Dossier Optimisation - Recherche Opérationnelle

M1 Informatique - 2020/2021

Responsable : Julien Ah-Pine

1 Objectif du dossier

L'objectif de ce dossier est de vous évaluer sur votre capacité à comprendre des **algorithmes traitant des graphes** et à les implémenter dans un langage de script et en particulier le **langage R**. Les compétences acquises en TP sont mises à profit. L'évaluation prend donc en compte votre implication lors de ces séances de travail.

2 Aspects liés à l'organisation et à la remise du dossier

Les dossiers s'effectuent par **binôme**. Il est attendu des étudiants qu'ils fournissent :

1. Un fichier .R dans lequel se trouveront toutes les fonctions demandées.
2. Un rapport au format PDF accompagnant le dossier.

Vous devrez créer une archive .zip contenant ces deux fichiers. Vous nommerez votre archive de la manière suivante NOM1_NOM2.zip où NOMi sont les noms des membres du binôme. **Vous devrez envoyer par email cette archive au plus tard le 20 Décembre 2020 23h59 précise à l'adresse mail : julien.ah-pine@univ-lyon2.fr**. **ATTENTION** : vous êtes responsables de votre envoi et donc toute absence de ressource ou tout problème conduisant à l'impossibilité d'accéder correctement à votre travail est de votre responsabilité.

3 Descriptif du rapport accompagnant votre dossier

Le rapport que vous allez rédiger est une partie importante de votre dossier. C'est également à travers ce document que j'apprécierai votre investissent dans ce travail. C'est pour cette raison qu'une partie importante de votre note sera attribuée vis à vis de la qualité de ce document. **Le rapport devra être fourni au format PDF**. **ATTENTION** : les fichiers Word (.doc, .rtf ou .odt) ne seront pas acceptés.

Le rapport doit obligatoirement contenir :

— Une introduction présentant l'objet et le contenu du dossier.

— Une section pour chaque algorithme présentant (i) l'objet de l'algorithme du point de vue conceptuel ainsi que son intérêt pratique ; (ii) le pseudo-code de l'algorithme ; (iii) le code R que vous avez écrit ; (iv) une illustration du bon fonctionnement de votre implémentation sur un ou plusieurs exemples.

— Une conclusion présentant une synthèse ainsi qu'une analyse critique du travail fourni.

Pour vous aider dans la rédaction de votre rapport, vous trouverez en pj un exemple d'organisation dont vous pourrez vous inspirez.

4 Descriptif du travail d'implémentation

Vous devez implémenter en R les 3 algorithmes suivants. Chaque algorithme devra faire l'objet d'une fonction.

4.1 Algorithme 1 : détection de l'arbre recouvrant de poids minimal par Prim

L'agorithme de Prim est une méthode distincte de celle de Kruskal permettant de déterminer un arbre partiel de poids minimal.

Q1 Ecrivez une fonction `Prim` qui prend en entrée un ensemble de sommets `X`, une matrice d'adjacence pondérée `A` et qui donne en sortie, l'arbre partiel de poids minimal obtenu par l'algorithme de Prim dont le pseudo-code est donné ci-dessous :

Input : $G = [X, U]$
$X' \leftarrow \{i\}$ où i est un sommet de X pris au hasard
$U' \leftarrow \emptyset$
Tant que $X' \neq X$ **faire**
    Choisir une arête (j,k) de poids minimal tel que $j \in X'$ et $k \notin X'$
    $X' \leftarrow X' \cup \{k\}$
    $U' \leftarrow U' \cup \{(j,k)\}$
Fin Tant que
Output : $G' = [X', U']$

Q2 Testez votre fonction sur le graphe représenté par la matrice d'adjacence pondérée suivante :

$$\mathbf{A} = \begin{pmatrix} 0 & 5 & 8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 4 & 2 & 0 & 0 \\ 8 & 0 & 0 & 0 & 5 & 2 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 7 \\ 0 & 2 & 5 & 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 7 & 3 & 3 & 0 \end{pmatrix}$$

4.2 Algorithme 2 : calcul des plus courts chemins par Ford-Bellman

Q1 Ecrivez une fonction `Ford_Bellman` qui prend en entrée un ensemble de sommets `X`, une matrice d'adjacence pondérée `A`, un sommet `s` de `X` et qui donne en sortie, les longueurs des plus courts chemins entre `s` et tous les autres sommets de `X` obtenus par l'algorithme de Ford et Bellman que nous rappelons ci-dessous :

