Tutorial 2 : Shortest path in graphs

Operations research, 3rd semester.

2022

Exercice 1 — *Routing table in a network*

In a packet switching network, we often need a table indicating, for each node, where an entering packet must be send in order to reach its destination, whatever its source is. This exercice explains how to build such a table.

Let be a network with 4 nodes, describe by the following adjacency matrix of a graph :

	1	2	3	4
1	-	3	-	3
2	2	-	2	2
3	1	-	-	1
4	-	4	4	-

1. Give two algorithms that can be used to compute the shortest path from the node **1** to every other node.
2. In fact, instead of one source, we need the paths from every possible source to every possible destination. Apply the Floyd-Warshall algorithm to do so. Give the matrices π and P at each step of the algorithm.
3. Using P , give the optimal routing table T in that network, *i.e.* for each couple (u, v) , give the successor of u in a shortest path from u to v .

Algorithm 1 Floyd-warshall ($G = (X, U)$)

$\pi_{ij}^{(0)} :=$ weight of (i, j)

// if the arc (i, j) does not exist, the weight is $+\infty$; unless $i = j$ where the weight is 0

$P^{(0)}(i, j) := i$ // At the end of the algorithm, $P^{(n)}(i, j)$ contains the predecessor of j in a shortest path from i to j

for $m = 1, 2, \dots, n$ **do**

for $i = 1, 2, \dots, n$ **do**

for $j = 1, 2, \dots, n$ **do**

 // Is it shorter to go from i to j by going through m ?

if $(\pi_{im}^{(m-1)} + \pi_{mj}^{(m-1)} < \pi_{ij}^{(m-1)})$ **then**

$\pi_{ij}^{(m)} := \pi_{im}^{(m-1)} + \pi_{mj}^{(m-1)}$

$P_{ij}^{(m)} := P_{mj}^{(m-1)}$

else

$\pi_{ij}^{(m)} := \pi_{ij}^{(m-1)}$

$P_{ij}^{(m)} := P_{ij}^{(m-1)}$

► **Correction**

1. Dijkstra et Bellman-Ford

2.

$$\pi^{(0)} = \begin{pmatrix} 0 & 3 & +\infty & 3 \\ 2 & 0 & 2 & 2 \\ 1 & +\infty & 0 & 1 \\ +\infty & 4 & 4 & 0 \end{pmatrix}, P^{(0)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

$$\pi^{(1)} = \begin{pmatrix} 0 & 3 & +\infty & 3 \\ 2 & 0 & 2 & 2 \\ 1 & \boxed{4} & 0 & 1 \\ +\infty & 4 & 4 & 0 \end{pmatrix}, P^{(1)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & \boxed{1} & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

$$\pi^{(2)} = \begin{pmatrix} 0 & 3 & \boxed{5} & 3 \\ 2 & 0 & 2 & 2 \\ 1 & 4 & 0 & 1 \\ \boxed{6} & 4 & 4 & 0 \end{pmatrix}, P^{(2)} = \begin{pmatrix} 1 & 1 & \boxed{2} & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 3 \\ \boxed{2} & 4 & 4 & 4 \end{pmatrix}$$

$$\pi^{(3)} = \begin{pmatrix} 0 & 3 & 5 & 3 \\ 2 & 0 & 2 & 2 \\ 1 & 4 & 0 & 1 \\ \boxed{5} & 4 & 4 & 0 \end{pmatrix}, P^{(3)} = \begin{pmatrix} 1 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 3 \\ \boxed{3} & 4 & 4 & 4 \end{pmatrix}$$

3. Il faut remonter P pour construire le chemin. Pour chaque couple (u, v) , on connaît le dernier nœud w du chemin de u à v , on veut le premier. Cela revient à chercher le premier nœud w' de u à w , etc. Pour arrêter la récurrence, on utilise la propriété suivante : si u est le prédécesseur de v dans un chemin de u à v alors v est le successeur de u dans ce même chemin.

Exercice 2 — Optimality condition

1. Let $G = (X, U)$ be a directed graph with a node s , and where each arc (u, v) is weighted with $\omega(u, v) \in \mathbb{R}$. We associate each node u with a number d_u . Demonstrate the following theorem : The graph has no absorbing circuit if and only if the system of inequations $\{ d_v - d_u \leq \omega(u, v) \text{ for every arc } (u, v) \in U \}$ has a solution. In that case, setting each number $d_u \in \mathbb{R}$ to the value of a shortest path from s to u is a solution.
2. Transform the following system into a shortest path problem in a graph and solve it :

$$\begin{aligned}x_3 - x_4 &\leq 5 \\x_4 - x_1 &\leq -10 \\x_1 - x_3 &\leq 8 \\x_2 - x_1 &\leq -11 \\x_3 - x_2 &\leq 2\end{aligned}$$

3. Same question for this system :

$$\begin{aligned}x_3 - x_2 &\leq 5 \\x_4 - x_3 &\leq -2 \\x_4 - x_2 &\leq 4 \\x_1 - x_2 &\leq 3 \\x_5 - x_1 &\leq 2 \\x_3 - x_5 &\leq 7 \\x_5 - x_4 &\leq -1\end{aligned}$$

