Algorithmes de plus courts chemins

Dans tous les algorithmes décrits ci-dessous, n représente le nombre de sommets et m le nombre d'arcs du graphe considéré. On a donc $n=|V|$ et $m=|A|$.

On notera $N^+(x)$ l'ensemble des successeurs immédiats de x et $N^-(x)$ l'ensemble de ses prédécesseurs immédiats. De plus, on notera $d_G^+(x)$ le nombre de sommets dans $N^+(x)$, et $d_G^-(x)$ le nombre de sommets dans $N^-(x)$.

A) RECHERCHE DES PLUS COURTS CHEMINS DU SOMMET 1 AUX AUTRES SOMMETS DU GRAPHE

Algorithme de Dijkstra
Hypothèse : tous les arcs ont des longueurs non négatives

(1) $\bar{S} :=\{2,\ldots,n\}$; $\pi(1) :=0$; pour tout $x\neq 1$ faire $\pi(x) := \begin{cases} d_{1x} & \text{si } x \in N^+(1) \\ \infty & \text{sinon} \end{cases}$

(2) Déterminer x tel que $\pi(x)\leq\pi(y)$ pour tout y dans $\bar{S}$ et poser $\bar{S} := \bar{S} - \{x\}$
Si $\bar{S} =\varnothing$ alors STOP

(3) Pour tout y dans $\bar{S} \cap N^+(x)$ faire $\pi(y):=\min\{\pi(y),\pi(x)+d_{xy}\}$ et retourner à (2)

Complexité : $O(n^2)$
(des structures de données adaptées permettent d'obtenir une complexité $O(m +n \log(n))$)

Remarque : Pour retrouver les plus courts chemins, il suffit de mémoriser le prédécesseur x de y à chaque fois que $\pi(y)$ est modifié.

Algorithme de Dijkstra adapté dans le cas où toutes les longueurs sont unitaires

(1) $S := \varnothing$; $\pi(1) :=0$; pour tout $x\neq 1$ faire $\pi(x) := \infty$; $k :=0$;

(2) Déterminer $S_k=\{x$ tel que tel que $\pi(x)=k\}$; $S:=S\cup S_k$
Si S=V STOP

(3) $k :=k+1$; pour tout arc (x,y) tel que $x\in S$ et $y\notin S$ poser $\pi(y) :=k$; retourner à (2)

Complexité : $O(m)$

Algorithme de Moore
Hypothèse : les longueurs peuvent être négatives, mais il n'existe pas de circuit de longueur négative

(1) $\bar{S} :=\{2,\ldots,n\}$; $\pi(1) :=0$; pour tout $x\neq 1$ faire $\pi(x) := \begin{cases} d_{1x} & \text{si } x \in N^+(1) \\ \infty & \text{sinon} \end{cases}$

(2) Déterminer x tel que $\pi(x)\leq\pi(y)$ pour tout y dans $\bar{S}$ et poser $\bar{S} := \bar{S} - \{x\}$;

(3) Pour tout y dans $N^+(x)$ faire
$\pi^*:=\min\{\pi(y),\pi(x)+d_{xy}\}$
si $\pi^*<\pi(y)$ alors poser $\pi(y):=\pi^*$ et rajouter y dans $\bar{S}$ s'il ne s'y trouve pas déjà.
Si $\bar{S} =\varnothing$ alors STOP sinon aller à (2)

Remarque : l'algorithme n'est pas très efficace, mais est une extension naturelle de l'algorithme de Dijkstra au cas où des longueurs peuvent être négatives.

Algorithme de Bellman

Hypothèse : aucune (c'est-à-dire que les longueurs peuvent être négatives, et il peut y avoir des circuits de longueur négative)

(1) $\pi^0(1):=0$; $\pi^0(x):=\infty$ pour tout $x\neq 1$; $k:=1$;

(2) $\pi^k(1):=0$; $\pi^k(x):=\min\{\pi^{k-1}(x), \min\limits_{y\in N^-(x)} \{\pi^{k-1}(y)+d_{yx}\}\}$

(3) Si $\pi^k(x)=\pi^{k-1}(x)$ pour tout x alors STOP
Si $k\leq n-1$ poser $k:=k+1$ et aller à (2)
Si $k=n$ STOP : il existe un circuit de longueur négative

Complexité : $O(mn)$

Algorithme de Ford

Hypothèse : aucune sur les longueurs des arcs (c'est-à-dire que les longueurs peuvent être négatives)
Par contre, on sait qu'il n'existe pas de circuits de longueur négative

(1) $\pi(1):=0$; $\pi(x):=\infty$ pour tout $x\neq 1$;

(2) Pour x=2 à n faire $\pi(x):=\min\{\pi(x), \min\limits_{y\in N^-(x)} \{\pi(y)+d_{yx}\}\}$

(3) Si $\pi(x)$ a été modifié pour au moins un sommet x, alors retourner à (2), sinon STOP.

Remarque : On a de la flexibilité sur la numérotation des sommets considérée à l'étape 2.

Algorithme dans le cas où on sait que le graphe n'a pas de circuit

Hypothèse : aucune sur les longueurs des arcs (c'est-à-dire que les longueurs peuvent être négatives).
Par contre on sait que le graphe n'a pas de circuit.

(a) *Calcul du rang des sommets*

(a.1) Pour tout x poser $d(x) := d_G^-(x)$; $k:=1$; $S:=\emptyset$

(a.2) Soit R^k l'ensemble des sommets $x\notin S$ tel que $d(x)=0$. Poser $S:=S\cup R^k$.
Pour tout x dans R^k et pour tout $y\in N^+(x)$ faire $d(y):=d(y)-1$;

(a.3) Si S=V alors poser $rang_{max}:=k$ et STOP; sinon poser $k:=k+1$ et retourner à (a.2)

(b) $\pi(1):=0$;
Pour k=2 à $rang_{max}$ faire
Pour tout x dans R^k faire $\pi(x):= \min\limits_{y\in N^-(x)} \{\pi(y)+d_{yx}\}$

Complexité : $O(m)$

B) RECHERCHE DES PLUS COURTS CHEMINS ENTRE TOUTE PAIRE DE SOMMETS DU GRAPHE

Algorithme de Floyd

Hypothèse : aucune (c'est-à-dire que les longueurs peuvent être négatives, et il peut y avoir des circuits de longueur négative)

(1) Pour tout i et j tel que $1\leq i,j\leq n$ faire $L_{ij} := \begin{cases} 0 & \text{si } i = j \\ d_{ij} & \text{si } (i, j)\in A \\ \infty & \text{sinon} \end{cases}$

(2) Pour k allant de 1 à n faire
Pour tout i allant de 1 à n faire
Pour tout j allant de 1 à n faire $L_{ij}:=\min\{L_{ij},L_{ik}+L_{kj}\}$
Si $L_{ii}<0$ alors il existe un circuit de longueur négative. STOP

Complexité : $O(n^3)$