'un arc $v = u''$ de μ crée l'arc u' dans G_g . La fonction Δ respecte donc les deux conditions d'une distance estimée pour G_g .

Considérons le cas d'une augmentation de la distance estimée du sommet z due à la procédure AUGMENTER-DISTANCE et notons Δ' la nouvelle fonction Δ . Pour tout sommet distinct de z on a $\Delta'(z) = \Delta(z)$ et pour le sommet z nous avons :

$$\Delta'(z) := \begin{cases} n & \text{si } S_f(z) \text{ est vide} \\ \min_{y \in S_f(z)} \{1 + \Delta(y)\} & \text{sinon} \end{cases}$$

Il en résulte que $\Delta'(z)$ est strictement supérieur à $\Delta(z)$. De plus, pour tout sommet x successeur de z on a $\Delta'(z) \leq \Delta'(x) + 1$ et pour tout prédécesseur y de z on a $\Delta'(y) < \Delta'(z) + 1$. La fonction Δ' est donc une nouvelle distance estimée pour G_f strictement supérieure à Δ . ■

Pour prouver que l'algorithme se termine nous utilisons le lemme 3.5 qui permet de majorer le nombre d'exécutions de la procédure AUGMENTER-FLOT entre deux exécutions de la procédure AUGMENTER-DISTANCE.

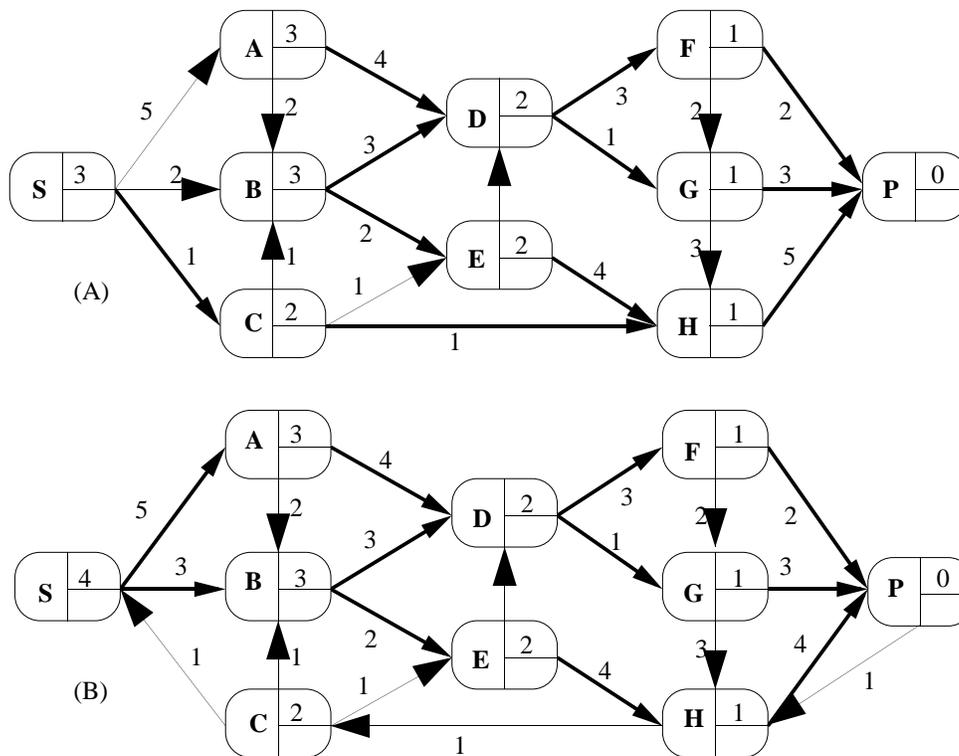
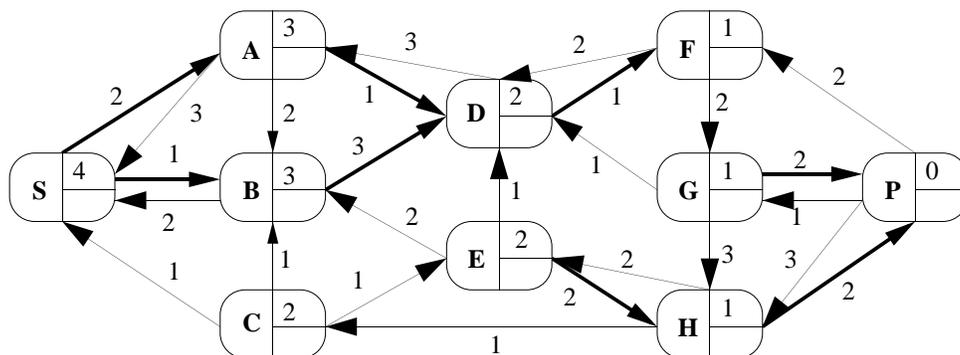
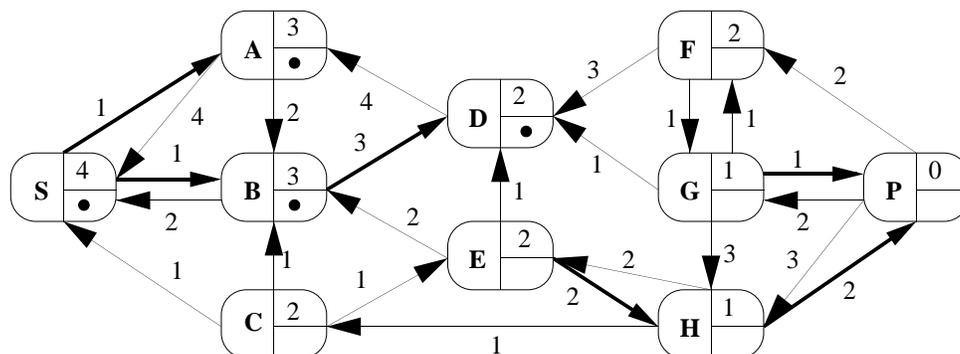


Figure 3.6: Initialisation et première augmentation de distance.

Lemme 3.5. *Entre deux suppressions successives d'un arc v du graphe d'écart, la distance estimée de v^+ a augmenté d'au moins deux unités.*

Figure 3.7: *Seconde augmentation de distance.*Figure 3.8: *Premier graphe d'écart d'un flot maximum.*

Preuve. Notons Δ_1 la distance estimée lors de la première suppression de l'arc v , on a $\Delta_1(v^-) = \Delta_1(v^+) + 1$ puisque v est admissible. Notons Δ_2 la distance estimée lors de la création suivante de v dans un graphe d'écart. Comme cette création ne peut avoir lieu que si l'arc conjoint de v appartient à un chemin admissible améliorant, on aura alors $\Delta_2(v^+) = \Delta_2(v^-) + 1$. Notons enfin Δ_3 la distance estimée lors de la suppression suivante de v , nous avons :

$$\Delta_3(v^+) \geq \Delta_2(v^+) = \Delta_2(v^-) + 1 \geq \Delta_1(v^-) + 1 = \Delta_1(v^+) + 2$$

■

Nous sommes maintenant en mesure de prouver la validité de l'algorithme des distances estimées au puits et de calculer sa complexité.

Théorème 3.6. *La procédure DEP calcule un flot maximum en temps $O(n^2m)$.*

Preuve. D'après le lemme 3.5, un même arc peut être supprimé d'un graphe d'écart au plus $\lceil n/2 \rceil$ fois. En effet, si l'arc v était supprimé au moins $\lceil n/2 \rceil + 1$ fois, la distance estimée de v^+ au début de la dernière augmentation de distance serait supérieure ou égale à $2\lceil n/2 \rceil \geq n$. Or comme v^+ est accessible à partir de s , on aurait $\Delta(s) \geq n$ en début d'itération. D'où la contradiction. Comme chaque

exécution de la procédure AUGMENTER-FLOT supprime au moins un arc, elle est appelée au plus $2m\lceil n/2\rceil$ fois.

Il en résulte que l'algorithme se termine puisque toute itération réalise soit une augmentation de flot soit une augmentation de distance. Lors de la terminaison, on a par définition $\Delta(s) \geq n$. Le dernier graphe d'écart ne contenant pas de chemin améliorant, le dernier flot est de valeur maximum.

Pour évaluer la complexité de la procédure DEP, nous allons suivre l'évolution de la longueur $L(\mu)$ du chemin courant μ . Cette longueur croît lors de chaque exécution de PROLONGER, décroît d'une unité au plus lors de l'exécution de AUGMENTER-DISTANCE, et devient nulle lors de l'exécution de AUGMENTER-FLOT. La complexité de la procédure PROLONGER est proportionnelle au nombre d'arcs ajoutés, c'est à dire à l'accroissement de $L(\mu)$. Comme le nombre d'exécutions de la procédure AUGMENTER-FLOT est au plus $2m\lceil n/2\rceil$ et la longueur $L(\mu)$ est au plus n , la somme des variations négatives de $L(\mu)$ dues aux exécutions de AUGMENTER-FLOT est majorée par $2nm\lceil n/2\rceil$. Comme AUGMENTER-DISTANCE est exécutée au plus n fois par sommet, la somme des variations négatives de $L(\mu)$ dues aux exécutions de AUGMENTER-DISTANCE est majorée par n^2 . Il en résulte que la somme cumulée des variations négatives de $L(\mu)$ est majorée par $n^2 + 2nm\lceil n/2\rceil$. Comme la longueur initiale de μ est nulle et sa longueur terminale inférieure ou égale à n , la somme cumulée des variations positives de $L(\mu)$ est majorée par $n + n^2 + 2nm\lceil n/2\rceil$. La complexité de toutes les exécutions de la procédure PROLONGER est donc $O(n^2m)$. La complexité d'une exécution de AUGMENTER-FLOT est $O(n)$, donc la complexité de toutes les exécutions de la procédure AUGMENTER-FLOT est aussi $O(n^2m)$. La complexité d'une exécution de AUGMENTER-DISTANCE est de l'ordre du nombre de successeurs de z dans G et la distance estimée d'un sommet est augmentée au plus n fois, donc la complexité de toutes les exécutions de la procédure AUGMENTER-DISTANCE est $O(nm)$. Enfin la complexité $O(1)$ de la procédure SUPPRIMER-DERNIER-ARC est dominée par celle de AUGMENTER-DISTANCE. Il en résulte que la complexité de la procédure DEP est $O(n^2m)$. ■

8.3.3 L'algorithme du préflot

L'algorithme des distances estimées au puits réalise des augmentations de flot le long de chemins améliorants et maintient de ce fait la satisfaction des conditions d'équilibre. A chaque étape on dispose ainsi d'un flot réalisable. Une idée *duale* consiste à saturer initialement les arcs sortant de s et à transporter *vers le puits* la plus grande partie des excès positifs ainsi créés en réalisant à chaque itération une réduction de l'excès d'un sommet. Dès que l'excès du sous-ensemble de sommets $S' = S - \{s, p\}$ est nul, le préflot obtenu est un flot de s à p de valeur maximum. Cet algorithme est une méthode duale car tant que l'algorithme n'est pas terminé, la fonction f n'est pas un flot de s à p .

Etant donné un problème de flot maximum (G, b, s, p) , un *préflot* est une fonction de $\mathcal{F}(A)$ telle que :

$$\begin{aligned} e_f(s) &\leq 0 \\ e_f(p) &\geq 0 \\ 0 &\leq f \leq b \\ \forall x \in S - \{s, p\}, \quad e_f(x) &\geq 0 \end{aligned}$$

Pour un préflot f , le graphe d'écart G_f , la fonction distance estimée au puits et le graphe d'admissibilité sont définis comme dans le cas d'un flot de s à p (voir section 8.3). Un sommet *actif* est un sommet *distinct de s et de p* dont l'excès est strictement positif.

