

# TP 2 Recherche Opérationnelle

## Licence III Informatique

### Etude et implémentation d'algorithmes de plus courts chemins

Serigne A. Gueye

3 février 2019

Soit un 1-graphe  $G = (V, E)$ , connexe, où  $V = \{1, 2, \dots, n\}$  et dont chaque arc  $(i, j)$  est valué par un coût (ou longueur)  $c_{ij} \in \mathbb{R}$ .

$G$  sera représentée par une matrice  $c \in \mathbb{R}^+ \times \mathbb{R}^+$  dont chaque élément  $c_{ij} = \infty$  si  $(i, j)$  n'existe pas.

## 1 Méthode Moore-Dijkstra

On considère dans cette section l'algorithme de Moore-Dijkstra vu en cours.

### 1.1 Implémentation

1- Ecrire une méthode :

MooreDijkstra(s)

calculant les plus courts chemins d'un sommet donné  $s$  à tous les autres sommets avec cette méthode.

- La méthode renverra le tableau “pere”. Pour tout sommet  $i$ ,  $\text{pere}[i]$  contient l’indice du prédécesseur de  $i$  dans le plus court chemin de  $s$  à  $i$ .

- Le vecteur  $d$  donnant pour chaque indice  $i$  la valeur du plus court chemin de  $s$  à  $i$  sera dans un premier temps implémenté avec un tableau.

2- Les plus courts chemins devront être affichés par une méthode

`display_shortest_paths(pere)`

affichant à l’écran les plus courts chemins calculés.

## 1.2 Amélioration de l’algorithme

Pour représenter  $d$ , on utilise maintenant une structure de tas (heap)<sup>1</sup> au lieu d’un tableau.

4- Montrer que la complexité de l’algorithme de Dijkstra en terme de nombre de comparaisons devient  $O(m \log(n))$ .

5- Changer la structure de données utilisée pour  $d$  par un tas.

## 2 Méthode de Moore, 1957

On considère l’algorithme suivant.

---

1. voir notes de cours sur le site (section 7)

```

1 Algorithm: Moore(s)
2  $C = \{s\}$  ;
3  $\overline{C} = \{1, 2, \dots, n\} \setminus \{s\}$  ;
4 pour  $i \in \overline{C}$  faire
5   | si  $i \in \Gamma_s$  alors
6   |   |  $d(i) = c_{si}$ 
7   | fin
8   | sinon
9   |   |  $d(i) = +\infty$ 
10  | fin
11 fin
12 tant que  $\overline{C} \neq \emptyset$  faire
13   |  $d(j) = \min_{i \in \overline{C}} d(i)$  ;
14   |  $\overline{C} = \overline{C} \setminus \{j\}$ ;
15   | pour  $i \in \Gamma(j)$  faire
16   |   | si  $d(j) + c_{ji} < d(i)$  alors
17   |   |   |  $d(i) = d(j) + c_{ji}$ ;
18   |   |   |  $\overline{C} = \overline{C} \cup \{i\}$ ;
19   |   | fin
20   | fin
21 fin

```

4 - Cet algorithme converge-t-il quand le graphe est sans circuit de longueur négative.<sup>2</sup>.

5- Que se passe-t-il quand il est appliqué sur un graphe contenant un circuit de longueur négative ?

6- Si l'on remplace le sommet  $j$  choisi à la ligne 13 par un sommet quelconque de  $\overline{C}$  cela change-t-il quelque chose à la convergence ?

7- Ajouter, comme pour l'algorithme de Moore-Dijkstra, une structure "pere" et des instructions permettant de stocker les plus courts chemins.

---

2. voir preuve analogue transparent 75 du cours

8- Si toutes les longueurs  $c_{ij}$  des arcs sont positives à quel algorithme est-il équivalent ?

### 3 Méthode de Bellman, 1958

9- Ecrire une méthode :

Bellman(s)

calculant les plus courts chemins d'un sommet donné  $s$  à tous les autres sommets avec cette méthode<sup>3</sup>.

---

3. La méthode à implémenter est celle vue en cours