TD 10 : Problèmes de recherche

Jeudi 6 décembre 2018

1 Recherche locale d'un arbre couvrant de poids minimal (ACM)

Soit $G = (S, A)$ graphe non orienté connexe et $T \subseteq A$ un arbre couvrant de G. On rappelle que si on ajoute un nouvel arc $a \in A \setminus T$ à T, on crée un cycle. On peut alors retirer un arc $a' \in T$ de ce cycle pour obtenir un nouvel arbre couvrant $T' = T \cup \{a\} \setminus \{a'\}$. On appelle $\text{ÉCHANGE}(a, a')$ l'opération qui permet d'obtenir T' à partir de T ; T' ainsi obtenu est appelé *voisin* de T.

QUESTION 1 – Montrer qu'en partant d'un arbre couvrant T, on peut obtenir n'importe quel arbre couvrant T' par une série d'échanges. Combien d'échanges successifs sont nécessaires au maximum ?

QUESTION 2 – Montrer que si T' est un ACM, on peut choisir ces échanges de telle façon que le poids des arbres couvrants soit décroissant (au sens large) dans la séquence, c'est-à-dire que si $T = T_0 \to T_1 \to \dots T_k = T'$ (où $\to$ représente l'opération ÉCHANGE), alors

$$\forall i < k, \qquad \text{POIDS}(T_{i+1}) \leq \text{POIDS}(T_i).$$

QUESTION 3 – On suppose maintenant que tous les poids sont distincts et on propose l'algorithme de recherche locale suivant :

Fonction ACM $\{T$ est un arbre couvrant quelconque$\}$
  tant que il existe un échange (a, a') qui permette de réduire le poids **faire**
    $T \leftarrow T \cup \{a\} \setminus \{a'\}$
  renvoyer T

Montrer que cet algorithme renvoie toujours un ACM. Quel est le nombre maximal d'itérations ?

2 Arbitrage de devises

Sur l'exemple de la Fig. 1, on considère les taux de change entre 5 devises étrangères.

Le tableau de la Fig. 1 peut être interprété comme un graphe complet orienté pondéré tel que représenté Fig. 2, dont le tableau est la matrice de poids (notée A).

	USD	EUR	GBP	CHF	CAD
USD	1	0.741	0.657	1.061	1.005
EUR	1.349	1	0.888	1.433	1.366
GBP	1.521	1.126	1	1.614	1.538
CHF	0.942	0.698	0.619	1	0.953
CAD	0.995	0.732	0.650	1.049	1

FIGURE 1 – Le tableau de conversion entre 5 devises. La case à ligne i et la colonne j indique le montant obtenu en devise j pour la conversion de 1 unité de la devise i. Par exemple, 1 USD donne droit à 0,741 EUR.

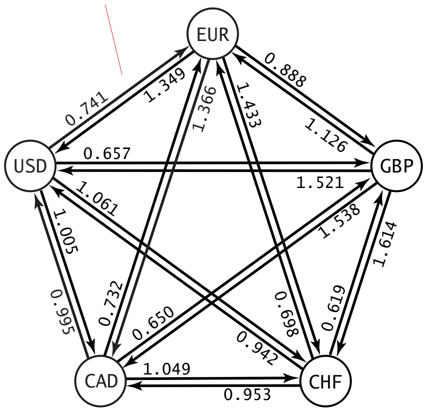

FIGURE 2 – Le graphe de conversion, équivalent au tableau de la Fig. 1.

QUESTION 4 – Étant donné un cycle d'échanges de devises $s = x_0 \to x_1 \to \ldots \to x_k = s$, quel est le taux de change résultant de cet échange ?

On peut s'apercevoir que si l'on dispose d'un capital de départ de 1000 USD, et que l'on fait les échanges successifs suivants : USD $\to$ EUR $\to$ CAD $\to$ USD, on obtiendra à la fin

$$1000 \times 0.741 \times 1.366 \times 0.995 = 1007,14497 \text{ USD}.$$

On appelle une telle transaction *arbitrage* ; étant donné un tableau de conversion, on se pose la question de l'existence d'un arbitrage pour une monnaie donnée, c'est-à-dire d'un cycle d'échanges dont le taux résultant est supérieur à 1.

QUESTION 5 – On définit généralement le poids d'un chemin dans un graphe pondéré comme la somme des poids des arcs qui le composent. Comment modifier la formalisation du graphe pour que le problème se rapproche des algorithmes vus en cours sur les graphes ?

3 Détection de cycle de poids négatif

En cours, on a vu que le parcours en largeur permettait de déterminer les plus courts chemins à partir d'une origine donnée, $s \in S$ dans un graphe orienté non pondéré. L'algorithme de Bellman-Ford ci-dessous permet de résoudre le problème dans le cas plus général du graphe pondéré. Il permet aussi de détecter si le graphe contient un cycle de poids négatif.

```
Fonction BELLMAN-FORD
    pour tout t ∈ S faire
        d[t] ← ∞
    d[s] ← 0
    pour k ← 1 à |S| − 1 faire
        pour tout (u, v) ∈ A faire
            d[v] ← min(d[u] + w(u, v), d[v])
    {Ce dernier bloc de code sert à détecter un cycle négatif}
    pour tout (u, v) ∈ A faire
        si d[v] > d[u] + w(u, v) alors
            renvoyer VRAI
    renvoyer FAUX
```

Question 6 – Montrer que s'il n'existe aucun cycle de poids négatif accessible depuis s, alors à la fin de l'exécution de Bellman-Ford, $\forall t \in S$, $d[t] = \delta(s, t)$.

Question 7 – Montrer que Bellman-Ford renvoie *VRAI* si et seulement si G contient un cycle de poids négatif.

4 Découverte d'un chemin simple de poids minimal

Dans le cas où le graphe contient un cycle de poids négatif accessible depuis s, il n'existe plus de solution au problème du chemin de poids minimal. En effet, on peut boucler de façon infinie sur un cycle de poids négatif pour diminuer arbitrairement le poids d'un chemin. Dans ce cas, on peut se poser la question de trouver le chemin simple de poids minimal.

Question 8 – Montrer que s'il est possible de résoudre le problème du chemin simple de poids minimal dans le cas général, alors il est possible de résoudre le problème du chemin simple de poids maximal.