Initialisation

Le préflot initial f_0 est obtenu en saturant tous les arcs sortant du sommet s et en allouant un flux nul à tous les autres arcs. La distance estimée initiale d'un sommet x de S' est la longueur minimum en nombre d'arcs d'un chemin de x à p dans le graphe d'écart G_{f_0} , elle est notée $l(x, p, G_{f_0})$. La distance estimée initiale de s est n et celle de p est nulle. La procédure INITIALISER-PRÉFLOT réalise ces initialisations.

```

procédure INITIALISER-PRÉFLOT ;
   $f := 0$  ;
  pour tout arc  $u$  sortant de  $s$  faire  $f(u) := b(u)$  finpour ;
   $\Delta(p) := 0$  ;
   $\Delta(s) := n$  ;
  pour tout sommet  $x$  de  $S'$  faire  $\Delta(x) := l(x, p, G_{f_0})$  finpour.

```

L'initialisation joue un rôle important. En effet, comme les arcs sortant de s sont initialement saturés, il n'existe pas de chemin améliorant dans le graphe d'écart initial. De plus la distance estimée du sommet s n'est jamais modifiée. Il n'existera donc jamais de chemin améliorant dans les graphes d'écart des préflots successifs.

Itération courante

Une itération de l'algorithme du préflot consiste à choisir un sommet actif x . S'il n'existe pas d'arc admissible d'origine x , la distance estimée de x au puits est augmentée strictement comme dans la procédure AUGMENTER-DISTANCE de l'algorithme des distances estimées au puits. Sinon on choisit un arc *admissible* v d'origine x et l'on réduit l'excès du sommet origine x en modifiant le flux du père u de v . La procédure RÉDUIRE-EXCÈS ci-dessous réalise cette réduction.

```

procédure RÉDUIRE-EXCÈS( $f, v$ );
  si  $v = u'$ 
    alors  $\epsilon := \min\{e_f(v^-), b(u) - f(u)\}$ ;  $f(u) := f(u) + \epsilon$ 
    sinon  $\epsilon := \min\{e_f(v^-), f(u)\}$ ;  $f(u) := f(u) - \epsilon$ 
  finsi.

```

Nous dirons qu'une réduction d'excès est *saturante* si ϵ est égal à la valuation de l'arc v choisi dans G_f . Si l'arc v est le représentant conforme de l'arc u de G (c'est-à-dire si $v = u'$), l'arc u est saturé après la réduction et v n'appartient donc pas au nouveau graphe d'écart. Si l'arc v est le représentant non conforme de l'arc u de G (c'est-à-dire si $v = u''$), l'arc u est vide après la réduction et v n'appartient donc pas au nouveau graphe d'écart. Une réduction saturante supprime donc un arc dans l'ancien graphe d'écart.

Terminaison

Si l'ensemble des sommets actifs est vide à l'issue d'une itération, le dernier préflot est un flot de s à p et il n'existe pas de chemin améliorant dans le graphe d'écart correspondant. Le critère de terminaison de l'algorithme du préflot est l'absence de sommets actifs. Il en résulte la procédure PRÉFLOT ci-dessous :

```

procédure PRÉFLOT( $G, b, s, p$ );
  INITIALISER-PRÉFLOT;
  tantque l'ensemble des sommets actifs est non vide faire
    choisir un sommet actif  $x$ ;
    s'il existe un arc admissible d'origine  $x$ 
      alors
        choisir un arc admissible  $v$  d'origine  $x$ ;
        RÉDUIRE-EXCÈS( $f, v$ )
      sinon AUGMENTER-DISTANCE( $x$ )
    finsi
  fintanque.

```

Les figures 3.9, 3.10 et 3.11 illustrent une exécution de l'algorithme sur l'exemple de la figure 3.1.

La distance estimée est dans le coin nord-est du sommet et l'excès dans le coin sud-est. La figure 3.9 montre le graphe d'écart du préflot initial. Les premières réductions d'excès codées par (sommet actif, nature, arc admissible) sont les suiv-

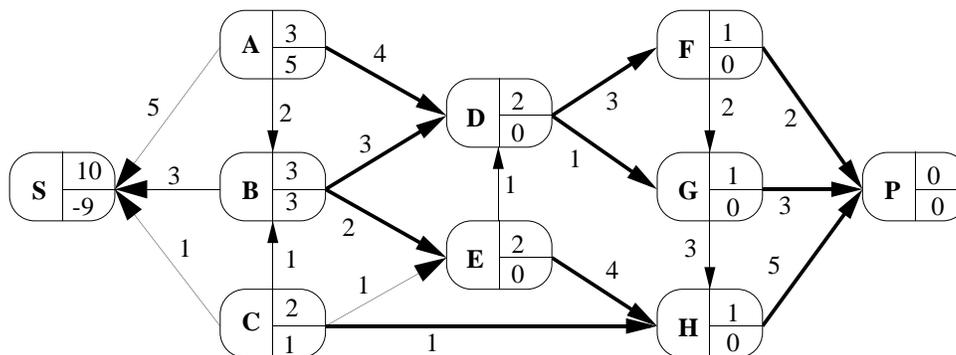


Figure 3.9: Graphe d'écart du préflot initial.

antes, ligne par ligne :

$(A, \text{sat}, (A, D))$	$(B, \text{sat}, (B, E))$	$(C, \text{sat}, (C, H))$
$(B, \text{nonsat}, (B, D))$	$(D, \text{sat}, (D, F))$	$(D, \text{sat}, (D, G))$
$(F, \text{sat}, (F, P))$	$(G, \text{nonsat}, (G, P))$	$(H, \text{nonsat}, (H, P))$
$(E, \text{nonsat}, (E, H))$	$(H, \text{nonsat}, (H, P))$	

Aucun des sommets actifs A , B ou F n'est alors l'origine d'un arc admissible. L'algorithme réalise une augmentation de distance pour F , les réductions $(F, \text{nonsat}, (F, G))$ et $(G, \text{nonsat}, (G, P))$, une augmentation de distance pour D et la réduction $(D, \text{nonsat}, (D, B))$. Il en résulte le graphe d'écart de la figure 3.10. La valeur maximum est atteinte mais certains excès sont encore positifs.

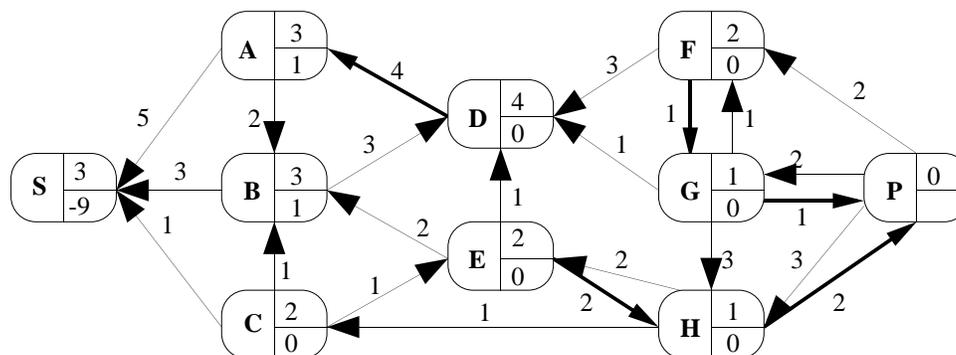


Figure 3.10: La valeur maximum est atteinte.

Tous ces excès doivent maintenant refluer vers l'entrée s . L'algorithme réalise alors des augmentations de distance jusqu'à ce que l'un des deux arcs (A,S) ou (B,S) devienne admissible. La figure 3.11 montre le dernier graphe d'écart lorsque les excès de tous les sommets de S' sont nuls.

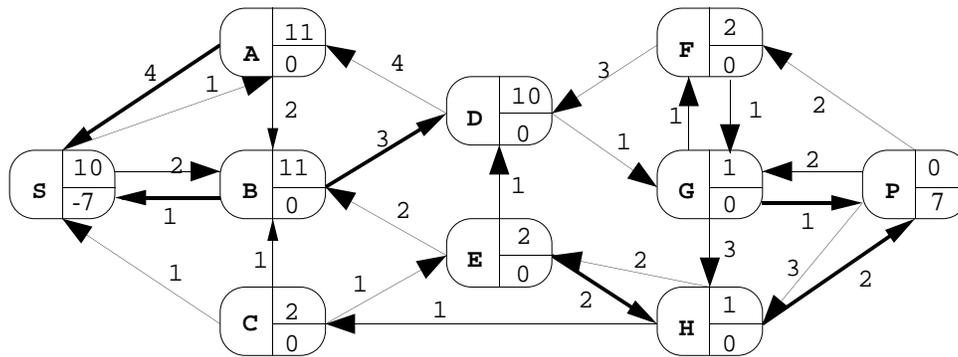


Figure 3.11: Le dernier graphe d'écart.

Convergence et complexité

Nous montrons dans cette section que l'algorithme du préflot se termine et nous calculons sa complexité. Deux lemmes préliminaires nous seront utiles.

Lemme 3.7. *La distance estimée d'un sommet actif est inférieure à $2n$.*

Preuve. Supposons le sommet y actif pour le préflot f et montrons qu'il existe alors un chemin élémentaire de s à y dans G dont tous les arcs ont un flux strictement positif. Considérons le graphe $H = (S \cup \{q\}, A \cup B)$ obtenu à partir de G en *ajoutant* un sommet q et les arcs suivants :

- pour tout sommet actif x , ajouter un arc w_{xq} d'origine x et d'extrémité q ;
- ajouter un arc w_1 d'origine p et d'extrémité q ;
- ajouter un arc w_0 d'origine q et d'extrémité s .

Définissons la fonction g de $\mathcal{F}(A \cup B)$ par :

$$g(u) = \begin{cases} f(u) & \text{si } u \in A \\ e_f(x) & \text{si } u = w_{xq} \\ -e_f(s) & \text{si } u = w_0 \\ e_f(p) & \text{si } u = w_1 \end{cases}$$

La fonction g est un flot strictement positif du graphe H car :

$$e_g(x) = \begin{cases} e_f(x) - g(w_{xq}) = 0 & \text{si } x \text{ est actif dans } S' \\ e_f(x) = 0 & \text{si } x \text{ n'est pas actif dans } S' \\ e_f(p) - g(w_1) = 0 & \text{si } x = p \\ e_f(s) + g(w_0) = 0 & \text{si } x = s \end{cases}$$

Le flot g est donc une combinaison linéaire à coefficients entiers strictement positifs de circuits de H . L'un au moins de ces circuits passe par l'arc w_{yq} et ce circuit emprunte l'arc w_0 . Soit μ le chemin élémentaire de s à y issu de ce circuit. Les arcs de μ sont des arcs de G dont les flux pour f sont strictement positifs. Il existe donc dans le graphe d'écart G_f un chemin de y à s formé par les représentants

non conformes des arcs de μ . Comme la distance estimée de s reste constante et égale à n , on a $\Delta(y) \leq \Delta(s) + (n - 1) < 2n$. ■

Le lemme 3.7 a deux conséquences importantes. Comme le nombre d'augmentations de la distance estimée *d'un même sommet* est inférieur à $2n$, le nombre total d'exécutions de la procédure AUGMENTER-DISTANCE est inférieur à $2n^2$. D'autre part le nombre d'exécutions de la procédure RÉDUIRE-EXCÈS entre deux augmentations de la distance est lui-aussi borné. En effet, cette procédure ne modifie pas la fonction distance estimée mais fait décroître d'au moins une unité le nombre $\sum_{x \in S'} e_f(x) \times \Delta(x)$. Il en résulte que l'algorithme du préflot se termine.

Nous allons maintenant évaluer le nombre d'exécutions de la procédure RÉDUIRE-EXCÈS en considérant séparément les réductions saturantes et non saturantes.

