Master Informatique Fondamentale (1ère année)
Algorithmes d'approximation — **TD2**

La méthode du mille feuilles

15 octobre 2004

Exercice 1 : Couverture par sommets. Nous considérons ici le problème de la couverture par sommets : étant donné un graphe non-orienté $G = (V, E)$ et une fonction de poids sur les sommets $w : V \to \mathbb{Q}_+$, trouver $S \subseteq V$ de poids minimum qui couvre les arêtes (c'est à dire $\forall e \in E$, $\exists s \in S : s \in e$). En cours, nous avons vu une 2-approximation pour le cas où w est constante. Nous allons étudier ici une 2-approximation dans le cas général. Pour ce faire nous nous intéressons à un type de fonction de poids particulier : les fonctions de poids par degré. w est par degré si

$$\exists c > 0, \forall v \in V : w(v) = c \cdot deg(v).$$

Question 1. *Soit $\big(G = (V, E), w\big)$ une instance du problème telle que w est par degrés. Montrer que $w(V) \leq 2OPT$.*

Réponse: OPT couvre toutes les arêtes, donc a : $\sum_{v \in OPT} deg(v) \geq m$ et par définition de c il vient : $OPT \geq c \cdot m$ or $w(v) = c. \sum_{v \in V} deg(v) = 2 \cdot c \cdot m$ d'où $w(v) \leq 2.OPT$. □

On sait donc traiter les instances avec une fonction de poids par degrés. La méthode du mille feuilles consiste alors à décomposer une instance quelconque en une famille d'instances avec fonction de poids par degrés.

Question 2. *À chaque étape de la décomposition, on cherche la plus grande fonction de poids par degrés inférieure à w. Expliciter.*

Réponse: On cherche une fonction w' telle que :
- $\forall v \in V, w'(v) \leq w(v)$ (inférieure à w),
- $\exists c', w'(v) = c' \cdot deg(v)$ (par degrés),
- w' est maximale pour les fonctions par degrés.

En travaillant dans $\mathbb{R} \cup +\infty$:

$$\forall x \in V, c' \leq \frac{w(v)}{deg(v)} \;\Leftrightarrow\; c' \leq min_{v \in V} \frac{w(v)}{deg(v)}.$$

On prend alors $c' = min_{v \in V} \frac{w(v)}{deg(v)}$ car c'est le plus grand que l'on puisse choisir. □

L'algorithme du mille feuilles est le suivant sur l'instance $\big(G = (V, E), w\big)$:

1. $t = 0$, $G_0 = G$ et $w_0 = w$;
2. Tant que G_t contient une arête faire :
   (a) $D_t \leftarrow \{u \in V_t : deg_t(u) = 0\}$;
   (b) Soit p_t la plus grande fonction de poids par degrés inférieure à w_t dans G_t ;
   (c) $S_t \leftarrow \{u \in V_t : p_t(u) = w_t(u)\}$;
   (d) $G_{t+1} \leftarrow G_t \setminus \big(D_t \cup S_t\big)$ et $w_{t+1} \leftarrow w_t - p_t$;
   (e) $t \leftarrow t + 1$;

3. Retourner $C = \bigcup_{k=0}^{t-1} S_k$.

Question 3. *Montrer que l'algorithme termine en temps polynomial.*

Réponse: On a à chaque étape $|S_t| \geq 1$, car p_t est maximale et $G_{t+1} \leftarrow G_t \setminus (D_t \cup S_t)$ donc $|G_{t+1}| \leq |G_t|$.
On exécute l'étape 2 au plus n fois, avec :
- pour (a) $O(n)$
- pour (b) $O(n)$
- pour (c) $O(n)$
- pour (d) $O(n^2)$
- pour (e) $O(1)$

On a donc bien un algorithme calculant en temps polynomial. □

Question 4. *Montrer que l'ensemble C de sommets retourné par l'algorithme est bien une couverture.*

Réponse: Soit $e = (u_l, u_m) \in E$, $\exists i$ tel que $e \in G_i, e \notin G_{i+1}$. On peut supposer : $u_l \in (S_i \cup D_i)$. On a $deg_i(u_l) \geq 1 \Rightarrow u_l \notin D_i$ et donc $u_l \in S_i$. Alors $u_l \in C$ et e est couvert par C. Donc C est bien une couverture. □

On note dorénavant C la couverture renvoyée par l'algorithme et C^* une couverture optimale.

Question 5. *Pour tout $v \in C$ exprimer $w(v)$ en fonction des poids par degrés p_k. Qu'en est-il pour $v \notin C$?*

Réponse: A chaque étape, $w_{t+1} \leftarrow w_t - p_t$. Soit $v \in C$: $v \in S_j$ et on cherche un majorant de $w(v)$. $w(v) = \sum_{k=0}^{j-1} p_k(v) + w_j(v)$ et $v \in S_j$ donc $w_j(v) = p_j(v)$. D'où $w(v) = \sum_{k=0}^{j} p_k(v)$.
Si $v \notin C$: $v \in D_K$ et on cherche un minorant de $w(v)$.

$$w(v) = \sum_{k=0}^{K-1} p_k(v) + w_k(v) \geq \sum_{k=0}^{K} p_k(v)$$

car ($p_K(v) = 0$). □

Question 6. *Pour toute étape i de l'algorithme, comparer $p_i(C \cap G_i)$ et $p_i(C^* \cap G_i)$. (Indication : on remarquera que $C^* \cap G_i$ est une couverture de G_i).*

Réponse: Une couverture est stable par restriction à un sous-graphe. Soit OPT une couverture optimale de (G_i, p_i). D'apres la question 1, $p_i(C \cap G_i) \leq 2 \cdot OPT$. Comme $C^* \cap G_i$ est une couverture de G_i, $p_i(C^* \cap G_i) \geq OPT$. Donc $p_i(C \cap G_i) \leq 2 \cdot p_i(C^* \cap G_i)$. □

Question 7. *Montrer que l'algorithme est une 2-approximation du problème de la couverture par sommet minimum avec des poids arbitraires. Montrer que l'analyse de l'algorithme est exacte.*

Réponse: Convention : si v n'est pas dans un graphe, son poids est nul. Soit T l'étape finale.
$w(C) = \sum_{v \in C} w(v) = \sum_{v \in C} \sum_{i=0}^{T} p_i(v)$
$w(C) = \sum_{i=0}^{T} \sum_{v \in C} p_i(v)$ (inversion des sommes)
$w(C) = \sum_{i=0}^{T} p_i(G_i \cap C)$ (d'apres question 5)
$\leq \sum_{i=0}^{T} 2 \cdot p_i(G_i \cap C^*)$
$w(C^*) = \sum_{v \in C^*} w(v) \geq \sum_{i=0}^{k} p_i(G_i \cap C^*)$ (d'apres question 5).
D'où : $w(C) \leq 2 \cdot w(C^*)$
Un graphe biparti complet avec des poids par degré (n/2 partout) constitue une instance critique car l'algo prend tous les sommets alors que l'optimal en prend la moitié. □

♣