► Correction

1. Supposons qu'il n'y ait pas de circuit absorbant, alors il existe un plus court chemin de valeur d_u entre s et u pour tout $u \in X$ alors, $d_v \leq d_u + \omega(u, v)$ car un chemin empruntant un plus court chemin de s à u puis l'arc (u, v) est un chemin de s à v .
Sinon, s'il existe un circuit absorbant $(u_1, u_2, \dots, u_p, u_{p+1} = u_1)$, alors, si le système a une solution, on vérifie $0 = \sum_{i=1}^p d_{u_{i+1}} - d_{u_i} \leq \sum_{i=1}^p \omega(u_i, u_{i+1}) < 0$ ce qui est exclu.
2. Si la question 1 est vraie, on construit un graphe avec 4 nœuds où $\omega(v_3, v_4) = 5$, $\omega(v_4, v_1) = 10$, $\omega(v_3, v_1) = 8$, $\omega(v_1, v_2) = -11$, $\omega(v_2, v_3) = 2$. On cherche un plus court chemin de n'importe quel nœud s vers les autres, ce qui nous donne x_i .
3. Comme pour la question précédente.

Dans la question 2 ou 3 on trouve un circuit absorbant, et donc pas de solution au problème.

Exercice 3 — Disjoint cuts and shortest paths

Let $G = (V, A)$ be a directed graph and x and y be two nodes of G . A cut separating x from y is a subset of arcs of A such that removing those arcs from G destroy every directed path from x to y .

1. Show that, after removing every arc of a cut C separating x from y , V can be partitioned into two sets X and $Y = V \setminus X$ such that every arc of G linking X to Y is in C .
2. Show that the maximum number of disjoint cut separating x from y equals the number of arcs in a shortest path from x to y .

► **Correction**

1. X est l'ensemble des descendants de x dans le graphe résultant de la suppression des arcs.
2. Supposons que l'on ait un chemin P avec k arcs entre x et y . On note $P = \{a_1, a_2, \dots, a_k\}$. Soit $C = (C_1, C_2, \dots, C_l)$ des coupes disjointes. Pour tout C_j , il existe un arc a_i de P dans C_j (sinon la coupe ne sépare pas x de y). Puisque deux coupes C_i et C_j sont disjointes, il ne peut y avoir plus de k coupes dans C : $l \leq k$. Donc le nombre maximum de coupes disjointes dans le graphe est plus petit que la taille d'un plus court chemin.

Montrons qu'il existe un ensemble de k coupes disjointes. Construisons récursivement l'ensemble suivant :

- $V_0 = x$
- $V_{i+1} = \Gamma^+(\bigcup_{j=0}^i V_j)$

Soit l le premier (et unique) entier tel que $y \in V_l$. Pour $i < l$, posons C_i l'ensemble des arcs allant d'un nœud de $\bigcup_{j=0}^i V_j$ à un nœud de V_{i+1} . Par construction, chaque couple (C_i, C_j) est disjoint si $i \neq j$. Chaque ensemble C_i est une coupe séparant x de y car

- $\bigcup_{j=0}^i V_j$ contient x mais pas y (car $i < l$)
- soit un chemin Q reliant x à y dans G , et soit u le dernier nœud de $\bigcup_{j=0}^i V_j$ dans Q .

Ce nœud ne peut être y car y n'est pas dans $\bigcup_{j=0}^i V_j$. Soit v le nœud qui suit u dans Q .

Nécessairement (u, v) est un arc de G et v n'est pas dans $\bigcup_{j=0}^i V_j$. Par définition de V_{i+1} , $v \in V_{i+1}$ et $(u, v) \in C_i$. Donc C_i coupe le chemin Q .

Montrons enfin que l est égal à la taille de P^* , un plus court chemin entre x et y . On note $\{u_0 = s, u_1, \dots, u_{k-1}, u_k = t\}$ les nœuds de P^* .

Propriété 1. Pour tout nœud v de V_{i+1} , il existe un nœud u de V_i tel que $(u, v) \in A$.

La propriété 1 permet de montrer par récurrence que si un nœud v est dans V_i alors il existe un chemin de taille i entre x et v .

Montrons par récurrence que $u_i \in V_i$.

- $u_0 = s \in V_0$ par définition
- Soit $i \leq l, k$. Si, pour tout $j \leq i, u_j \in V_j$. Le nœud $u_{i+1} \notin \bigcup_{j=0}^i V_j$ car alors il existerait un chemin de x à u_{i+1} de taille inférieure à $i + 1$ et donc un chemin de x à y de taille inférieure à k . Donc $u_{i+1} \in V_{i+1}$.

Donc $u_k \in V_l$, donc $k = l$. Donc il existe un ensemble de k coupes disjointes.

Exercice 4 — Currency converter

Let G be a graph where each node is a currency and where an arc (u, v) exists if and only if it is possible to convert the currency u into the currency v . In that case, the arc is weighted with the exchange rate.

What condition must this graph satisfy so that someone can become as rich as he wants? Transform this graph into a weighted graph H such that searching for a shortest path in H is equivalent to deciding if someone can become as rich as he wants. Is the Dijkstra algorithm useful to answer this question?

► **Correction**

On met un nœud par monnaie, et comme poids $-\log(c(u, v))$ où $c(u, v)$ est la conversion de u vers v . On obtient donc un problème de plus court chemin (plutôt qu'un problème de recherche d'un chemin dont le produit des poids est maximum). Il y a un circuit absorbant ssi on peut devenir infiniment riche.