TD 10 – Algorithme d'approximation pour le problème de coupe *multiway* optimale

Margot Catinaud margot.catinaud@lmf.cnrs.fr
Valentin Dardilhac valentin.dardilhac87@gmail.com

Dans tout ce TD, on considère $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ un graphe connexe et non-orienté et on y associe une fonction de poids $\omega : \mathcal{E} \longrightarrow \mathbb{R}_+^*$. Pour un sous-ensemble d'arêtes $\mathcal{A} \subseteq \mathcal{E}$, on note $\omega(\mathcal{A}) = \sum\limits_{e \in \mathcal{A}} \omega(e)$.

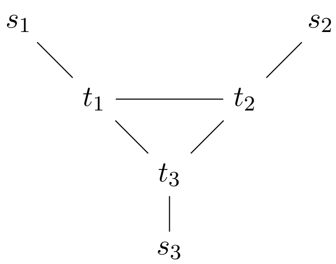
I Coupe *multiway*

Étant donné un ensemble de sommets $\mathbb{S} \stackrel{\text{def}}{=} \{s_i\}_{i=1}^n \subseteq \mathcal{V}$, une coupe *multiway* (en anglais, *multiway cut*) est un ensemble $\mathcal{A} \subseteq \mathcal{E}$ d'arêtes tel que, pour toute paire $(i, j) \in [\![1; n]\!]^2$ avec $i \neq j$, s_i et s_j ne sont pas dans la même composante connexe dans le graphe $\mathcal{G}' = (\mathcal{V}, \mathcal{E} \setminus \mathcal{A})$. Le problème de la **coupe *multiway* optimale** est de trouver *multiway* $\mathcal{A}$ de poids $\omega(\mathcal{A})$ minimal, *i.e.* le problème est de trouver un sous-ensemble $\mathcal{A}$ d'arêtes tel que

$$\mathcal{A} \in \underset{\mathcal{A}' \subset \mathcal{E}}{\arg\min}\, \omega(\mathcal{A}').$$

Pour un entier $n \geqslant 3$ fixé, le problème est NP-dur. Pour $n = 2$, ce problème est résoluble en temps polynomial (avec un algorithme de calcul de flot maximum). Par la suite, on va utiliser ce résultat en boîte noire. Lorsque $\mathbb{S} = \{s, t\}$, on parlera de (s, t)-*coupe* en lieu et place de "*coupe* multiway ".
Le but de cet exercice est de donner un algorithme d'approximation $2\left(1 - \frac{1}{n}\right)$ pour le problème de la coupe *multiway* optimale.

Question 1 *Trouver une coupe* multiway *optimale pour $\mathbb{S} = \{s_1, s_2, s_3\}$ dans le graphe ci dessous, où $\omega(s_i, t_i) = 2 - \varepsilon$ (pour un certain $\varepsilon > 0$) et $\omega(t_i, t_j) = 1$.*



Considérons l'algorithme suivant :

Question 2 *Montrer que $\mathcal{C} =$ **coupe_multiway** $\mathcal{G}$ est une coupe* multiway *pour $\mathbb{S} = \{s_i\}_{i=1}^n$.*

Question 3 *Expliquer comment réaliser la boucle en temps polynomial. Notons que l'on cherche à utiliser le cas $n = 2$.*

Question 4 Supposons que $\mathcal{A}$ est une coupe *multiway* optimale. On peut voir $\mathcal{A}$ comme une union de n coupes de la manière suivante :

(a) Le graphe $\mathcal{G}' = (\mathcal{V}, \mathcal{E} \setminus \mathcal{A})$ contient n composantes connexes, chacune contenant un sommet s_i. *Expliquer pourquoi $\mathcal{G}'$ ne peut pas contenir plus de n composantes.*

