Théorie des graphes

Chapitre 4 : Parcours de graphes, algorithme de Dijkstra

8 décembre 2020

On considère dans cette partie des graphes pondérés, i.e. à chaque arête/arc est associé un coût positif ou nul. On suppose toujours que le graphe est simple (pas de boucle, pas d'arêtes/arcs multiples).
L'**algorithme de Dijkstra** détermine pour un graphe connexe le plus court chemin d'un sommet choisi à n'importe quel autre sommet du graphe.

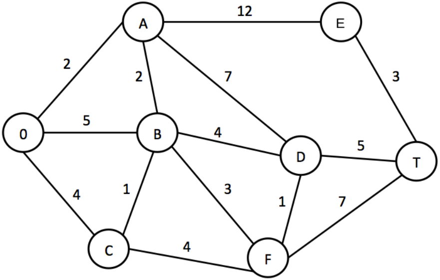
- Cet algorithme peut s'exécuter sur des graphes orientés ou non.
- Il construit un arbre couvrant du graphe, admettant le sommet d'origine comme racine.
- C'est un algorithme en temps polynomial.

- On suppose que le sommet de départ est le sommet 0.
- On prend un vecteur C de taille n, $C(i)$ contient la distance courante entre le sommet 0 et le sommet i.
- Un ensemble S contient les sommets déjà visités, un autre R contient les sommets restants.

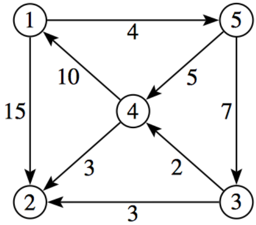
Initialisation
si l'arête $\{v_0, v_i\}$ existe **alors**
    $C(i) = c_{0i}$
sinon
    $C(i) = +\infty$
fin si
$S = \{v_0\}$
$R = \{v_1, \ldots, v_n\}$
$i = 0$
Corps
tant que $R \neq \emptyset$ **faire**
    Choisir i tel que $i = \mathrm{argmin}_{j \in R} C(j)$
    $\{$Mise à jour de $C\}$
    pour Tous les sommets v_j voisins de v_i **faire**
        $C(j) = \min(C(j), C(i) + c_{ij})$
    fin pour
    Ajouter v_i à S
    Retirer v_i à R
fin tant que

- Argument de terminaison : on visite les sommets un à un, donc $\#R$ est strictement décroissant.
- C'est un algorithme de type glouton qui mène à une solution globale optimale.
- Pour la complexité, on parcourt les sommets (passage d'un sommet de R à S et les arêtes sont toutes parcourues –les arcs 1 fois ; les arêtes 2 fois–). A chaque passage sur un sommet, on chercher le minimum dans R. Au pire, la recherche du minimum est linéaire, donc $O(m+n+n^2)=O(m+n^2)$, on peut aussi faire mieux en gardant R trié, mais du coup, il faut pour chaque visite d'arête, insérer dans la liste triée $O((m+n)ln(n))$.

Exercice 4.1.1. Trouver le plus court chemin de 0 à T



Exercice 4.1.2. Trouver les plus courts chemins partant de 1 aux autres sommets du graphe



Remarque 4.1.1. si les poids peuvent être négatifs, il faut utiliser l'algorithme de Bellman-Ford, capable d'identifier un cycle absorbant (cycle tel que $d(v_i, v_i) < 0$).