Arbres couvrants

Master 2 Informatique - UFR S.A.T

Pr. Ousmane THIARE

ousmane.thiare@ugb.edu.sn
http://www.ousmanethiare.com/

16 avril 2020

Chapitre 4 : Arbres couvrants

Dans ce chapitre on s'intéresse au calcul d'arbres couvrants (BFS, DFS, de poids minimum) en mode asynchrone. On suppose donc dans la suite qu'on est en mode asynchrone (model ASYNC).

Un arbre BFS de racine r_0 est un arbre T tel que $d_T(r_0, u) = d_G(r_0, u)$ pour tout sommet u de T. L'arbre T est couvrant s'il passe par tous les sommets de G, c'est-à-dire si V(T) = V(G).
Nous avons vu au chapitre précédant comment construire un arbre couvrant à partir de Flood (corollaire). En mode synchrone cet arbre est précisément un BFS. L'enjeux ici est de le faire en mode asynchrone.

Il s'agit maintenant de visiter tous les sommets d'un graphe en suivant les arêtes. En séquentiel on utilise un parcours en profondeur d'abord (DFS). Le principe est de démarrer depuis un sommet r_0 et pour chaque sommet v faire : si v a des voisins non encore visités, alors on en visite un. Sinon, on revient vers le sommet qui a visité v en premier. Si aucun de ces sommets n'existe, la recherche est terminée. En séquentiel, cela prend un temps $\Theta(|E|)$, tous les voisins de tous les sommets pouvant être testés.

...

En séquentiel, il y a principalement deux algorithmes : PRIM et KRUSKAL. Pour PRIM, on maintient un arbre T de racine r_0 qui grossit en ajoutant à chaque fois l'arête sortante de poids minimum, c'est-à-dire l'arête incidente dont une seule extrémité est dans T et qui est de poids le plus petit possible. L'arbre est initialisé au sommet seul r_0.
Pour KRUSKAL, on trie les arêtes, puis on ajoute à la forêt l'arête la plus petite ne créant pas de cycle. Au départ T est vide.
Prendre un exemple et faire tourner avec PRIM puis KRUSKAL.

Remarque de culture générale

Le problème de MST peut être généralisé de la manière suivante, appelé « arbre de Steiner » : on fixe un ensemble de sommets, disons $U \subseteq V(G)$, et on demande de construire un arbre couvrant U et de poids minimum. Donc MST=Steiner avec U=V(G). Motivation : un groupe d'utilisateurs doit recevoir le même film vidéo et il faut optimiser le coût de diffusion.

Calculer un arbre de Steiner est un problème est NP-complet [Karp 1972] (c'est-à-dire on ne connaît pas encore d'algorithme polynomial en n et on pense qu'il n'y en a pas), et il le reste même si tous les poids sont égaux.