Lemme 3.8. *La procédure préflot réalise au plus $2nm$ réductions saturantes.*

Preuve. Comme dans le cas de l'algorithme des distances estimées au puits (Lemme 3.5), entre deux suppressions successives d'un même arc v du graphe d'écart, la distance estimée du sommet v^+ a augmenté d'au moins deux unités. Un même arc du graphe d'écart est donc supprimé au plus n fois (lemme 3.7). Comme une réduction saturante supprime au moins un arc, l'algorithme réalise au plus $2nm$ réductions saturantes. ■

Lemme 3.9. *La procédure préflot réalise $O(n^2m)$ réductions non saturantes.*

Preuve. Nous notons T l'ensemble des sommets actifs et t un élément générique de T . Nous considérons comme fonction potentiel la somme $\phi = \sum_{t \in T} \Delta(t)$ des distances estimées *des sommets actifs*. Remarquons d'abord qu'à l'initialisation de l'algorithme, on a $\phi < n^2$ et qu'à sa terminaison ϕ est nul. Lors d'une itération, le sommet actif x est choisi et trois cas peuvent se présenter.

- a) *Il n'existe pas d'arc admissible d'origine x .* L'ensemble T ne change pas mais la distance estimée de x augmente strictement. L'accroissement de ϕ est alors égal à celui de $\Delta(x)$. Il résulte alors du lemme 3.7 que la somme δ_1 des accroissements de ϕ dus aux augmentations de distance est majorée par $2n^2$.
- b) *Une réduction saturante est réalisée sur v .* Notons y l'extrémité de l'arc v . La fonction distance estimée ne change pas mais le sommet y peut devenir actif. L'accroissement de ϕ est alors majoré par $2n$ (lemme 3.7). Comme l'algorithme exécute au plus $2nm$ réductions saturantes, l'accroissement δ_2 de ϕ dû aux réductions saturantes est majoré par $2n^2m$.
- c) *Une réduction non saturante est réalisée sur v .* Notons y l'extrémité de l'arc v . La fonction distance estimée ne change pas, le sommet y peut devenir actif mais le sommet x n'est plus actif. L'accroissement de ϕ est majoré par $\Delta(y) - \Delta(x) = -1$. Donc si l'algorithme réalise N réductions non saturantes, l'accroissement correspondant δ_3 de ϕ est plus petit que $-N$.

En résumé, si nous notons δ l'accroissement total de ϕ , nous avons $\delta = \delta_1 + \delta_2 + \delta_3$ et $\delta \geq -n^2$. En utilisant les majorations précédentes il vient :

$$N \leq \delta_3 = -\delta + (\delta_1 + \delta_2) \leq 3n^2 + 2n^2m.$$

■

Les lemmes 3.7, 3.8 et 3.9 conduisent à l'évaluation de la complexité de l'algorithme du préflot.

Théorème 3.10. *La procédure préflot calcule un flot de valeur maximum en temps $O(n^2m)$.*

Preuve. La complexité de la procédure AUGMENTER-DISTANCE est $O(k)$ si k est le nombre de successeurs de x dans G . Comme pour un même sommet la distance estimée est augmentée au plus $2n$ fois, le temps opératoire des augmentations de distance est aussi en $O(nm)$. Si l'on associe à chaque sommet actif t un indicateur $i(t)$ qui est un arc admissible sortant de t s'il en existe ou un symbole spécial sinon, il suffit d'utiliser une structure de données (par exemple une liste doublement chaînée avec pointeurs inverses), on réalise en un temps $O(1)$ les opérations de mise à jour de l'ensemble des couples $(t, i(t))$ lors d'une réduction d'excès. Le temps opératoire global des réductions d'excès est alors en $O(n^2m)$. Il en résulte une complexité $O(n^2m)$ pour la procédure préflot. ■

L'algorithme du préflot peut être considéré comme un algorithme générique car il laisse libre le choix à chaque itération du nouveau sommet actif et de l'arc admissible. Nous allons présenter deux variantes efficaces de cet algorithme. L'algorithme de Karzanov choisit le sommet actif le plus éloigné du puits, l'algorithme des «excès échelonnés» réduit en priorité les sommets actifs dont les excès sont les plus grands.

8.3.4 L'algorithme de Karzanov

Cet algorithme sélectionne à chaque itération un sommet actif dont la distance estimée au puits est maximum. Il en résulte qu'entre deux exécutions consécutives de la procédure AUGMENTER-DISTANCE il y a au plus n réductions non saturantes. En effet, une réduction non saturante annule l'excès d'un sommet actif x et l'excès de x restera nul puisque, tant que la fonction distance estimée n'est pas mise à jour, les sommets actifs choisis après x ont une distance estimée inférieure ou égale à celle de x . Le nombre total de réductions non saturantes est donc en $O(n^3)$. Si l'on maintient pour chaque valeur possible r de la distance estimée une liste des sommets actifs dont la distance estimée au puits est r et un pointeur vers la liste non vide de plus grand r , la sélection d'un sommet actif de plus grande distance estimée est en $O(1)$. La complexité globale de l'algorithme de Karzanov est alors $O(n^3)$.

8.3.5 L'algorithme des excès échelonnés

L'idée de réaliser un *échelonnement* des excès et de réduire en priorité les plus grands excès conduit à une variante efficace de l'algorithme du préflot.

Cette variante découpe l'exécution de l'algorithme du préflot en phases. Si au début de la phase k on dispose d'un majorant E_{k-1} de l'ensemble des excès, cette phase sélectionnera tant qu'il en existe un sommet actif dont l'excès est supérieur ou égal à $E_{k-1}/2$. Lorsqu'il n'existe plus de tels sommets, la phase k se termine et l'on pose $E_k = E_{k-1}/2$ pour la phase $k+1$. Une phase consiste donc à réduire en priorité les plus gros excès. Il est cependant nécessaire d'affiner ce principe général qui pourrait trop souvent faire converger sur un même sommet une quantité de flux trop grande pour pouvoir être équilibrée. Une telle situation pour un sommet x force en effet l'algorithme à augmenter suffisamment la distance estimée du sommet x pour qu'il apparaisse dans le graphe d'écart un arc v *admissible* permettant de réduire l'excès du sommet x . La nouvelle règle de sélection d'un sommet actif est la suivante :

Choisir un sommet actif x tel que
 $e_f(x) \geq E_{k-1}/2$ et $\Delta(x)$ est minimum.

La nouvelle règle de réduction de l'excès décrite par la procédure RÉDUIRE-EXCÈS-MODIFIÉ ci-dessous assure qu'au cours de la phase k les excès de tous les sommets actifs seront majorés par E_{k-1} .

```

procédure RÉDUIRE-EXCÈS-MODIFIÉ( $f, v$ );
  si  $v = u'$ 
    alors
       $\epsilon := \min\{e_f(v^-), b(u) - f(u), E_{k-1} - e_f(v^+)\}$ ;
       $f(u) := f(u) + \epsilon$ 
    sinon
       $\epsilon := \min\{e_f(v^-), f(u), E_{k-1} - e_f(v^+)\}$ ;
       $f(u) := f(u) - \epsilon$ 
  finsi.

```

L'algorithme est décrit par la procédure EXCÈS-ÉCHELONNÉS ci-dessous où l'on note B la capacité maximale d'un arc de G et $T(f, E)$ l'ensemble des sommets actifs pour le préflot f et dont l'excès est supérieur ou égal à E .

```

procédure EXCÈS-ÉCHELONNÉS(  $G, b, s, p$  );
   $E := 2^{\lceil \log B \rceil}$ ;  $k := 1 + \lceil \log B \rceil$ ;
  INITIALISER-PRÉFLOT;
  pour  $i$  de 1 à  $k$  faire
    tantque  $T(f, E) \neq \emptyset$  faire
      choisir un sommet  $x$  dans  $T(f, E)$  de distance estimée minimum;
      si  $x$  est une sortie du graphe d'admissibilité
        alors AUGMENTER-DISTANCE(  $x$  )
        sinon
          choisir un arc admissible  $v$  d'origine  $x$ ;
          RÉDUIRE-EXCÈS-MODIFIÉ(  $f, v$  )
      finsi
    fintantque
     $E := E/2$ 
  finpour.

```

Les nouvelles règles de réduction et de sélection utilisées par l'algorithme induisent la propriété suivante :

Lemme 3.11. *Pendant la phase k , une réduction non saturante porte sur au moins $E_{k-1}/2$ unités de flux et l'excès de tout sommet reste inférieur à E_{k-1} .*

Preuve. Soit x un sommet actif choisi lors d'une réduction non saturante de la phase k et soit v l'arc admissible d'origine x et d'extrémité y sélectionné pour cette réduction. On a $\Delta(y) = \Delta(x) - 1$ puisque v est admissible et par conséquent $e_f(y) < E_{k-1}/2$ d'après la nouvelle règle de sélection. La réduction n'étant pas saturante, la valuation de v est supérieure ou égale à $\min\{e_f(x), E_{k-1} - e_f(y)\}$. Comme $e_f(x) \geq E_{k-1}/2$ et $E_{k-1} - e_f(y) > E_{k-1}/2$, la variation de flux du père de v est supérieure ou égale à $E_{k-1}/2$.

Considérons maintenant une réduction quelconque de l'excès du sommet actif x à partir de l'arc admissible v d'origine x et d'extrémité y . Après cette réduction, seul le sommet y a pu voir son excès augmenter et cette augmentation est inférieure ou égale à $E_{k-1} - e_f(y)$. L'excès du sommet y inférieur à E_{k-1} au début de la phase k restera donc inférieur à cette valeur pendant toute cette phase. ■

Théorème 3.12. *La complexité en temps de l'algorithme EXCÈS-ÉCHELONNÉS est $O(nm + n^2 \log B)$.*

Preuve. Evaluons le nombre de réductions non saturantes au cours de la phase k en considérant la fonction potentiel :

$$F = \frac{1}{E_{k-1}} \sum_{x \in S'} e_f(x) \Delta(x).$$

Au début de cette phase, F est majorée par $2n^2$ puisque pour tout sommet x , $e_f(x) < E_{k-1}$ et $\Delta(x) \leq 2n$ (Lemme 3.7). Lors de la sélection d'un sommet actif x , examinons les variations de F dues aux augmentations de distance et aux réductions saturantes

- a) *Augmentation de distance.* L'algorithme augmente la distance estimée $\Delta(x)$ de d unités ($d \geq 1$). Il en résulte une augmentation de F majorée par d . La somme des augmentations de la distance estimée d'un sommet étant majorée par $2n$, l'augmentation totale de F due aux occurrences de ce premier cas est majorée par $2n^2$.
- b) *Réduction saturante.* La fonction F décroît au moins de la valeur $1/2$ puisque, d'après le lemme 3.11, l'excès de x diminue d'au moins $E_{k-1}/2$ unités et l'excès de y augmente de la même quantité.

Comme F reste positif par définition et qu'une réduction quelconque fait décroître F , le nombre de réductions non saturantes au cours de la phase k est majoré par $8n^2$.

Dans l'étude de l'algorithme du préflot, nous avons montré que la complexité des opérations autres que les réductions non saturantes est $O(nm)$. On peut d'autre part gérer l'ensemble des sommets actifs dont l'excès est supérieur ou égal à $E_{k-1}/2$ en associant à chaque valeur possible r de la distance estimée une liste doublement chaînée des sommets dont la distance estimée est r et l'excès supérieur à $E_{k-1}/2$. Un index sur la liste non vide de plus petit r est également maintenu. Les opérations d'ajout, de retrait et de sélection d'un sommet actif sont alors en $O(1)$ et la reconstitution des listes à chaque nouvelle phase en $O(n)$. Le nombre de phases étant égal à $1 + \lceil \log B \rceil$, la complexité globale de l'algorithme EXCÈS-ÉCHELONNÉS est $O(nm + n^2 \log B)$. ■

8.4 Flot de coût minimum

Nous considérons dans cette section le problème de la recherche d'un flot compatible de coût minimum dans un réseau valué $R = (G, a, b, c)$. Nous présentons d'abord les propriétés fondamentales de dualité sur lesquelles s'appuient tous les algorithmes de résolution du problème. Nous décrivons ensuite l'algorithme primal de Golberg et Tarjan pour la recherche d'un flot de coût minimum dans un réseau $R = (G, 0, b, c)$. Nous montrons ensuite que l'on peut ramener le problème général à la recherche d'un flot maximum de coût minimum dans un réseau muni d'une entrée, d'une sortie et dont les arcs sont valués par une capacité maximale et un coût. Nous décrivons un algorithme de type dual pour la résolution de ce problème lorsque les coûts sont positifs ou nuls. Nous considérons enfin le problème de la recherche d'un plan de transport de coût minimum et sa résolution par l'algorithme dual d'Edmonds et Karp.