Algorithme 1 : Algorithme de calcul d'une coupe multiway

fun coupe_multiway $\mathcal{G}$ =
  pour $i = 1$ **à** n **faire**
    trouver une coupe optimale $\mathcal{C}_i \subseteq \mathcal{E}$ qui sépare le sommet $s_i \in \mathbb{S}$ de $\mathbb{S} \setminus \{s_i\}$;
  $p \leftarrow \underset{i \in [\![1;n]\!]}{\arg\max}\, \omega(\mathcal{C}_i)$;
  $\mathcal{C} \leftarrow \bigcup_{i \in [\![1;n]\!],\, i \neq p} \mathcal{C}_i$;
  retourner $\mathcal{C}$;

(*b*) Notons par $\mathcal{A}_i \subset \mathcal{A}$ la coupe qui sépare s_i des autres noeuds dans $\mathbb{S} \setminus \{s_i\}$. *Montrer que l'on a*

$$\sum_{i=1}^{m} \omega(\mathcal{A}_i) = 2\omega(\mathcal{A}).$$

Question 5 *Montrer que, pour tout* $i \in [\![1;n]\!]$, $\omega(\mathcal{C}_i) \leqslant \omega(\mathcal{A}_i)$.

Question 6 *Montrer que l'on a la minoration suivante :*

$$\omega(\mathcal{C}_n) \geqslant \frac{1}{n} \sum_{i=1}^{n} \omega(\mathcal{C}_i).$$

Question 7 *Conclure que l'algorithme fournit une garantie de performance égale à* $2\left(1 - \frac{1}{n}\right)$.

Question 8 *Donner un exemple de graphe qui montre que la garantie de performance* $2\left(1 - \frac{1}{n}\right)$ *trouvée à la question précédente est optimale pour cet algorithme.*

II Arbres de Gomory-Hu

Notons par $\mathcal{T} = (\mathcal{V}, \mathcal{E}')$ un arbre ayant le même ensemble de sommets que le graphe $\mathcal{G}$, mais où $\mathcal{E}'$ n'est pas nécessairement contenu dans $\mathcal{E}$. On associe à $\mathcal{T}$ une fonction de poids $\omega' : \mathcal{E}' \longrightarrow \mathbb{R}_+^*$.
Pour $\{u, v\} \in \mathcal{E}'$, le graphe $\mathcal{T}' = (\mathcal{V}, \mathcal{E}' \setminus \{\{u, v\}\})$ contient deux composantes connexes $\mathcal{T}_u = (\mathcal{V}_u, \mathcal{E}_u)$ et $\mathcal{T}_v = (\mathcal{V}_v, \mathcal{E}_v)$ telles que $\mathcal{V} = \mathcal{V}_u \sqcup \mathcal{V}_v$, $u \in \mathcal{V}_u$, $v \in \mathcal{V}_v$ et $\mathcal{E}_u \sqcup \mathcal{E}_v \cup \{\{u, v\}\} = \mathcal{E}'$. On appelle *coupe associée à l'arête* $\{u, v\}$ *dans* $\mathcal{G}$ la coupe de $\mathcal{G}$ définie par la partition $(\mathcal{V}_u, \mathcal{V}_v)$ et on la dénote par $\mathcal{A}(\{u, v\})$. Une (u, v)*-coupe de* $\mathcal{G}$ est une coupe *multiway* pour $\mathbb{S} = \{u, v\}$.

Définition 1 : Arbre de Gomory-Hu

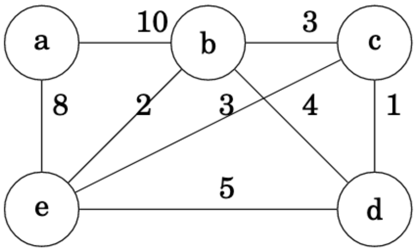
L'arbre $\mathcal{T} = (\mathcal{V}, \mathcal{E}', \omega')$ est un *arbre de Gomory-Hu* pour $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ lorsque

