

Recherche Opérationnelle 1A

Théorie des graphes

Plus courts chemins

Zoltán Szigeti

Laboratoire G-SCOP
INP Grenoble, France

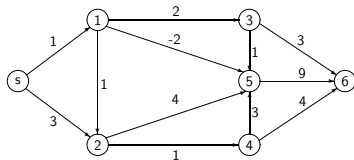
Plus courts chemins

Problème

Arriver d'un point à un autre point le plus vite possible.

Modélisation

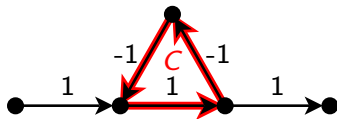
- ❶ **réseau** : (G, c) où
 - ❶ $G = (V, A)$ un graphe orienté,
 - ❷ $c : A \rightarrow \mathbb{R}$ un coût sur les arcs.
- ❷ **distance** : $\text{dist}_{(G,c)}(s, t) = \text{coût minimum d'un } (s, t)\text{-chemin.}$



Problème de circuit absorbant

Définition

Circuit absorbant : un circuit C dont le coût total est négatif, $c(C) < 0$.



Remarques

- ❶ avec circuit absorbant :
 - ❶ chemin : pas de solution optimale,
 - ❷ chemin élémentaire : problème (NP-)difficile.
- ❷ sans circuit absorbant :
 - ❶ les deux problèmes coïncident,
 - ❷ on peut les résoudre.

Réseaux sans circuit absorbant

Remarque

On calcule un plus court chemin de s à tous les sommets.

Propriété des sous-chemins optimaux

Tout sous-chemin d'un plus court chemin est un plus court chemin.

Algorithmes pour les cas spéciaux:

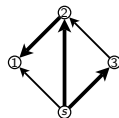
- ① pas de circuit : algorithme de Bellman,
- ② pas de coût négatif : algorithme de Dijkstra,
- ③ pas de circuit absorbant : algorithme de Bellman-Ford.

Définitions

- ❶ **racine** d'un graphe orienté G : un sommet s tel que dans G chaque sommet peut être atteint par un chemin à partir de s .
- ❷ **s -arborescence** de G : un graphe partiel F de G tel que
 - ❶ sans cycle et s racine de F , \iff
 - ❷ sans circuit et $d_F^-(v) = 1 \ \forall v \in V \setminus s$. \iff
 - ❸ il existe un et un seul chemin dans F de s à chaque sommet.

Théorème

s est une racine de $G \iff G$ possède une s -arborescence.

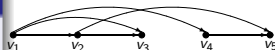


Démonstration

- ❶ suffisance : par définition.
- ❷ nécessité : l'algorithme Marquage fournit une s -arborescence.

Théorème

G est sans circuit \iff il existe un ordre **topologique** v_1, \dots, v_n de $V(G)$: si $v_i v_j \in A(G)$ alors $i < j$.



Démonstration

- 1 suffisance : l'ordre topologique montre qu'on ne peut jamais revenir.
- 2 nécessité : on peut construire l'ordre topologique en choisissant successivement un sommet de degré entrant zéro dans le graphe qui reste après avoir effacé les sommets déjà choisis.

l'idée de l'algorithme pour trouver les distances s'il n'y a pas de circuit

On suppose que s est une racine. Par l'optimalité des sous-chemins, les distances peuvent être calculées dans l'ordre topologique :

- 1 $v_1 := s$ et $\ell(v_1) := 0$, et
- 2 $\ell(v_{i+1}) := \min\{\ell(v_j) + c(v_j v_{i+1}) : v_j v_{i+1} \in A\}$, $\ell(v_j)$ est déjà connu !

Algorithme de Bellman :

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

SORTIE : fonction $\ell : V \rightarrow \mathbb{R}$ et s -arborescence F de G tq $\forall v \in V$:
le (s, v) -chemin unique dans F est de coût $\ell(v) = \text{dist}_{(G, c)}(s, v)$.

