

Exercice d'approximation polynomiale

Recherche d'une coupe séparant C_j des autres terminaux

Soient un graphe non orienté $G = (V, E)$, une fonction de poids positive w sur les arêtes et un ensemble $S = \{s_1, s_2, \dots, s_k\}$ de sommets terminaux de G . Considérons un sommet terminal de G appelé s_j et $G' = (V', E')$ le graphe obtenu par fusion des sommets $s_i \in \{S \setminus s_j\}$, le sommet résultant sera appelé s_r , on remarque alors que $V' = V \setminus \{(uv) \text{ tq } (u, v) \in \{S \setminus s_j\}\}$. Enfin considérons la fonction de poids w' définie comme suit :

$$\begin{cases} V' & \longrightarrow \mathbb{R} \\ w'(v) & = w(v) \end{cases} \quad \forall v \in V'$$

Lemme 0.0.1 *Rechercher une coupe C_j de poids minimum dans G revient alors à chercher une (s_r, s_j) -coupe minimum dans G' .*

Preuve :

Supposons qu'il existe une coupe C_j de poids minimum séparant s_j des autres terminaux dans le graphe G dont le poids est inférieur à celui de la coupe minimum $C_{G'}$ trouvée dans le G' . Par construction, toute arête appartenant à V' est pondérée de la même manière dans G et dans G' , on en déduit que pour que C_j soit différente de $C_{G'}$, alors $C_j \setminus \{V' \cap C_j\} \neq \emptyset$. Ceci est impossible puisque C_j est une coupe séparant s_j de tous les autres terminaux. Donc $C_{G'}$ est une coupe de poids minimum séparant s_j des autres terminaux.

La recherche d'une (s, t) -coupe dans un graphe s'apparente à une recherche de flot maximum dans le même graphe pour laquelle il existe des algorithmes polynomiaux¹. La transformation de G vers G' se faisant en temps linéaire, l'algorithme est un algorithme s'exécutant en temps polynomial.

Calcul des bornes

Somme des poids des arêtes

Considérons une arête (uv) telle que $u \in C_i^*$ et $v \in C_j^*$. Lorsque l'on somme le poids de l'ensemble des coupes extraites de C^* , le poids de l'arête est comptabilisé dans $w(C_i^*)$ et dans $w(C_j^*)$, donc le poids de chaque arête est comptabilisé deux fois. On en déduit alors que :

$$\sum_{i=1}^k w(C_i^*) = 2w(C^*)$$

Minorant de $w(C_m)$

Par définition, C_m est la coupe de poids maximum parmi les j coupes séparatrices de poids minimum. On a donc :

$$\begin{aligned} w(C_i) &\leq w(C_m) & \forall i \in \{1, \dots, k\} \\ \Rightarrow \sum_{i=1}^k w(C_i) &\leq k * w(C_m) \\ \Rightarrow w(C_m) &\geq \frac{1}{k} \sum_{i=1}^k w(C_i) \end{aligned}$$

1. On peut par exemple citer les algorithmes de préflots, Dinic ou Edmonds-Karp.

Une $2(1 - \frac{1}{k})$ -approximation

On sait que :

$$\begin{aligned}
 w(C) + w(C_m) &\leq \sum_{i=1}^k w(C_i) \\
 \Rightarrow w(C) &\leq \sum_{i=1}^k w(C_i) - w(C_m) \\
 \Rightarrow w(C) &\leq \sum_{i=1}^k w(C_i) - \frac{1}{k} \sum_{i=1}^k w(C_i) \\
 \Rightarrow w(C) &\leq (1 - \frac{1}{k}) \sum_{i=1}^k w(C_i)
 \end{aligned}$$

Or par définition, C_i est une coupe minimale séparant s_i des autres sommets terminaux, on a donc : $w(C_i) \leq w(C_i^*)$. Ce qui nous donne :

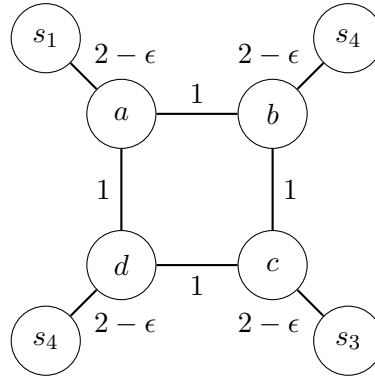
$$w(C) \leq (1 - \frac{1}{k}) \sum_{i=1}^k w(C_i^*)$$

Or nous avons vu que $\sum_{i=1}^k w(C_i^*) = 2w(C^*)$, on a alors :

$$w(C) \leq 2(1 - \frac{1}{k})w(C^*)$$

Atteindre la borne

Considérons le graphe suivant :



En utilisant l'algorithme d'approximation donné, il faut calculer pour chaque terminal la coupe minimale séparant le terminal de tous les autres. Les coupes retournées sont donc les suivantes :

$$C_1 = \{(s_1 a)\}, C_2 = \{(s_2 b)\}, C_3 = \{(s_3 c)\} \text{ et } C_4 = \{(s_4 d)\}$$

Les coupes étant égales, on choisit de retirer C_4 , on obtient alors :

$$C = \{(s_1 a), (s_2 b), (s_3 c)\}$$

On a donc $w(C) = 6 - 3\epsilon$.

Il est facile de voir que la coupe multi séparatrice minimale pour le graphe donné est donnée par :

$$C^* = \{(ab), (bc), (cd), (ad)\}$$

Ce qui nous donne $w(C^*) = 4$.

Or : $6 = 8 \times (1 - \frac{1}{k})$, donc lorsque ϵ tend vers 0, l'instance donnée tend vers la borne précédemment calculée.