8.4.1 Graphe d'écart et conditions d'optimalité

Soit f un flot d'un réseau valué $R = (G, 0, b, c)$. Le graphe d'écart du flot f est obtenu à partir du graphe d'écart G_f défini par la procédure GRAPHE-D'ÉCART de la section 8.3 en munissant simplement chaque arc v d'un coût, noté également $c(v)$, et défini par :

$$c(v) = \begin{cases} c(u) & \text{si } v = u' \\ -c(u) & \text{si } v = u'' \end{cases}$$

Soit $\mu = ((s_0, u_1, s_1), (s_1, u_2, s_2), \dots, (s_{q-1}, u_q, s_q))$ un cycle améliorant pour le flot f . Par définition, un arc avant de μ n'est pas saturé et un arc arrière de μ n'est pas vide. Il en résulte que la suite $\nu = ((s_0, v_1, s_1), (s_1, v_2, s_2), \dots, (s_{q-1}, v_q, s_q))$ où

$$v_k = \begin{cases} u'_k & \text{si } u_k \text{ est arc avant de } \mu \\ u''_k & \text{si } u_k \text{ est arc arrière de } \mu \end{cases}$$

est un circuit de coût négatif du graphe d'écart G_f . Réciproquement à un circuit de coût négatif du graphe d'écart G_f correspond un cycle améliorant de G .

La proposition 4.1, conséquence directe du théorème 2.7, énonce une condition nécessaire et suffisante d'optimalité qui porte uniquement sur le graphe d'écart G_f .

Proposition 4.1. *Un flot f d'un réseau $R = (G, 0, b, c)$ est de coût minimum si et seulement si son graphe d'écart G_f ne possède aucun circuit de coût strictement négatif.*

8.4.2 Problème dual et conditions d'optimalité

Nous avons établi que la recherche d'un flot de coût minimum correspond à la résolution d'un programme linéaire dont la matrice est la matrice d'incidence sommets-arcs du graphe G . Les algorithmes de résolution efficaces de ce problème vont d'une part exploiter la structure de cette matrice (c'est-à-dire travailler sur le graphe d'écart) et d'autre part tirer parti des propriétés de dualité issues de la programmation linéaire.

Le *programme linéaire primal* (PLP) associé à un problème de flot de coût minimum dans un réseau $R = (G, 0, b, c)$ s'écrit :

$$\begin{aligned} \forall u \in A, \quad & f(u) \geq 0 \\ \forall u \in A, \quad & f(u) \leq b(u) \\ \forall x \in S, \quad & e_f(x) = 0 \end{aligned} \tag{PLP}$$

$$\text{MIN} \quad \sum_{u \in A} c(u) f(u)$$

où les variables $f(u)$ sont rationnelles.

Le *programme linéaire dual* (PLD) associe à chaque sommet x du réseau la variable sans contrainte de signe $\pi(x)$ appelée *potentiel* du sommet x , et associe

à chaque arc u du réseau une variable positive ou nulle notée $\delta(u)$. Le programme dual (*PLD*) s'écrit :

$$\begin{aligned} \forall u \in A, \quad & -\delta(u) + \pi(u^+) - \pi(u^-) \leq c(u) \\ & \forall u \in A, \quad \delta(u) \geq 0 \\ \text{MAX} \quad & - \sum_{u \in A} b(u)\delta(u) \end{aligned} \quad (\text{PLD})$$

L'existence d'un sous-ensemble dominant des solutions du programme dual montre que les véritables inconnues du programme dual sont les potentiels des sommets de G . En effet si nous *fixons* le potentiel de chaque sommet, il est immédiat de vérifier que la meilleure solution du programme dual correspond aux valeurs des variables $\delta(u)$ calculées par la formule (Δ) ci-dessous :

$$\forall u \in A, \quad \delta(u) = \max\{0, -c(u) + \pi(u^+) - \pi(u^-)\}. \quad (\Delta)$$

Nous ne considérerons donc dans la suite que des *solutions dominantes du programme dual*, c'est-à-dire des couples (π, δ) pour lesquels les valeurs des $\delta(u)$ résultent de π par la formule (Δ). Une solution du dual sera donc complètement caractérisée par un ensemble de potentiels π .

Etant donné un flot f solution du programme linéaire primal et une solution (π, δ) du programme linéaire dual, le théorème *des écarts complémentaires*, corollaire du théorème de la dualité, fournit une condition *nécessaire et suffisante* d'optimalité des deux solutions. Cette condition, notée (*OPT*), s'écrit :

$$\begin{aligned} \forall u \in A, \quad & \delta(u)[f(u) - b(u)] = 0 \\ \forall u \in A, \quad & f(u)[c(u) + \delta(u) - \pi(u^+) + \pi(u^-)] = 0 \end{aligned} \quad (\text{OPT})$$

Les conditions d'optimalité précédentes peuvent être exprimées de manière plus condensée à partir du graphe d'écart en utilisant la notion de coût réduit. Soit π une fonction potentiel définie sur les sommets de G et f un flot, le *coût réduit* relatif à π d'un arc v du graphe d'écart G_f est défini par $\bar{c}_\pi(v) = c(v) + \pi(v^-) - \pi(v^+)$. La Proposition 4.2 fournit une condition nécessaire et suffisante d'optimalité fondée sur les coûts réduits du graphe d'écart G_f .

Proposition 4.2. *Un flot f et un ensemble de potentiels π sont optimaux si et seulement si le coût réduit relatif à π de tout arc du graphe d'écart G_f est positif ou nul.*

Preuve. Soient f un flot et (π, δ) une solution dominante du dual qui satisfait la condition (*OPT*). Considérons un arc v du graphe d'écart G_f .

Si $v = u'$ (c'est-à-dire si v est le représentant conforme de l'arc u de G), l'arc u n'est pas saturé et l'on a d'après (*OPT*) : $\delta(u) = 0$. Il résulte alors de la formule (Δ) que $\pi(u^+) - \pi(u^-) - c(u) \leq 0$ et donc que $\bar{c}_\pi(v) = c(v) + \pi(v^-) - \pi(v^+) \geq 0$.

Si $v = u''$ (c'est-à-dire si v est le représentant non conforme de l'arc u de G), l'arc u n'est pas vide et l'on a d'après (*OPT*) : $0 \leq \delta(u) = -c(u) + \pi(u^+) - \pi(u^-)$. Comme $v^+ = u^-$ et $v^- = u^+$, nous avons $\bar{c}_\pi(v) = c(v) + \pi(v^-) - \pi(v^+) \geq 0$.

Supposons maintenant que le coût réduit de tout arc du graphe d'écart soit positif ou nul. Pour un arc u de G , trois cas sont possibles :

L'arc u est libre. Les deux arcs u' et u'' de G_f ont des coûts réduits positifs ou nuls, et donc nuls puisque $\bar{c}_\pi(u') = -\bar{c}_\pi(u'')$; il en résulte que $\delta(u) = 0$ et que la condition (*OPT*) est satisfaite pour l'arc u .

L'arc u est saturé. Le coût réduit de l'arc u'' étant positif ou nul, on a $-c(u) + \pi(u^+) - \pi(u^-) \geq 0$. On a donc d'après (Δ) : $\delta(u) = -c(u) + \pi(u^+) - \pi(u^-)$ et la condition (*OPT*) est satisfaite pour l'arc u .

L'arc u est vide. Le coût réduit de l'arc u' étant positif ou nul, on a $c(u) + \pi(u^-) - \pi(u^+) \geq 0$. On a donc d'après (Δ) : $\delta(u) = 0$. La condition (*OPT*) est satisfaite pour l'arc u . ■

8.4.3 Un algorithme primal

Le problème considéré dans cette section est la recherche d'un flot compatible de coût minimum dans un réseau valué $R = (G, 0, b, c)$. Aucune restriction n'est faite ici sur le signe des coûts $c(u)$. L'algorithme que nous allons décrire, dû à Goldberg et Tarjan détruit systématiquement à chaque itération le cycle améliorant de G associé à un circuit de coût moyen minimum du graphe d'écart. Cet algorithme est de type primal puisqu'à chaque étape, il fournit un flot compatible tout en améliorant la satisfaction des conditions d'optimalité duales. La procédure **GOLDBERG-TARJAN**(R) ci-dessous réalise cet algorithme.

```

procédure GOLDBERG-TARJAN( $R$ );
   $f := 0$ ;
   $\epsilon := \max_{u \in A} |c(u)|$ ;
  tantqu'il existe un circuit négatif dans  $G_f$  faire
     $\rho := \text{CIRCUIT-COÛT-MOYEN-MINIMUM}(G_f)$ ;
     $f := \text{AUGMENTER-FLOT-SUR-CIRCUIT}(f, \rho)$ 
  fintantque.

```

Soit f un flot et G_f le graphe d'écart associé. Le *coût moyen* d'un circuit ρ de q arcs dans G_f est la quantité $\sum_{v \in \rho} c(v)/q$. Remarquons que pour toute fonction potentiel π , le coût réduit moyen d'un circuit de G_f est égal à son coût moyen. Soit $\gamma(f)$ le coût moyen minimum d'un circuit de G_f . Remarquons qu'il existe un circuit élémentaire de coût $\gamma(f)$. L'algorithme de Karp, de complexité $O(nm)$, permet de calculer le coût moyen minimum d'un circuit dans un graphe valué de n sommets et m arcs (voir exercices). Le flot f est dit ϵ -*optimal* s'il existe une fonction potentiel π telle que :

$$\forall v \in A_f, \quad \bar{c}_\pi(v) \geq -\epsilon.$$

Il résulte de cette définition qu'un flot 0-optimal est de coût minimum. La proposition suivante montre que si ϵ est suffisamment petit, un flot ϵ -optimal est de coût minimum.

Proposition 4.3. *Si $0 \leq \epsilon < 1/n$, un flot ϵ -optimal f est de coût minimum.*

Preuve. Soit ρ un circuit élémentaire de G_f . On a $c(\rho) = \bar{c}_\pi(\rho) > -1$. La valeur de ρ est donc positive ou nulle. ■

Etant donné un flot f , on appelle *déviaton* de f la valeur minimum de ϵ pour laquelle le flot f est ϵ -optimal. La déviaton de f , notée $\epsilon(f)$, est liée au coût moyen minimum d'un circuit de G_f .

Théorème 4.4. *Soit f un flot. La déviaton de f est égale à $\max\{0, -\gamma(f)\}$.*

Preuve. Si le flot f est ϵ -optimal, alors pour un circuit quelconque ρ de q arcs dans G_f , on a $c(\rho) \geq -q\epsilon$. Il en résulte que $\epsilon \geq -\gamma(f)$. Nous montrons maintenant qu'il existe effectivement une fonction potentiel π pour laquelle le flot f est $-\gamma(f)$ -optimal.

Supposons d'abord que le graphe G_f soit fortement connexe. Définissons pour chaque arc v de G_f la valuation $c'(v) = c(v) - \gamma(f)$ et choisissons un sommet source s . Pour tout circuit ρ de q arcs dans G_f , on a $c'(\rho) = c(\rho) - q\gamma(f) \geq 0$. Pour tout sommet x , il existe donc un chemin de coût minimum de s à x dans G_f pour la valuation c' . Si l'on définit le potentiel $\pi(x)$ du sommet x par la valeur de ce chemin, il vient :

$$\forall v \in A_f, \quad \pi(v^+) \leq \pi(v^-) + c(v) - \gamma(f).$$

Il en résulte que le flot f est $-\gamma(f)$ -optimal.