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

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

Etape 2: **Choix du sommet.** Soit $v_{i+1} \in V(G_{i+1})$ tel que $d_{G_{i+1}}^-(v_{i+1}) = 0$.

Etape 3: **Calcul de la distance.**

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

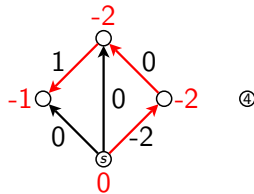
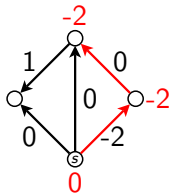
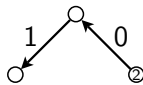
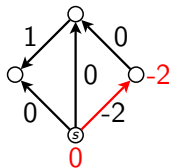
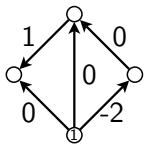
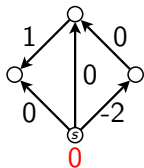
Etape 4: **Construction de l'arborescence.**

$$F_{i+1} := F_i + v_j v_{i+1} \text{ où } v_j v_{i+1} \text{ donne le minimum à l'Etape 3.}$$

Etape 5: **Test d'arrêt.** $i := i + 1$. Si $i < n$ ALLER à l'Etape 1 sinon STOP.

Dans un réseau sans circuit on peut trouver un **plus long** chemin aussi, en remplaçant min par max à l'Etape 3.

Exécution de l'algorithme de Bellman



④

Pas de coût négatif

l'idée de l'algorithme pour trouver les distances si c est positif

- ❶ à chaque étape :
 - ❶ V est coupé en deux parties :
 - ❶ les sommets **marqués** et
 - ❷ les sommets **pas marqués**,
 - ❸ au début : tous les sommets sont pas marqués.
 - ❷ Une valeur $\ell(v)$ est associée à chaque sommet v ,
 - ❶ la distance de s à v si v est marqué,
 - ❷ le coût d'un plus court (s, v) -chemin qui n'utilise que des sommets marqués et v si v est pas marqué,
 - ❸ au début : $\ell(s) = 0$ et $\ell(v) = \infty$ sinon.
- ❷ Le sommet suivant **v** qui sera marqué est choisi comme
 - ❶ un sommet pas marqué
 - ❷ dont la valeur est finie et minimale,
- ❸ on fait une mise à jour : pour chaque arc vu où u est pas marqué :
 $\ell(u) := \min\{\ell(u), \ell(v) + c(vu)\}.$

Algorithme de Dijkstra :

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

SORTIE : ensemble S de sommets atteignables depuis s ,
fonction $\ell : V \rightarrow \mathbb{R}$ et s -arborescence F sur S tq $\forall v \in V$:
le (s, v) -chemin unique dans F est de coût $\ell(v) = \text{dist}_{(G, c)}(s, v)$.

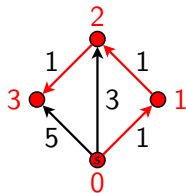
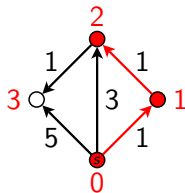
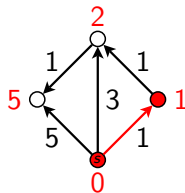
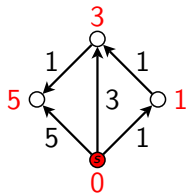
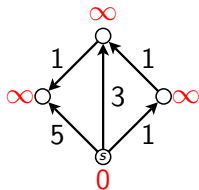
Etape 0: $\ell(s) := 0, \ell(v) := \infty, a(v) := \emptyset \forall v \in V, S := \emptyset, F := \emptyset$.

Etape 1: Tant qu'il existe $u \in V \setminus S$ tel que $\ell(u)$ soit fini faire
Soit $v \in V \setminus S$ tel que $\ell(v) = \min\{\ell(u) : u \in V \setminus S\}$.
 $S := S \cup v$,
 $F := F \cup a(v)$.