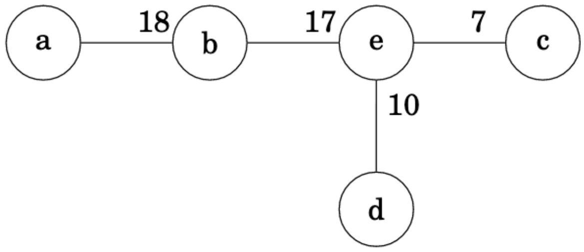
- Pour tout sommets $u, v \in \mathcal{V}$, si $\mathcal{A}$ est une (u, v)-coupe optimale dans $\mathcal{G}$ et $\mathcal{A}'$ est une (u, v)-coupe optimale dans $\mathcal{T}$, alors $\omega(\mathcal{A}) = \omega'(\mathcal{A}')$.
- $\forall\, e \in \mathcal{E}'$, $\omega(\mathcal{A}(e)) = \omega'(e)$.

Le but de cette partie est de trouver un algorithme qui calcule un arbre de Gomory-Hu pour un graphe $\mathcal{G}$ [1].

1. L'arbre de Gomory-Hu permet de résoudre divers problèmes d'optimisation combinatoire sur les graphes. Par exemple, le problème de la n-coupe optimale. On dit qu'un ensemble d'arêtes $\mathcal{A} \subseteq \mathcal{E}$ est une n*-coupe* lorsque $\mathcal{G}' = (\mathcal{V}, \mathcal{E} \setminus \mathcal{A})$ contient n composantes connexes. Le problème de la n-coupe optimale est de trouver une n-coupe $\mathcal{A}$ avec $\omega(\mathcal{A})$ minimal. Pour n fixé, le problème est résoluble en temps polynomial. Lorsque n fait partie de l'entrée, le problème est NP-dur.

Question 9 *Montrer que l'arbre à droite est un arbre de Gomory-Hu pour le graphe à gauche.*





Question 10 *Notons $f(u,v)$ le poids minimal des (u,v)-coupes dans $\mathcal{G}$. Montrer la propriété suivante :*

$$\text{Si } f(u,v) \leqslant f(u,w) \leqslant f(v,w), \text{ alors } f(u,v) = f(u,w).$$

Question 11 *Montrer que, parmi les $\binom{\mathsf{Card}(\mathcal{V})}{2}$ valeurs $f(u,v)$, il y a au plus $\mathsf{Card}(\mathcal{V}) - 1$ valeurs distinctes.*

Question 12 *Montrer les propriétés suivantes :*

1. $\forall\, u,v,w \in \mathcal{V},\ f(u,v) \geqslant \min\big(f(u,w), f(w,v)\big)$;
2. $\forall\, u,v,w_1,\ldots,w_n \in \mathcal{V},\ f(u,v) \geqslant \min\limits_{i=0}^{n} f(w_i, w_{i+1})$ *avec* $w_0 \stackrel{def}{=} u$ *et* $w_{n+1} \stackrel{def}{=} v$.

Question 13 *Soit $\mathbb{K} = \Big(\mathcal{V}, \mathcal{V}^2 \setminus \big\{\{u,u\} \mid u \in \mathcal{V}\big\}\Big)$ le graphe complet où l'on associe à l'arête $\{u,v\}$ le poids $f(u,v)$. Soit $\mathcal{T}$ un arbre couvrant de $\mathbb{K}$ de poids maximal. Montrer que $\mathcal{T}$ satisfait la première condition d'un arbre de Gomory-Hu. On pourra considérer pour cela l'unique chemin de u à v dans $\mathcal{T}$.*

Question 14 *Soit $(\mathcal{V}_1, \mathcal{V}_2)$ une (s,t)-coupe optimale telle que $s \in \mathcal{V}_1$. Soient $x, y \in \mathcal{V}_1$ deux sommets. Considérons le graphe $\mathcal{G}' = \big(\mathcal{V}_1 \cup \{v\}, \mathcal{E}'\big)$ obtenu à partir de $\mathcal{G}$ en fusionnant tous les sommets de $\mathcal{V}_2$ en un sommet $v \notin \mathcal{V}$. Le poids d'une arête (a,v) dans le graphe $\mathcal{G}'$ est donnée par*

$$\omega'(a,v) \stackrel{def}{=} \sum_{b \in \mathcal{V}_2} \omega(a,b).$$

Montrer qu'une (x,y)-coupe optimale de $\mathcal{G}'$ définit une (x,y)-coupe optimale de $\mathcal{G}$ de même poids.