Si le graphe G_f n'est pas fortement connexe, on note $\{C_1, \dots, C_p\}$ ses composantes fortement connexes. Tous les circuits du graphe induit par C_i ont une valeur positive ou nulle pour la valuation c' . En considérant le graphe induit par C_i *seul*, nous pouvons associer à chaque sommet x de C_i une fonction $\pi'(x)$ qui vérifie l'ingalité précédente sur tous les arcs du graphe (fortement connexe) induit par C_i . La valuation a_{ij} d'un arc (C_i, C_j) du graphe réduit de G_f est définie par :

$$a_{ij} = \min\{c'(v) + \pi'(v^-) - \pi'(v^+) \mid v \in A_f, v^- \in C_i, v^+ \in C_j\}.$$

Le graphe réduit de G_f étant sans circuits, il existe pour tout sommet C_i un chemin de coût minimum α_i d'extrémité C_i . Par définition des valeurs α_i , nous avons $\alpha_j - \alpha_i \leq a_{ij}$ pour tout arc (C_i, C_j) . Le potentiel d'un sommet x de C_i est alors défini par $\pi(x) = \pi'(x) + \alpha_i$. Pour tout arc v de G_f appartenant à un graphe induit par une composante fortement connexe, on a bien sûr $\bar{c}_\pi(v) \geq -\gamma(f)$. Soit maintenant un arc v tel que $v^- \in C_i$ et $v^+ \in C_j$, on a :

$$\pi(v^+) - \pi(v^-) = \pi'(v^+) - \pi'(v^-) + \alpha_j - \alpha_i \leq \pi'(v^+) - \pi'(v^-) + a_{ij} \leq c'(v).$$

Le flot f est donc $-\gamma(f)$ -optimal pour le potentiel π . ■

Il est utile dans la suite de considérer le graphe partiel de G_f constitué des arcs dont le coût réduit pour un potentiel π est strictement négatif. Nous appellerons ce graphe le *graphe d'inadmissibilité* de f pour la fonction potentiel π et nous le noterons $G_f^-(\pi)$. La proposition suivante montre que si un flot f est ϵ -optimal pour la fonction potentiel π et si $G_f^-(\pi)$ est sans circuits, la déviation de f est au plus $(1 - 1/n)\epsilon$.

Proposition 4.5. *Soit f un flot ϵ -optimal pour le potentiel π . Si le graphe d'inadmissibilité de f est sans circuit, alors le flot f est $(1 - 1/n)\epsilon$ -optimal.*

Preuve. Soit ρ un circuit élémentaire de q arcs dans G_f . Le circuit ρ contient au moins un arc de coût réduit positif. Le coût moyen de ρ qui est au moins égal à $-(1 - 1/q)\epsilon$ est minoré par $-(1 - 1/n)\epsilon$. Il en résulte que pour un circuit de coût moyen minimum, on a $\gamma(f) \geq -(1 - 1/n)\epsilon$. On déduit alors le résultat du théorème 4.4. ■

Convergence

La convergence de l'algorithme est assurée car le flot g résultant de la destruction d'un circuit ρ de coût moyen minimum dans G_f par la procédure AUGMENTER-FLOT-SUR-CIRCUIT(f, ρ) possède une déviation au plus égale à celle du flot f . Les deux lemmes suivants précisent les conditions de cette convergence.

Lemme 4.6. *Soit g le flot résultant d'un flot ϵ -optimal f après la destruction d'un circuit de coût moyen minimum dans G_f . Les déviations de f et g satisfont $\epsilon(g) \leq \epsilon(f)$.*

Preuve. D'après la définition de la déviation $\epsilon(f)$ du flot f , il existe un potentiel π tel que pour tout arc v de G_f on a $\bar{c}_\pi(v) \geq -\epsilon(f)$. Soit ρ le circuit détruit. Comme le coût moyen des arcs de ρ est égal à $-\epsilon(f)$, le coût réduit pour π d'un arc quelconque de ρ est égal à $-\epsilon(f)$. Soit v un arc de G_g qui n'existe pas dans G_f . Si $v = u'$, alors l'arc u'' est un arc de ρ . On a donc $\bar{c}_\pi(u'') = -\epsilon(f)$ et $\bar{c}_\pi(v) = \epsilon(f)$. On aboutit à la même conclusion si $v = u''$. Il en résulte que $\epsilon(g) \leq \epsilon(f)$. ■

Nous noterons dans la suite m_e le nombre d'arcs du graphe d'écart G_f qui reste toujours inférieur ou égal à $2m$. Le lemme suivant évalue le gain obtenu lorsque l'algorithme exécute m_e itérations successives.

Lemme 4.7. *Soit f un flot et soit g le flot obtenu après m_e destructions de circuits. Si g n'est pas de coût minimum, sa déviation satisfait $\epsilon(g) \leq (1 - 1/n)\epsilon(f)$.*

Preuve. Soit f un flot ϵ -optimal pour le potentiel π . La destruction d'un circuit ρ de coût moyen minimum dont tous les arcs ont un coût réduit négatif ne peut créer que des arcs de coût réduit positif ou nul et supprime au moins un arc de $G_f^-(\pi)$. Nous considérons alors deux cas pour le flot g obtenu après les m_e itérations successives :

Lors de chaque destruction, les arcs du circuit détruit ont un coût réduit négatif ou nul. D'après la remarque précédente, le graphe d'inadmissibilité $G_f^-(\pi)$ est vide et le flot g est donc de coût minimum.

Soit ρ le premier circuit supprimé contenant un arc de coût réduit positif. D'après le lemme 4.6, la déviation du flot h obtenu avant la suppression de ρ est inférieure ou égale à $\epsilon(f)$. D'après la preuve de la proposition 4.5, le coût moyen de ρ vaut au moins $-(1 - 1/n)\epsilon$. Il résulte alors du théorème 4.4 et du lemme 4.6 que la déviation des flots obtenus après h est inférieure ou égale à $(1 - 1/n)\epsilon$. ■

Les lemmes 4.6 et 4.7 sont alors suffisants pour majorer polynomialement le nombre d'itérations de l'algorithme.

Théorème 4.8. *L'algorithme de Goldberg et Tarjan réalise $O(nm \log nC)$ itérations.*

Preuve. Soit f le flot courant. Lors de l'initialisation de l'algorithme, on a $\epsilon(f) \leq C$. D'après la proposition 4.3, l'algorithme se termine dès que la déviation du flot courant est strictement inférieure à $1/n$. Soit K le nombre total d'itérations et posons $K - 1 = pq + r$, ($0 \leq r < p$). La déviation obtenue après pq destructions est inférieure ou égale à $(1 - 1/n)^{pq}C$ d'après le lemme 4.7 et supérieure ou égale à $1/n$ puisque l'algorithme n'est pas terminé. On a donc $(1 - 1/n)^{pq}C \geq 1/n$ ou encore $(1 - 1/n)^{\lfloor (K-1)/p \rfloor}C \geq 1/n$. Comme pour $n > 1$ on a $\ln(1 - 1/n) < -1/n$, il vient :

$$\lfloor (K - 1)/p \rfloor \leq -\ln nC / \ln(1 - 1/n) \leq n \ln nC.$$

Il en résulte que $K = O(mn \log nC)$. ■

Si l'on utilise l'algorithme de Karp pour déterminer à chaque itération un circuit de coût moyen minimum, la complexité de l'algorithme de Goldberg et Tarjan est $O(n^2 m^2 \log nC)$. En fait Goldberg et Tarjan ont montré que le nombre total d'itérations est majoré par un polynôme en n et m . La preuve due à Tardos repose sur une propriété supplémentaire des flots ϵ -optimaux que nous énonçons sans démonstration.

Proposition 4.9. *Soit f un flot ϵ -optimal pour le potentiel π et v un arc de G_f de coût réduit supérieur ou égal à $2n\epsilon$. Le flux du père de v dans G est le même pour tous les flots ϵ -optimaux.*

Cette proposition permet d'établir que la complexité en temps de l'algorithme de Goldberg et Tarjan est $O(n^2 m^2 \min\{\log nC, m \log n\})$.

Une amélioration de l'algorithme

La version de l'algorithme de Goldberg et Tarjan décrite dans la section précédente est intéressante à un double point de vue, d'une part elle exploite directement la condition nécessaire et suffisante d'optimalité d'un flot compatible et d'autre part il s'agit d'un algorithme *fortement polynomial*, c'est-à-dire polynomial lorsque le coût d'une opération arithmétique élémentaire est polynomial en le nombre de bits nécessaires pour coder ses opérandes. Cependant cet algorithme est moins efficace que d'autres algorithmes fortement polynomiaux fondés sur des techniques d'échelonnement ou d'approximations.

Goldberg et Tarjan ont alors proposé une nouvelle version qui constitue l'un des meilleurs algorithmes connus aujourd'hui pour des réseaux pas trop denses dont les coûts (entiers) ne sont pas trop grands. Cette amélioration consiste à remplacer la destruction d'un circuit de coût moyen minimum (qui coûte $O(nm)$ opérations élémentaires) par une série d'au plus m destructions de circuits du graphe d'inadmissibilité suivie d'un ajustement de la fonction potentiel. Une structure de données ad-hoc, appelée *arbres dynamiques*, permet de réaliser une destruction de circuit ou un ajustement de potentiel en $O(\log n)$.

Soit f le flot courant et soit π le potentiel courant. Le plus petit ϵ , noté $\epsilon(f, \pi)$, pour lequel le flot f est ϵ -optimal à π fixé, est défini par :

$$\epsilon(f, \pi) = \max\{0, -\min\{\bar{c}_\pi(v) \mid v \in A_f\}\}.$$

Rappelons que le graphe d'inadmissibilité $G_f^-(\pi)$ est formé des arcs v de G_f dont le coût réduit $\bar{c}_\pi(v)$ par rapport à π est strictement négatif. L'algorithme peut alors être décrit comme suit :

```

procédure GOLDBERG-TARJAN-PLUS( $R$ );
   $f := 0$ ;  $\pi := 0$ ;
  tantque  $\epsilon(f, \pi) > 1/n$  faire
    tantque  $G_f^-(\pi)$  possède un circuit faire
      DÉTRUIRE-CIRCUIT( $G_f^-(\pi)$ )
    fintantque
      AJUSTER-POTENTIEL( $\pi$ )
  fintantque.

```

La procédure DÉTRUIRE-CIRCUIT détermine un circuit ρ de $G_f^-(\pi)$, et détruit ce circuit par un appel à la procédure AUGMENTER-FLOT-SUR-CIRCUIT. La procédure AJUSTER-POTENTIEL transforme π en un potentiel π' tel que $\epsilon(f, \pi') \leq \epsilon(f, \pi)$.

Si l'on définit une itération comme une série de destructions de circuits suivie d'un ajustement de la fonction potentiel (boucle tantque extérieure), le théorème suivant montre que cet algorithme converge après au plus $O(n \log nC)$ itérations.

Théorème 4.10. *La procédure GOLDBERG-TARJAN-PLUS calcule un flot de coût minimum en $O(n \log nC)$ itérations.*

Preuve. Considérons une itération. Chaque destruction de circuit dans $G_f^-(\pi)$ n'ajoute pas d'arcs à $G_f^-(\pi)$ mais en détruit au moins un. Le graphe $G_f^-(\pi)$ sera donc sans circuits après au plus m_e destructions de circuits. Si, après ces destructions, le graphe $G_f^-(\pi)$ est vide, le flot f est de coût minimum, sinon le flot f est $(1 - 1/n)\epsilon(f, \pi)$ -optimal d'après le lemme 4.7. La convergence et le nombre maximum d'itérations résultent alors du lemme 4.3 et du théorème 4.4. ■

Goldberg et Tarjan ont créé une structure de données spécifique, appelée *arbre dynamique* qui permet de réaliser chaque destruction de circuit et ajustement de potentiel en $O(\log n)$. Indiquons simplement qu'elle permet de gérer une famille d'arborescences dont chaque arc correspond à un arc v du graphe $G_f^-(\pi)$ valué par $b(u) - f(u)$ si $v = u'$ ou par $f(u)$ si $v = u''$. Cette structure conduit à un algorithme de complexité $O(nm(\log n) \min\{\log nC, m \log n\})$ qui est l'un des meilleurs connus aujourd'hui.