Input : $G = [X, U], s$
$\pi(s) \leftarrow 0$
Pour tout $i \in \{1, 2, \ldots, N\} \setminus \{s\}$ **faire**
    $\pi(i) \leftarrow +\infty$
Fin Pour
Répéter
    Pour tout $i \in \{1, 2, \ldots, N\} \setminus \{s\}$ **faire**
        $\pi(i) \leftarrow \min(\pi(i), \min_{j \in \Gamma^{-1}(i)} \pi(j) + l_{ji})$;
    Fin Pour
Tant que une des valeurs $\pi(i)$ change dans la boucle Pour
Output : π

Q2 Testez votre fonction sur le graphe représenté par la matrice d'adjacence pondérée suivante. Vous prendrez au hasard un sommet de départ.

$$\mathbf{A} = \begin{pmatrix} 0 & 5 & 8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 4 & -2 & 0 & 0 \\ 8 & 0 & 0 & 0 & 5 & 2 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 7 \\ 0 & -2 & 5 & 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 7 & 3 & -3 & 0 \end{pmatrix}$$

Expliquez en quoi vous êtes sur un cas particulier.

Q2 Testez votre fonction sur le graphe représenté par la matrice d'adjacence pondérée suivante. Vous prendrez 7 comme sommet de départ.

$$\mathbf{A} = \begin{pmatrix} 0 & 5 & 8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 4 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 5 & 2 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 & 3 & -3 & 0 \end{pmatrix}$$

4.3 Algorithme 3 : détermination d'un flot maximal dans un réseau avec capacités par Ford-Fulkerson

Q1 Ecrivez une fonction `Ford_Fulkerson` qui prend en entrée un ensemble de sommets `X`, un réseau avec capacités (de poids non négatifs) `A` (sa matrice d'adjacence), un sommet source `s` et un sommet puits `p` de `X` et qui donne en sortie, le flot `P` de valeur maximale obtenus par l'algorithme de Ford-Fulkerson que nous rappelons ci-dessous :

Input : $G = [X, U, C]$, φ un flot réalisable
1 $m_s \leftarrow (\infty, +)$ et $S = \{s\}$
2 **Tant que** $\exists (j \in \overline{S}, i \in S) : (c_{ij} - \varphi_{ij} > 0) \vee (\varphi_{ji} > 0)$ **faire**
3   **Si** $c_{ij} - \varphi_{ij} > 0$ **faire**
4         $m_j \leftarrow (i, \alpha_j, +)$ avec $\alpha_j = \min\{\alpha_i, c_{ij} - \varphi_{ij}\}$
5   **Sinon Si** $\varphi_{ji} > 0$ **faire**
6         $m_j \leftarrow (i, \alpha_j, -)$ avec $\alpha_j = \min\{\alpha_i, \varphi_{ji}\}$
7   **Fin Si**
8     $S \leftarrow S \cup \{j\}$
9   **Si** $j = p$ **faire**
10         $V(\varphi) \leftarrow V(\varphi) + \alpha_p$
11   Aller en 14
12   **Fin Si**
13 **Fin Tant que**
14 **Si** $p \in S$ **faire**
15   **Tant que** $j \neq s$ **faire**
16   **Si** $m_j(3) = +$ **faire**
17             $\varphi_{m_j(1)j} \leftarrow \varphi_{m_j(1)j} + \alpha_p$
18   **Sinon Si** $m_j(3) = -$ **faire**
19             $\varphi_{jm_j(1)} \leftarrow \varphi_{jm_j(1)} - \alpha_p$
20   **Fin Si**
21         $j \leftarrow m_j(1)$
22   **Fin Tant que**
23   Aller en 1
24 **Sinon faire**
25   **Output :** φ
26 **Fin Si**

Q2 Testez votre fonction sur l'exemple représenté dans la Figure 1 où les sommets 4 et 2 sont respectivement la source et le puit du réseau.

5 Aspects liés à l'évaluation

Pour l'évaluation de votre dossier je tiendrai compte des points suivants :

— la bonne implémentation et le bon fonctionnement des fonctions codées (est-ce qu'elles mettent en oeuvre correctement le calcul attendu ?),

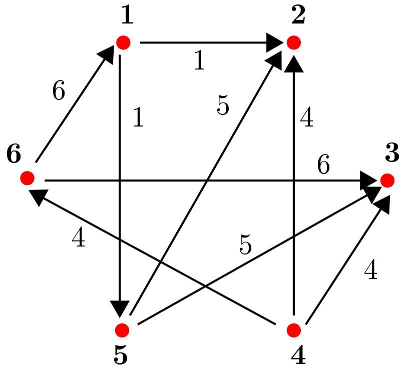

Figure 1 – Réseau avc capacités.

— la qualité des commentaires dans le code (idéalement une personne tierce qui doit comprendre et/ou reprendre le code que vous avez écrit doit pouvoir trouver dans les commentaires les informations utiles),
— la qualité de la structure et de la rédaction du rapport accompagnant le dossier.