L'algorithme maintient une partition $(\mathbb{S}_i)_{i=1}^{t}$ de $\mathcal{V}$ et un arbre $\mathcal{T}$ dont les noeuds sont $\mathbb{S}_i$ pour $i \in [\![1;t]\!]$. Les poids des arêtes de $\mathcal{T}$ sont donnés par ω'. L'arbre $\mathcal{T}$ satisfait la propriété $\mathcal{I}$ suivante :

$$\textit{Pour toute arête } (\mathbb{S}_i, \mathbb{S}_j) \textit{ de } \mathcal{T}, \textit{ il existe deux sommets } s \in \mathbb{S}_i \textit{ et } t \in \mathbb{S}_j \textit{ tels que } \omega'(\mathbb{S}_i,\mathbb{S}_j) = f(s,t)$$
$$\textit{et la coupe définie par l'arête } (\mathbb{S}_i,\mathbb{S}_j) \textit{ est une } (s,t) \textit{ coupe optimale de } \mathcal{G}. \quad (\mathcal{I})$$

L'algorithme commence avec la partition triviale $\mathcal{V}$. À chaque étape, l'algorithme choisit un élément de la partition $\mathbb{S}_i$ tel que $\mathsf{Card}(\mathbb{S}_i) \geqslant 2$ et le raffine de la manière suivante :

(1) On choisit deux sommets $a, b \in \mathbb{S}_i$;

(2) On construit $\mathcal{G}'$ de la manière suivante :

  (a) $\mathcal{G}'$ contient tout les sommets de $\mathbb{S}_i$;

  (b) En prenant $\mathbb{S}_i$ comme racine de $\mathcal{T}$, on représente chaque fils $\mathbb{S}_j$ de $\mathbb{S}_i$ par un sommet de $\mathcal{G}'$ en fusionnant les sommets de $\mathbb{S}_j$;

  (c) Les arêtes de $\mathcal{G}'$ sont pondérées comme à la question 14.

(3) On calcule $(\mathbb{A}, \mathbb{B})$ une (a,b)-coupe optimale de poids $\omega_{a,b}$ dans $\mathcal{G}'$;

(4) On sépare $\mathbb{S}_i$ en $\mathbb{S}_i^{(a)} \stackrel{\text{def}}{=} \mathbb{S}_i \cap \mathbb{A}$ et $\mathbb{S}_i^{(b)} \stackrel{\text{def}}{=} \mathbb{S}_i \cap \mathbb{B}$ et on ajoute une arête $\left(\mathbb{S}_i^{(a)}, \mathbb{S}_i^{(b)}\right)$ de poids $\omega'\left(\mathbb{S}_i^{(a)}, \mathbb{S}_i^{(b)}\right) \stackrel{\text{def}}{=} \omega_{a,b}$;

(5) Lorsqu'un sous-arbre $\mathbb{S}_j$ est dans $\mathbb{A}$, alors on ajoute une arête entre $\mathbb{S}_i^{(a)}$ et $\mathbb{S}_j$ du même poids que l'arête $(\mathbb{S}_i, \mathbb{S}_j)$ de l'étape précédente ;

(6) Lorsque $\mathbb{S}_j$ est dans $\mathbb{B}$, alors on ajoute une arête entre $\mathbb{S}_i^{(b)}$ et $\mathbb{S}_j$ du même poids que l'arête de l'étape précédente.

Question 15 *Montrer que la propriété $\mathcal{I}$ est invariante pour chaque étape.*

Devoir maison 10 *Conclure que l'algorithme termine et est correct,* i.e. *il retourne effectivement un arbre de Gomory-Hu.*