8.4.4 Flot maximum de coût minimum

Soit $R = (G, a, b, c)$ un réseau valué et f un flot compatible de ce réseau. Nous montrons que l'on peut ramener la recherche d'un flot compatible de coût minimum du réseau R à celle d'un flot maximum de coût minimum dans un réseau R' muni d'une entrée, d'une sortie, et dont les arcs sont valués par une capacité maximale et un coût. La construction du réseau R' repose sur le signe des excès $e_a(x)$ des sommets de G pour la fonction a . Nous noterons S_+ (respectivement S_-) l'ensemble des sommets de G dont l'excès pour a est strictement positif (respectivement négatif) et α la quantité $\sum_{x \in S_+} e_a(x) = -\sum_{x \in S_-} e_a(x)$. Nous supposons α strictement positif.

Les sommets du réseau R' sont ceux du graphe G auxquels on adjoint un sommet source s et un sommet puits p . L'ensemble des arcs du réseau R' est formé :

- des arcs u du graphe G où chaque arc u est muni d'une capacité minimale nulle, d'une capacité maximale égale à $b(u) - a(u)$ et du coût $c(u)$;
- pour chaque sommet x de S_+ d'un nouvel arc u_{sx} , d'origine s , d'extrémité x , de capacité minimale nulle, de capacité maximale $e_a(x)$ et de coût nul;
- pour chaque sommet x de S_- d'un nouvel arc u_{xp} , d'origine x , d'extrémité p , de capacité minimale nulle, de capacité maximale $-e_a(x)$ et de coût nul;
- d'un arc de retour u_0 d'origine p , d'extrémité s , de capacité minimale nulle, de capacité maximale α et de coût nul.

La figure 4.1 montre en (A) un réseau valué R où sur chaque arc sont inscrits le triplet $(a(u), b(u), c(u))$ et le flux $f'(u)$ en gras et où à côté de chaque sommet l'excès pour la fonction a est souligné. Sur la partie (B) est représenté le réseau R' où sur chaque arc sont inscrits le couple $(b(u), c(u))$ et en gras le flux $f(u)$.

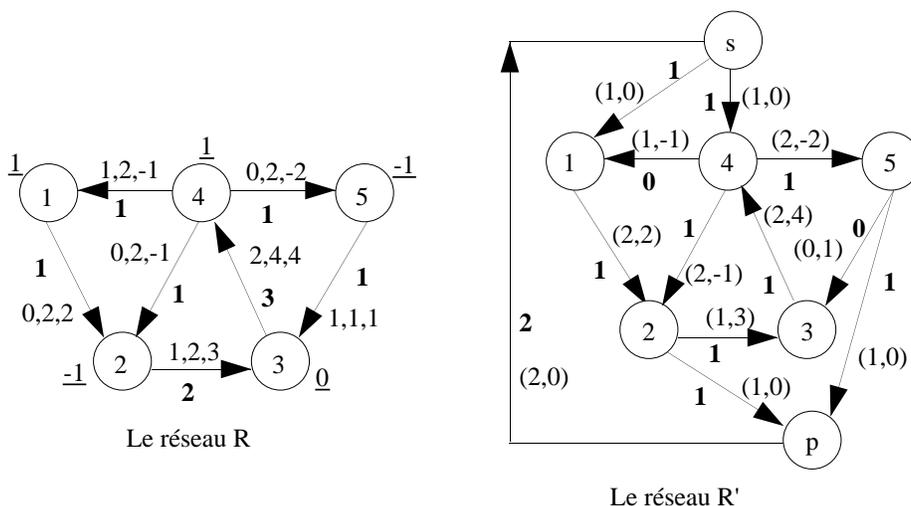


Figure 4.1: Les réseaux R et R' .

On appelle *flot saturant* du réseau R' un flot compatible de valeur α . La proposition 4.11 établit la correspondance entre les flots compatibles de R et les flots saturants de R' .

Proposition 4.11. *Il existe une bijection entre les flots compatibles du réseau R et les flots saturants du réseau R' . L'image d'un flot compatible de coût minimum du réseau R est un flot maximum de coût minimum du réseau R' .*

Preuve. Soit f un flot compatible de R , nous lui associons l'application f' définie sur les arcs du réseau R' par :

$$f'(u) = \begin{cases} e_a(x) & \text{si } u = u_{sx} \\ -e_a(x) & \text{si } u = u_{xp} \\ f(u) - a(u) & \text{si } u \in A \\ \alpha & \text{si } u = u_0 \end{cases}$$

La fonction f' satisfait les contraintes de capacité du réseau R' , montrons que cette fonction est aussi un flot de R' en calculant les excès des différents sommets. Il est d'abord clair que $e_{f'}(s) = e_{f'}(p) = 0$. Considérons un sommet x de S_+ , nous avons :

$$e_{f'}(x) = (e_f(x) - e_a(x)) + f'(u_{sx}) = 0;$$

de même pour un sommet x de S_- , nous avons :

$$e_{f'}(x) = (e_f(x) - e_a(x)) - f'(u_{xp}) = 0.$$

La fonction f' est donc un flot saturant du réseau R' .

Réciproquement, soit g un flot saturant du réseau R' . La fonction f définie sur les arcs de G par $f(u) = g(u) + a(u)$ est le seul flot compatible de R vérifiant $f' = g$.

La correspondance $f \mapsto f'$ est donc bien une bijection. On remarque de plus que les coûts des flots f et f' diffèrent de la constante $a \star c$, ce qui implique la seconde partie de la proposition. ■

Il résulte de la propriété précédente que la recherche d'un flot de valeur maximum de s à p dans le réseau R' permet de répondre au problème de l'existence d'un flot compatible pour le réseau R . En effet, si la valeur maximum d'un flot de s à p dans R' est inférieure strictement à α , le réseau R ne possède pas de flot compatible. Dans le cas contraire, un tel flot existe et il en est de même d'un flot compatible de coût minimum (voir section 2.3).

Un algorithme dual

Une donnée (G, b, s, p, c) d'un problème de flot maximum de coût minimum est spécifiée par un énoncé (G, b, s, p) d'un problème de flot maximum et une fonction c de $\mathcal{F}(A)$ définissant le coût sur chaque arc d'une unité de flux. Nous supposons dans cette section la fonction c positive ou nulle. Si α est la valeur maximum d'un flot de s à p pour le problème (G, b, s, p) , un flot maximum de coût minimum est un flot de valeur α dont le coût est minimum.

Un couple (f, π) où f est un flot du réseau R est dit *dual-réalisable* si les coûts réduits relatifs à π des arcs de G_f sont positifs ou nuls. Il est dit *primal-réalisable* si f est un flot maximum de s à p . Le principe de base de cet algorithme dual est de réaliser à chaque itération une augmentation de la valeur du flot à partir d'un chemin améliorant de coût réduit minimum dans le graphe d'écart. Comme nous le démontrerons, ce choix permet de déterminer un nouvel ensemble de potentiels pour lequel les arcs du nouveau graphe d'écart ont un coût réduit positif ou nul.

Initialisation

Le flot initial est le flot nul et le potentiel initial de chaque sommet est nul également. Comme la fonction c est positive ou nulle, les coûts réduits initiaux du premier graphe d'écart sont positifs ou nuls. Le couple $(0, 0)$ est donc dual-réalisable.

Itération courante

Soit (f, π) le couple courant dual-réalisable (et non primal-réalisable) résultat de l'itération précédente. Comme le flot f n'est pas de valeur maximum, il existe au moins un chemin améliorant de s à p dans le graphe d'écart G_f . Comme les coûts réduits des arcs du graphe d'écart G_f sont positifs ou nuls, il existe également un chemin améliorant μ de coût réduit minimum. L'itération courante réalise d'abord une augmentation de flot à partir de μ en exécutant la procédure AUGMENTER-FLOT(f, μ) et met ensuite à jour le potentiel de chaque sommet comme l'indique

la procédure MODIFIER-POTENTIELS(f, π) ci-dessous. Dans cette procédure, on note T l'ensemble des sommets accessibles à partir de s dans G_f et $\bar{l}_\pi(t)$ le coût réduit minimum d'un chemin de s à t dans G_f .

procédure MODIFIER-POTENTIELS(f, π);
 $M := \max_{t \in T} \bar{l}_\pi(t)$;
 pour tout sommet t de T faire $\pi(t) := \pi(t) + \bar{l}_\pi(t)$;
 pour tout sommet z de $S - T$ faire $\pi(z) := \pi(z) + M$.

Cette procédure augmente le potentiel de tout sommet t de T du coût réduit minimum d'un chemin de s à t dans G_f et augmente le potentiel des autres sommets d'une même quantité M égale au plus grand coût réduit minimum d'un chemin d'origine s dans G_f . L'itération courante, implémentée par la procédure CHANGER-DE-COUPLE ci-dessous réalise le changement de flot et la modification des potentiels :

procédure CHANGER-DE-COUPLE(f, π, μ);
 AUGMENTER-FLOT(f, μ);
 MODIFIER-POTENTIELS(f, π).

La proposition 4.12 montre que les coûts réduits du nouveau graphe d'écart et relatifs aux nouveaux potentiels sont positifs ou nuls.

Proposition 4.12. *Les coûts réduits du couple (f', π') calculés par la procédure CHANGER-DE-COUPLE(f, π, μ) sont positifs ou nuls.*

Preuve. Considérons d'abord le cas d'un arc v de $G_{f'}$ qui n'existait pas dans G_f . Si $v = u'$, l'arc u'' est un arc de μ . Comme μ est un chemin de coût réduit minimum pour le couple (f, π) , on a :

$$\bar{l}_\pi(u^-) = \bar{l}_\pi(u^+) + \bar{c}_\pi(u'').$$

Comme u^- et u^+ appartiennent à T , nous avons :

$$\bar{c}_{\pi'}(u') = c(u') + \pi'(u^-) - \pi'(u^+) = -c(u'') + (\pi(u^-) + \bar{l}_\pi(u^-)) - (\pi(u^+) + \bar{l}_\pi(u^+)).$$

En regroupant les termes correspondant à $\bar{c}_\pi(u'')$, il vient :

$$\bar{c}_{\pi'}(u') = -\bar{c}_\pi(u'') + \bar{l}_\pi(u^-) - \bar{l}_\pi(u^+) = 0.$$

Si $v = u''$, un calcul analogue montre que le nouveau coût réduit de l'arc v est nul.

Considérons maintenant un arc v de $G_{f'}$ qui est aussi dans G_f . Il résulte de la définition de T que seuls les trois cas suivants sont possibles :

(1) $v^- \in T, v^+ \in T$. On a alors par définition de \bar{l}_π :

$$\bar{c}_{\pi'}(v) = \bar{c}_\pi(v) + \bar{l}_\pi(v^-) - \bar{l}_\pi(v^+) \geq 0.$$

(2) $v^- \in S - T, v^+ \in S - T$. On a alors puisque (f, π) est dual-réalisable :

$$\bar{c}_{\pi'}(v) = \bar{c}_\pi(v) + M - M \geq 0.$$

(3) $v^- \in S - T, v^+ \in T$. On a alors par définition de M :

$$\bar{c}_{\pi'}(v) = \bar{c}_\pi(v) + M - \bar{l}_\pi(v^+) \geq 0.$$

Il en résulte que tous les nouveaux coûts réduits sont positifs ou nuls. ■

Terminaison

Chaque augmentation de la valeur du flot réalisée par la procédure CHANGER-DE-COUPLE porte sur au moins une unité de flux. L'algorithme se termine donc après au plus $B = \min\{b^-(p), b^+(s)\}$ itérations. A l'issue de la dernière itération, le flot f est de valeur maximum et les arcs de G_f ont un coût réduit positif ou nul par rapport au dernier ensemble de potentiels. La procédure FLOT-MAX-COÛT-MIN(G, b, s, p, c) ci-dessous décrit l'algorithme complet.

```

procédure FLOT-MAX-COÛT-MIN( $G, b, s, p, c$ );
   $f := 0$ ;  $\pi := 0$ ;  $T := S$ ;
  pour tout  $t$  de  $T$  faire calculer  $\bar{l}_\pi(t)$ ;
  tantque  $p \in T$  faire
    soit  $\mu$  un chemin améliorant de coût réduit  $\bar{l}_\pi(p)$ ;
    CHANGER-DE-COUPLE( $f, \pi, \mu$ );
     $T :=$ SOMMETS-ACCESSIBLES( $f, \pi$ );
    pour tout  $t$  de  $T$  faire calculer  $\bar{l}_\pi(t)$ 
  fintantque.

