2A – Optimisation Combinatoire

Feuille : Plus courts chemins

ALGORITHMES

Algorithme recherche en largeur d'abord :

Entrée : $G = (V, A)$ graphe orienté et $s \in V$.

Sortie : Les ensembles $V_0, V_1, ..., V_k, V_\infty$ tels que si $p \in V_i$ alors la distance de s à p est i, une s-arborescence F sur $G[\bigcup_1^k V_i]$ telle que $\forall v \in V_i$ $(i \leq k)$, $F[s, v]$ (le chemin unique de s à v dans F) est de la longueur i.

Etape 0 : $V_0 := \{s\},\ i := 0,\ S_0 := \{s\},\ F_i := \emptyset$.

Etape 1 : Tant que $\delta^+(S_i) \neq \emptyset$:

$V_{i+1} := \{v \in V \setminus S_i$: il existe $u_v \in V_i$ tel que $u_v v \in A\}$,

$\forall\ v \in V_{i+1}$ soit $u_v \in V_i$ tel que $u_v v \in A$,

$F_{i+1} := F_i + \{u_v v : v \in V_{i+1}\},\ S_{i+1} := S_i \cup V_{i+1},\ i := i + 1$.

Etape 2 : $V_\infty := V - S_i$ et arrêter.

Algorithme de Bellman :

Entrée : $(G = (V, A), c)$ réseau sans circuit et une racine s de G.

Sortie : $\forall v \in V$ $l(v)$ le coût d'un chemin de s à v de coût minimum, une s-arborescence F de G telle que pour chaque sommet $v \in V$ le chemin unique de s à v dans F est de coût $l(v)$.

Etape 0 : $l(s) := 0,\ F_1 := \emptyset,\ G_1 := G,\ v_1 := s,\ i := 1,\ n := |V|$.

Etape 1 : $G_{i+1} := G_i - v_i$.

Etape 2 : Soit $v_{i+1} \in V(G_{i+1})$ tel que $d^-_{G_{i+1}}(v_{i+1}) = 0$.

Etape 3 : $l(v_{i+1}) := \min\{l(v_j) + c(v_j v_{i+1}) : j \leq i,\ v_j v_{i+1} \in A\}$.

Etape 4 : $F_{i+1} := F_i + v_j v_{i+1}$ où $l(v_{i+1}) = l(v_j) + c(v_j v_{i+1})$ tel que $j \leq i,\ v_j v_{i+1} \in A$.

Etape 5 : $i := i + 1$. Si $i < n$ aller à l'Etape 1.

Etape 6 : Arrêter.

Algorithme de Dijkstra :

Entrée : $(G = (V, A), c)$ réseau tel que $c \geq 0$ et $s \in V$.

Sortie : l'ensemble S de sommets qui peuvent être atteints par un chemin de s, $\forall v \in V$ $l(v)$ le coût d'un chemin de s à v de coût minimum, une s-arborescence F sur S telle que pour chaque sommet $v \in S$ le chemin unique de s à v dans F est de coût $l(v)$.

Etape 0 : $l_0(s) := 0,\ l_0(v) := \infty,\ a_0(v) := \emptyset\ \forall v \in V - s,\ S_0 := \emptyset,\ F_0 := \emptyset,\ i := 0$.

Etape 1 : tant qu'il existe $u \in V - S_i$ tel que $l_i(u)$ soit fini faire

Soit $v_i \in V - S_i$ tel que $l_i(v_i) = \min\{l_i(u) : u \in V - S_i\}$.

$S_{i+1} := S_i \cup v_i,\ F_{i+1} := F_i \cup a_i(v_i)$.

$\forall v_i u \in A$ tel que $u \in V - S_{i+1}$

si $l_i(u) > l_i(v_i) + c(v_i u)$ alors $l_{i+1}(u) := l_i(v_i) + c(v_i u)$ et $a_{i+1}(u) := v_i u$

sinon $l_{i+1}(u) := l_i(u)$ et $a_{i+1}(u) := a_i(u)$.

$i := i + 1$.

Etape 2 : Arrêter.

ALGORITHME DE BELLMAN-FORD :
ENTRÉE : $(G = (V, A), c)$ réseau sans circuit absorbant et s une racine de G.
SORTIE : Le coût $w(v)$ d'un chemin de s à v de coût minimum $\forall v \in V$.
Etape 0 : $w_0(s) := 0$, $w_0(v) := \infty$ $\forall v \in V - s$, $k := 0$, $n := |V|$.
Etape 1 : Tant que $k < n$:
$w_{k+1}(v) = \min\{w_k(v), c(uv) + w_k(u) : uv \in A\}$.
Etape 2 : $w(v) := w_n(v)$ $\forall v \in V$.
Etape 3 : Arrêter.