$\forall vu \in A$ tel que $u \in V \setminus S$ et $\ell(u) > \ell(v) + c(vu)$:
 $\ell(u) := \ell(v) + c(vu)$ et $a(u) := vu$.

Etape 2: STOP.

Exécution de l'algorithme de Dijkstra



Algorithme de Dijkstra

Justification

- ① On montre que quand on marque le sommet v , $\ell(v) = \text{dist}_{(G,c)}(s, v)$.
- ② Soit P un plus court (s, v) -chemin, $c(P) = \text{dist}_{(G,c)}(s, v)$.
- ③ Soient zu le premier arc de P qui sort de S .
- ④ $c(P) \leq \ell(v)$ car $\ell(v)$ est le coût d'un (s, v) -chemin,
- ⑤ $\ell(v) \leq \ell(u)$ par le choix de v ,
- ⑥ $\ell(u) \leq \ell(z) + c(zu)$ par la mise à jour en z ,
- ⑦ $\ell(z) + c(zu) = \text{dist}(s, z) + c(zu)$ par récurrence,
- ⑧ $\text{dist}(s, z) + c(zu) = c(P[s, z]) + c(zu)$ par optimalité du sous-chemin,
- ⑨ $c(P[s, z]) + c(zu) = c(P[s, u])$ par définition,
- ⑩ $c(P[s, u]) \leq c(P)$ par non-négativité de c .
- ⑪ On a égalité partout, en particulier, $\ell(v) = c(P) = \text{dist}_{(G,c)}(s, v)$.

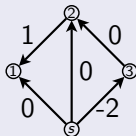


Pas de circuit absorbant : Programmation dynamique

L'idée de l'algorithme de Bellman-Ford

- 1 $w_k(v) :=$ coût minimum d'un (s, v) -chemin ayant au plus k arcs,
- 2 au début : $w_0(s) = 0$ et $w_0(v) = \infty$ pour tout sommet $v \in V \setminus s$.
- 3 Par récurrence (d'après l'optimalité des sous-chemins) :
 $w_{k+1}(v) = \min\{w_k(v), w_k(u) + c(uv) : uv \in A\},$
- 4 à la fin : $w_n(v) = \text{dist}_{(G,c)}(s, v)$ pour tout sommet v .

Exemple



| | 0 | 1 | 2 | 3 | 4 |
|---|----------|----|----|----|----|
| s | 0 | 0 | 0 | 0 | 0 |
| 1 | ∞ | 0 | 0 | -1 | -1 |
| 2 | ∞ | 0 | -2 | -2 | -2 |
| 3 | ∞ | -2 | -2 | -2 | -2 |

Algorithme de Bellman-Ford

ENTRÉE: Un réseau $(G = (V, A), c)$ et une racine s de G .

SORTIE : Une fonction $\ell : V \rightarrow \mathbb{R}$ telle que $\ell(v) = {}_{(G, \ell)}(s, v) \forall v \in G$,
ou bien " G contient un circuit absorbant".

Etape 0: *Initialisation.*

$$\ell_0(s) := 0, \ell_0(v) := \infty \forall v \in V \setminus \{s\}, i := 0.$$

Etape 1: *Mise à jour.*

$$\ell_{i+1}(v) := \min\{\ell_i(v), \ell_i(u) + c(uv) : uv \in A\} \forall v \in V.$$

Etape 2: *Test de fin d'algorithme.*

Si $\ell_{i+1} \neq \ell_i$ et $i + 1 < |V|$ alors $i := i + 1$ et aller à l'Etape 1,
si $\ell_{i+1} = \ell_i$ alors $\ell := \ell_i$ et STOP,
si $i + 1 = |V|$ alors afficher " G contient un circuit absorbant" et
STOP.

Résultats

- 1 Caractérisation des réseaux sans circuit absorbant.
- 2 Théorème min-max sur la distance.
- 3 Résultat structurel : arborescences des plus courts chemins.

Applications

- 1 Ordonnancement,
- 2 Sac à dos,
- 3 Transport maritime,
- 4 Système d'inéquations formé de différences.