```

Complexité

Le nombre d'itérations de l'algorithme est majoré par $B = \min\{b^-(p), b^+(s)\}$ puisque chaque changement de couple augmente la valeur du flot d'au moins une unité. La complexité d'une itération est dominée par le calcul des coûts réduits minimum des chemins de s aux sommets de T . Pour ce calcul on peut utiliser l'algorithme de Dijkstra dont la complexité est $O(m \log n)$ pour un graphe d'écart

quelconque. Il en résulte une complexité globale $O(mB \log n)$ pour l'algorithme FLOT-MAX-COÛT-MIN(G, b, s, p, c). Cet algorithme n'est donc pas polynomial, mais seulement pseudo-polynomial en raison du terme B . La faiblesse de cet algorithme tient clairement au fait qu'une itération complète est réalisée pour une augmentation insuffisante de la valeur du flot.

8.4.5 Plan de transport de coût minimum

Nous considérons dans cette section un problème de transport spécifié par un graphe $G = (S, A)$, une fonction de coût de $\mathcal{F}(A)$ positive ou nulle et une fonction d de $\mathcal{F}(S)$, définissant l'offre et la demande et vérifiant $\sum_{s \in S} d(s) = 0$. La fonction d permet de classer les sommets en sommets fournisseurs (si $d(s) > 0$), sommets clients (si $d(s) < 0$) et sommets de transit (si $d(s) = 0$). On note respectivement F , C et T les sous-ensembles des sommets fournisseurs, clients et de transit. On suppose de plus qu'il existe au moins un chemin de G liant un sommet fournisseur quelconque à un sommet client quelconque. Un *plan de transport* est une fonction $x : A \mapsto \mathbb{Q}^+$ telle que :

$$\begin{aligned} \forall u \in A, \quad x(u) &\geq 0 \\ \forall s \in S, \quad e_x(s) &= -d(s). \end{aligned}$$

Un plan de transport est de coût minimum si son coût $c \star x$ est minimum.

Ce problème est un cas particulier important du problème du flot maximum de coût minimum, il se pose en effet fréquemment dans les applications.

Le graphe d'écart G_x d'un plan de transport x est défini de la même manière que le graphe d'écart G_f d'un flot f , à cette remarque près que chaque arc de G possède un représentant conforme valué par $+\infty$. Tout chemin de G produit donc un chemin conforme dans G_x passant par les mêmes sommets. Un raisonnement tout à fait analogue à celui mené dans la section 8.4.4 permet d'établir une condition d'optimalité d'un couple (x, π) où x est un plan de transport et π un potentiel sur les sommets de S . Cette proposition est en fait un corollaire de la proposition 4.2.

Proposition 4.13. *Un plan de transport x et un potentiel π sont optimaux si et seulement si les coûts réduits des arcs du graphe d'écart G_x sont positifs ou nuls.*

L'algorithme de Edmonds et Karp que nous allons décrire est une extension de l'algorithme FLOT-MAX-COÛT-MIN du paragraphe précédent. Il s'agit donc d'un algorithme dual qui repose sur l'idée qu'il convient de satisfaire en priorité les couples (fournisseur, client) qui sont les plus déficitaires du point de vue de l'offre et de la demande. A cet effet, on appellera *pseudo-plan* une fonction x de $\mathcal{F}(A)$ positive ou nulle et l'on notera $\delta_x(s) = d(s) - e_x(s)$ le *déficit* du sommet s pour le pseudo-plan x . Si un couple (x, π) dual-réalisable est tel que tous les déficits sont nuls, alors x est un plan de transport de coût minimum.

Soit x un pseudo-plan et soit Δ un entier naturel. Nous notons $P_x(\Delta)$ (respectivement $N_x(\Delta)$) le sous-ensemble des sommets dont le déficit est supérieur ou égal à Δ (respectivement inférieur ou égal à $-\Delta$). Le pseudoplan x est dit Δ -*optimal* si au moins l'un des deux ensembles $P_x(\Delta)$ ou $N_x(\Delta)$ est vide. Une phase de l'algorithme de Edmonds et Karp consiste à transformer un pseudoplan Δ -optimal en un pseudoplan $\Delta/2$ -optimal en exécutant au plus n fois l'algorithme de Dijkstra pour la recherche des chemins de coût réduit minimum.

Initialisation

L'algorithme est initialisé par le pseudoplan $x = 0$, le potentiel $\pi = 0$ et $\Delta = B$ où $B = \max_{s \in S} d(s)$. Remarquons que ce premier pseudoplan est Δ -optimal puisque $P_x(\Delta) = \emptyset$.

Phase courante

Soit x le pseudoplan courant 2Δ -optimal résultant de la phase précédente. Une itération de la phase suivante choisit d'abord un couple de sommets (s, t) où $s \in P_x(\Delta)$ et $t \in N_x(\Delta)$. Elle détermine ensuite un chemin μ de coût réduit minimal dans G_x de s à t , diminue les déficits des sommets s et t de la quantité Δ en augmentant de Δ les flux transportés par les arcs du chemin de G formé par les pères des arcs de μ . Elle modifie enfin les potentiels pour que le nouveau pseudoplan soit dual-réalisable. Cette itération est répétée tant que le pseudoplan x n'est pas Δ -optimal. La procédure ci-dessous réalise la phase courante de l'algorithme de Edmonds et Karp.

```

procédure PHASE-EDMONDS-KARP( $\Delta$ );
   $P := \{s \mid \delta_x(s) \geq \Delta\}$ ;
   $N := \{s \mid \delta_x(s) \leq -\Delta\}$ ;
  tantque ( $P \neq \emptyset$  et  $N \neq \emptyset$ ) faire
    choisir  $s$  dans  $P$  et  $t$  dans  $N$ ;
    pour tout sommet  $z$  de  $S$  faire
      calculer  $\bar{l}_\pi(s, z)$ 
    finpour;
    soit  $\mu$  un chemin de coût réduit minimum de  $s$  à  $t$ ;
    MODIFIER-PSEUDOPLAN( $x, \mu$ );
    pour tout sommet  $z$  faire  $\pi(z) := \pi(z) + \bar{l}_\pi(s, z)$ ;
  fintantque;
   $\Delta := \Delta/2$ .

```

La procédure MODIFIER-PSEUDOPLAN est une variante simplifiée de la procédure AUGMENTER-FLOT qui augmente systématiquement de Δ les flux des pères des arcs de μ .

Terminaison

L'algorithme de Edmonds et Karp se termine dès que $\Delta < 1$. Tous les déficits sont alors nuls. D'où la procédure pour l'algorithme complet :

```

procédure EDMONDS-KARP( $R$ );
  ( $x, \pi$ ) := (0, 0);  $\Delta := B$ ;
  tantque  $\Delta \geq 1$  faire
    PHASE-EDMONDS-KARP( $\Delta$ )
  fintantque.

```

Les figures 4.2 et 4.3 illustrent le comportement de l'algorithme sur un exemple. Pour chaque sommet, le coin sud-est contient le coût réduit minimum de l'origine au sommet, le coin nord-est contient le potentiel courant, le coin nord-ouest contient le nom du sommet et le coin sud-ouest contient le déficit de ce sommet. Sur chaque arc est inscrit le coût réduit par rapport au potentiel en cours. La partie supérieure de la figure 4.2 représente le réseau initial qui est aussi le premier réseau d'écart; sa partie inférieure représente le réseau d'écart après la première phase ($\Delta = 2$). Cette première phase choisit le chemin (arcs épais) $\mu_1 = (a, e, f, c)$ et augmente les flux de ses arcs de trois unités. La figure 4.3 correspond à la

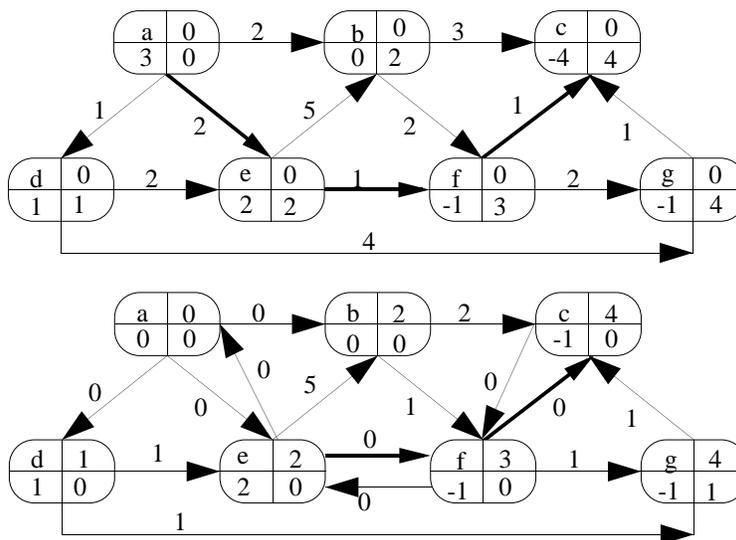
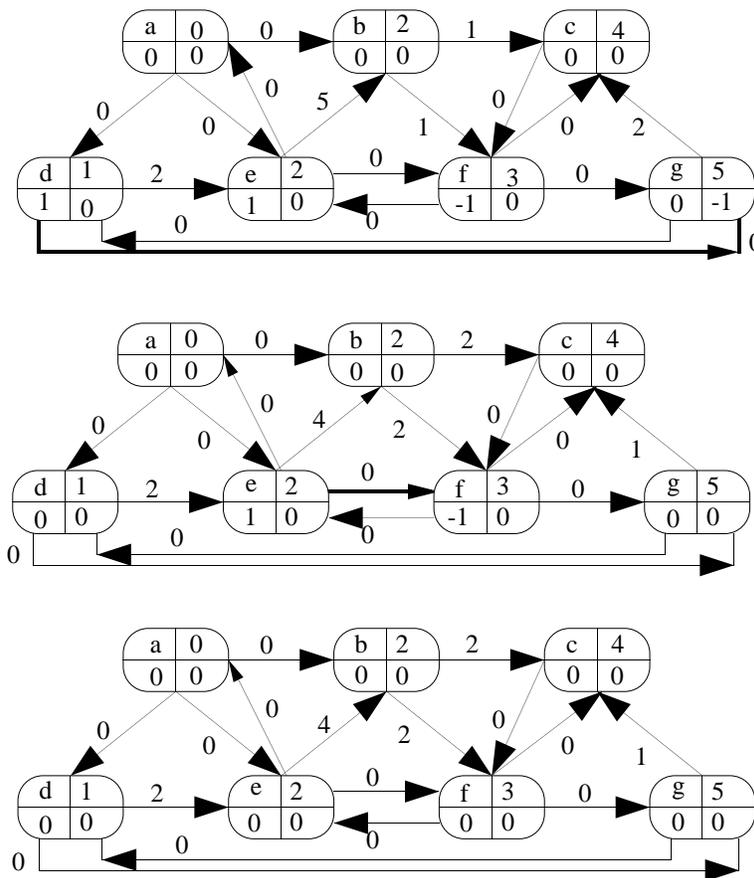


Figure 4.2: Première phase.

seconde phase ($\Delta = 1$). Les chemins successivement choisis sont $\mu_2 = (e, f, c)$, $\mu_3 = (d, g)$ et $\mu_4 = (e, f)$, et chacun conduit à une réduction du déficit de une unité. Après ces trois réductions, le déficit total est nul et le plan $3\chi_{\mu_1} + \chi_{\mu_2} + \chi_{\mu_3} + \chi_{\mu_4}$ est optimal.