ALGORITHME DE FORD-FULKERSON-DANTZIG :
ENTRÉE : $(G = (V, A), c)$ réseau et une racine s de G.
SORTIE : (i) Soit une s-arborescence F de G tel que pour chaque sommet $v \in V$ le chemin unique de s à v dans F soit de coût minimum dans (G, c).
(ii) Soit un circuit absorbant C.
Pour une s-arborescence F et $v \in V(F) - s$, l'arc unique de F qui entre v est noté par $F(v)$.
Etape 1 : En exécutant l'algorithme de marquage orienté trouver une s-arborescence F_1 de G.
$i := 1$.
Etape 2 : Pour tout $v \in V$, soit $\pi_i(v)$ le coût du chemin unique de s à v dans F_i.
Etape 3 : Si $\pi_i(v) - \pi_i(u) \leq c(uv)$ pour tout $uv \in A$ alors arrêter avec $F := F_i$.
Etape 4 : Sinon, soit uv un arc pour lequel $\pi_i(v) - \pi_i(u) > c(uv)$.
Etape 5 : Si $F_i + uv$ contient un circuit C alors arrêter avec C.
Etape 6 : Sinon, $F_{i+1} := F_i + uv - F_i(v)$, $i := i + 1$ et aller à l'Etape 2.

Exercice 2.1. — (**Justification de l'algorithme de Ford-Fulkerson-Dantzig**)
(a) Montrer que pour chaque i, F_i est une s-arborescence de G.
(b) Montrer que si on s'arrête à l'Etape 5, alors C est un circuit absorbant.
(c) Montrer que si on s'arrête à l'Etape 3, alors F est l'arborescence désirée.
(d) Montrer que $\sum_{w \in V} \pi_{i+1}(w) < \sum_{w \in V} \pi_i(w)$.
(e) Montrer que l'algorithme s'arrête en temps fini.
(f) Appliquer l'algorithme de Ford-Fulkerson-Dantzig dans le réseau de la FIG. 1 pour trouver les plus courts chemins issus du sommet s ou pour trouver un circuit absorbant en commençant avec l'arborescence donnée. On étudiera les deux cas :
(1) $d = 1$,
(2) $d =$-3.

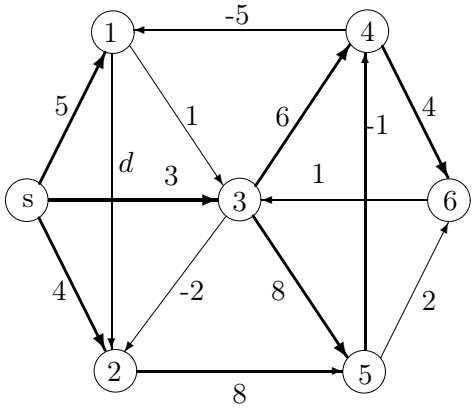

FIGURE 1 – Le réseau pour l'Exercice 2.1

THÉORIE

Exercice 2.2. — Soit P un plus court chemin de s à p et soit $P[x,y]$ la portion de P située entre x et y. Montrer que $P[x,y]$ est un plus court chemin de x à y.

Exercice 2.3. — Soit (G,c) un réseau sans circuit absorbant. On ajoute un nouveau sommet s et pour chaque sommet v on ajoute l'arc sv avec coût 0. Soit (G',c') le nouveau réseau.

(a) Montrer que (G',c') est un réseau sans circuit absorbant.

(b) Montrer que si (G',c') admet un potentiel alors (G,c) admet un potentiel.

Exercice 2.4. — **(Caractérisation des réseaux sans circuit absorbant)** Soit (G,c) un réseau. Montrer que (G,c) est sans circuit absorbant si et seulement si (G,c) admet un potentiel.

Exercice 2.5. — **(Théorème min-max)** Soit (G,c) un réseau sans circuit absorbant et soit s une racine de G. Montrer que pour chaque sommet t, $\min\{c(P) : P \ (s,t)\text{-chemin}\} = \max\{\pi(t) - \pi(s) : \pi \text{ potentiel}\}$.

Exercice 2.6. — Soit π un potentiel d'un réseau (G,c). Un arc uv est dit π-serré si $c(uv) = \pi(v) - \pi(u)$. Soit P un (s,t)-chemin de G tel que chaque arc de P soit π-serré. Montrer que P est un (s,t)-chemin de coût minimum dans (G,c).

Exercice 2.7. — Soient P un plus court chemin de s à t et π le potentiel canonique d'un réseau (G,c). Montrer que chaque arc de P est π-serré.

Exercice 2.8. — Soit (G,c) un réseau sans circuit absorbant et soit s une racine de G. Montrer qu'il existe une s-arborescence F de G telle que pour, chaque sommet v, le chemin de s à v dans F soit de coût minimum dans (G,c).

APPLICATIONS

Exercice 2.9. — **Problème du sac à dos** : Etant donnés un sac à dos de volume K et des objets O_i de volume a_i et d'utilité u_i $(i = 1,2,...,\ell)$, il faut choisir quelques uns des objets de telle sorte que la somme de leurs volumes soit inférieure ou égale à K et que la somme de leurs utilités soit maximum.

On introduit le graphe auxiliaire $D = (V,A)$ tel que

$V := \{v_{i,j} : 1 \le i \le \ell,\ 0 \le j \le K\} \cup \{v_{0,0}, v_{\ell+1,K}\}$ et $A := \cup_{i=0}^{\ell} A_i \cup_{i=0}^{\ell-1} A'_i$ où

$A_0 := v_{0,0}v_{1,0}$, $A'_0 := v_{0,0}v_{1,c_1}$, $A_\ell := \{v_{\ell,j}v_{\ell+1,K} : 0 \le j \le K\}$,

$A_i := \{v_{i,j}v_{i+1,j} : 0 \le j \le K\}$ et $A'_i := \{v_{i,j}v_{i+1,j+a_{i+1}} : 0 \le j \le K - a_{i+1}\}$ pour $1 \le i \le \ell - 1$.

Soit le coût d'un arc e est 0 si $e \in \cup_{i=0}^{\ell} A_i$ et u_{i+1} si $e \in A'_i$. Soient $s := v_{0,0}$ et $t := v_{\ell+1,K}$.

(a) Montrer que D est sans circuit.

(b) Montrer qu'il existe une bijection entre les solutions du problème du sac à dos et les (s,t)-chemins dans D.

(c) Montrer que le coût maximum d'un (s,t)-chemin dans D est la solution optimale du problème du sac à dos.

(d) Résoudre le problème du sac à dos où le volume du sac est 12 et il y a quatre objets A, B, C, D :

Objets	A	B	C	D
Volumes	2	3	5	7
Valeurs	3	4	8	9

Exercice 2.10. — Approximation d'une fonction linéaire par morceaux : Soit $f_1(x)$ une fonction linéaire par morceaux définie par N points successifs (x_i, y_i) où $x_1 < x_2 < \ldots < x_N$. En choisissant certains de ces points (de l'indice $1 = i_1 < \ldots < i_j < \ldots < i_\ell = N$) on a une autre fonction linéaire par morceaux $f_2(x)$ dont le stockage est moins coûteux mais dont l'erreur de précision est plus coûteuse. Il s'agit de trouver une telle approximation $f_2(x)$ de $f_1(x)$ de coût minimum où le coût de $f_2(x)$ est $\sum_{j=2}^{\ell}(\alpha + \beta \sum_{k=i_{j-1}}^{i_j} |f_1(x_k) - f_2(x_k)|)$.

Exercice 2.11. — Allocation des contrôles sur une chaîne de production : Une chaîne de production est une liste de n étapes de production. Chaque étape consiste en une phase de fabrication (de coût unitaire p_i) éventuellement suivie d'une séance de contrôle (où on contrôle soit tous les objets soit aucun). La i-ième phase de fabrication peut introduire des défauts dans chaque objet avec la probabilité α_i. On veut qu'à la fin tous les objets soient sans défaut. Le problème se pose d'optimiser la position des contrôles sachant que le coût unitaire du contrôle à la phase i dépend de la position j du contrôle précédent, et est noté $c(j, i)$.

Exercice 2.12. — Transport maritime : Une compagnie maritime offre des transports réguliers entre les différentes villes autours du Lac Balaton en Hongrie. On sait que le voyage de la ville i à la ville j dure t_{ij} minutes et rapporte p_{ij} Forints. Le rapport moyen d'une tournée C est $\frac{\sum_{ij \in C} p_{ij}}{\sum_{ij \in C} t_{ij}}$. Le problème du directeur de la compagnie est de décider s'il existe une tournée (un circuit) dont le rapport moyen est supérieur à une valeur fixée.

Exercice 2.13. — Système d'équations formé de différences : Étant donné un système d'équations dont toutes les inégalités sont de la forme $x_{j_k} - x_{i_k} \leq b_k$, décider s'il existe une solution réalisable. Si oui, trouver une telle solution et si non, trouver les contraintes contradictoires.

Exercice 2.14. — Echange de devises : Un agent de change reçoit de la bourse les taux de change t_{ij} pour chaque couple de devises i et j : à un client qui lui donne une unité de devise i il doit rendre t_{ij} unités de devise j. Le problème est de savoir s'il existe une suite circulaire d'échanges de devises qui pourrait diminuer le capital de l'agent si un client demande de la réaliser.