Figure 4.3: *Seconde phase.*

Convergence et Complexité

Comme l'algorithme exécute exactement $1 + \log B$ phases, il se termine si chaque phase réalise un nombre fini d'itérations. Comme nous allons montrer que chaque phase réalise au plus n itérations, l'algorithme se termine et le dernier pseudoplan est un plan de transport optimal car tous les déficits sont nuls.

Lemme 4.14. *Une phase exécute au plus n itérations.*

Preuve. Supposons qu'une phase se termine parce que l'ensemble $P_x(2\Delta)$ est vide. Considérons alors une itération quelconque de la phase suivante et soit μ le chemin de G_x d'origine s et d'extrémité t associé à cette itération. Le déficit du sommet s est réduit de Δ unités par construction et par conséquent le sommet s dont le déficit devient strictement inférieur à Δ ne sera plus candidat à une réduction de déficit au cours de cet même phase. Un raisonnement analogue peut être fait si c'est l'ensemble $N_x(2\Delta)$ qui est vide. Le lemme en résulte. ■

La complexité de l'algorithme de Edmonds et Karp est $O(nT(m,n) \log B)$ où $T(m,n)$ est la complexité de l'algorithme de Dijkstra utilisé pour déterminer à chaque itération les chemins de coût réduit minimum.

Lorsque les capacités maximales du réseau initial R sont finies, il est possible (voir exercices) de le remplacer par un réseau «équivalent» R' sans limitations de capacités et comportant $n+m$ sommets et $2m$ arcs. En appliquant l'algorithme de Edmonds et Karp au réseau R' et en utilisant une version modifiée de l'algorithme de Dijkstra permettant le calcul des coûts réduits minimum pour le réseau R' avec la même complexité $T(m, n)$ que sur le réseau R , on aboutit à un algorithme de complexité $O(\min\{m \log B, m \log n\}T(m, n))$ qui est l'un des plus efficaces connus aujourd'hui.

Notes

Comme les flots constituent le modèle de base de nombreuses applications, ils ont fait l'objet d'une recherche très intense depuis les premiers résultats de Ford et Fulkerson en 1956. Ces recherches ont eu essentiellement pour objectif de développer des algorithmes de plus en plus efficaces. Citons deux ouvrages qui contiennent les résultats les plus récents du domaine :

R.E. Tarjan, *Data Structures and Network Algorithms*, SIAM Philadelphia, chapitre 7, 1983

R.K. Ahuja, T.L. Magnanti, et J.B. Orlin, *Network Flows*, Sloan School of Management, Work Report, MIT, 1988

L'idée, due à Edmonds et Karp, de choisir comme chemins améliorants les plus courts en nombre d'arcs a conduit à un algorithme de complexité $O(m^2n)$. Indépendamment, E.A. Dinic a introduit les réseaux à niveaux qui sont les sous-graphes du réseau initial contenant les sommets et les arcs appartenant à au moins un plus court chemin améliorant. Par un calcul de flots bloquants dans ces réseaux à niveaux, il obtient un flot maximum en temps $O(n^2m)$.

E.A. Dinic, Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation, *Soviet Math. Dokl.*, 11(1970), 1277-1280

La notion de distance estimée au puits, due à Goldberg, est une variante du calcul des réseaux à niveaux de Dinic. Elle présente le triple avantage d'être de compréhension plus aisée, de manipulation plus simple et de conduire à des algorithmes plus performants. L'algorithme des distances estimées présenté dans la section 8.3.2 énumère exactement les mêmes chemins améliorants que l'algorithme de Dinic.

La première variante de l'algorithme du préflot est due à A.V. Karzanov qui l'a introduite pour les réseaux à niveaux.

A.V. Karzanov; Determining the Maximal Flow in a Network by the method of Preflows, *Soviet Math. Dokl.*, 15(1974), 434-437

La variante de l'algorithme du préflot obtenue par échelonnement des excès est due à J.B. Orlin et R.K. Ahuja.

Le problème du flot de coût minimum a aussi été très étudié. Citons quelques étapes depuis les travaux de Ford et Fulkerson en 1962 qui présentèrent les premiers algorithmes primal-dual comme l'algorithme «out of kilter». Les améliorations sur des chemins du graphe d'écart de coût réduit minimum sont dues indépendamment à Jewell, Iri, Busacker et Gowen. L'utilisation simultanée des ensembles de potentiel pour obtenir des chemins dont les arcs sont de coût réduit positif est due à Edmonds et Karp. C'est dans ce cadre que nous avons présenté l'algorithme dual pour le problème du flot maximum de coût minimum.

Le concept d' ϵ -optimalité fut introduit par Bertsekas en 1979. L'algorithme primal de Goldberg et Tarjan de la section 8.4.3 ainsi que sa variante fortement polynomiale sont développés dans :

A.V.Goldberg et R.E.Tarjan; Finding Minimum Cost Circulations by Canceling Negative Cycles, *Proc. 20th ACM Symp. on the Theory of Computing*, 388-397,(1988).

Pour un problème de transport, l'idée de réaliser un échelonnement des déficits des sommets est due à Edmonds et Karp. L'algorithme qui en résulte pour la recherche d'un plan de transport de coût minimum est développé dans :

J. Edmonds et R.M. Karp, Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems, *J. ACM*,19,(1972),248-264.

Cet algorithme a été étendu aux réseaux avec capacités maximales par J.B. Orlin qui a obtenu une complexité en temps de $O(m \log n(m + n \log n))$ dans :

J.B. Orlin; A Faster Strongly Polynomial Minimum-Cost Flow Algorithm, *Proc. 20th ACM Symp. on the Theory of Computing*, 377-387,(1988).

Exercices

8.1. Soit $R = (G, a, b)$ un réseau.

a) Construire à partir de R un réseau de transport R' muni d'une source s et d'un puits p tel que R est consistant si et seulement si R' possède un flot saturant.

b) En déduire un algorithme pour déterminer si un réseau est consistant.

8.2. Soit $G = (S, A)$ un graphe orienté biparti où $S = X \cup Y$, $X \cap Y = \emptyset$ et $A \subset X \times Y$. On appelle *couplage* de G une partie C de A telle que deux arcs quelconques de C n'ont pas d'extrémité commune. Un couplage C est dit parfait si tout sommet de X est l'origine d'un arc de C .

a) Démontrer que les couplages de G sont en bijection avec les flots entiers d'un réseau que l'on construira.

b) En déduire que G possède un couplage parfait si et seulement si :

$$\forall T \subset X, \quad \text{Card}(\Gamma^+(T)) \geq \text{Card}(T)$$

8.3. Soit $G = (S, A)$ un graphe orienté connexe à n sommets et m arcs et soit T un sous-ensemble de sommets. On appelle *cocycle associé à T* le sous-ensemble d'arcs $\omega(T) = \omega^+(T) \cup \omega^-(T)$. On associe au cocycle $\omega(T)$ le vecteur caractéristique \vec{T} de $\{0, 1, -1\}^m$ défini par :

$$\vec{T}(u) = \begin{cases} 1 & \text{si } u \in \omega^+(T) \\ -1 & \text{si } u \in \omega^-(T) \\ 0 & \text{sinon} \end{cases}$$

a) Montrer que les vecteurs caractéristiques d'un cycle et d'un cocycle quelconques sont orthogonaux. Montrer que le sous-espace vectoriel de \mathbb{R}^m engendré par les cocycles est de dimension $n - 1$.

8.4. Démontrer la proposition 2.8 . **8.5.** Soit $R = (G, b, s, p)$ le réseau d'un problème de flot maximum. Construire un réseau R contenant plusieurs coupes de valeur minimum. On note $\mathcal{C}(R)$ l'ensemble des coupes de valeur minimum du réseau R . Montrer que $\mathcal{C}(R)$ est stable pour l'union et l'intersection.

8.6. Démontrer la proposition 4.1.

8.7. Soit $G = (S, A)$ un graphe orienté fortement connexe et $c : A \mapsto \mathbb{Z}$ une fonction coût sur les arcs. On suppose qu'il n'existe pas de circuit de coût strictement négatif. On note $F_k(x)$ le coût minimum d'un chemin de k arcs de s à x dans G en posant $F_k(x) = +\infty$ si un tel chemin n'existe pas. Si γ est un circuit de G à p arcs, le *coût moyen* de γ est défini par : $\bar{c}(\gamma) = c(\gamma)/p$. Il s'agit alors de calculer par l'*algorithme de Karp* le coût moyen minimum d'un circuit de G , c'est-à-dire la valeur : $\lambda^* = \min\{\bar{c}(\gamma) \mid \gamma \in \mathcal{C}(G)\}$ où $\mathcal{C}(G)$ est l'ensemble des circuits de G .

a) Montrer que :

$$\min_{x \in S} \max_{k \in \{0, \dots, n-1\}} \left\{ \frac{F_n(x) - F_k(x)}{n - k} \right\} = 0 \Rightarrow \lambda^* = 0$$

On notera $\pi(x)$ le coût minimum d'un chemin de s à x dans G et on montrera que :

$$\max_{k \in \{0, \dots, n-1\}} \left\{ \frac{F_n(x) - F_k(x)}{n - k} \right\} \geq 0$$

et que l'égalité a lieu si et seulement si $F_n(x) = \pi(x)$. On montrera ensuite qu'il existe un sommet y tel que $F_n(y) = \pi(y)$.

En déduire que :

$$\lambda^* = \min_{x \in S} \max_{k \in \{0, \dots, n-1\}} \left\{ \frac{F_n(x) - F_k(x)}{n - k} \right\}$$

8.8. Soit (G, b, s, p) le réseau d'un problème de flot maximum. On suppose que :

$$2^{K-1} < B = \max_{u \in A} b(u) \leq 2^K - 1$$

On peut donc coder chaque capacité avec au plus K bits. Si p est un entier naturel codé sur K bits, on note $p^{(k)}$, $k \in \{1, \dots, K\}$, l'entier associé aux k bits de poids fort de p (par exemple si $p = 7$ et $K = 5$, alors $p^{(3)} = 1$). On considère le problème $P_k = (G, b_k, s, p)$ où $b_k(u) = b(u)^{(k)}$.

a) Montrer que

$$b_{k+1}(u) \in \{2b_k(u), 2b_k(u) + 1\}$$

Soit f_k^* la valeur maximum d'un flot de P_k , montrer que

$$f_{k+1}^* \leq 2f_k^* + m$$

où m est le nombre d'arcs du réseau.

b) En déduire une variante polynomiale de l'algorithme «générique» de Ford et Fulkerson, de complexité $O(nm \log B)$.

8.9. Soit $R = (G, a, b, c)$ un réseau valué muni d'une fonction d'offre et de demande $d : S \mapsto \mathbb{Z}$ telle que $\sum_S d(s) = 0$. Soit u un arc de G dont la capacité minimale $a(u)$ est strictement positive.

a) Montrer que l'on peut transformer $d(u^-)$, $d(u^+)$, $a(u)$ et $b(u)$ en $d'(u^-)$, $d'(u^+)$, $a'(u) = 0$ et $b'(u)$ de sorte que les flots de R et de R' soient en bijection et que les coûts de deux flots associés dans la bijection diffèrent d'une constante.

Soit $R = (G, 0, b, c)$ un réseau valué muni d'une fonction d'offre et de demande $d : S \mapsto \mathbb{Z}$ telle que $\sum_S d(s) = 0$. Soit u un arc de G .

b) Montrer que l'on peut transformer l'arc u en deux arcs u_1 et u_2 de capacité minimale nulle et de capacité maximale infinie, tels que :

$$u_1^- = u^-, \quad u_2^- = u^+, \quad u_1^+ = u_2^+ = \alpha \text{ où } \alpha \text{ est un nouveau sommet ;}$$

les flots de R et du nouveau réseau R' se correspondent dans une bijection conservant le coût.