TD d'algorithmique numéro 12
Daniel Hirschkoff
18 décembre 2000

**Question 1** ⋆ Couverture des sommets d'un graphe ⋆

On couvre un graphe non orienté $G = (S, A)$ en se donnant $S' \subseteq S$ vérifiant la propriété suivante: pour toute arête $a$ de $A$, au moins l'un des sommets reliés par $a$ est dans $S'$. La *taille* de la couverture $S'$ est alors définie comme le nombre d'éléments de cet ensemble. On cherche une couverture de taille minimale.

**Question 1.1**

Exprimer le problème sous forme d'un problème de décision. Montrer que ce problème appartient à NP.

**Question 1.2**

Montrer, en raisonnant avec la notion de *complément* d'un graphe non orienté (le graphe comportant les arêtes non présentes dans le graphe de départ), que le problème de la couverture des sommets d'un graphe est NP-complet.

**Question 2** ⋆ Somme d'un sous-ensemble ⋆

On se donne un ensemble fini $E \subseteq \mathbb{N}$ et une valeur *cible* $t \in \mathbb{N}$, et on cherche un ensemble $E' \subseteq E$ tel que la somme des éléments de $E'$ soit égale à $t$.

**Question 2.1** ⋆ NP complétude ⋆

On veut montrer que le problème de la somme d'un sous-ensemble est NP-complet en le réduisant à la couverture des sommets d'un graphe $G$. On suppose dans ce qui suit que les algorithmes travaillent sur des entiers codés en binaire.

Dire que le problème appartient à NP.

On se donne, pour montrer que *Couverture_Sommets* $\leq_P$ *Somme_Sous_Ensemble*, un graphe non orienté $G = (S, A)$ (avec $S = \{v_0, v_1, \ldots, v_{|S|-1}\}$ et $A = \{e_0, e_1, \ldots, e_{|A|-1}\}$), que l'on représente par une matrice d'incidence $B = b_{ij}$ définie de la manière suivante:

$$b_{ij} = \begin{cases} 1 & \text{si l'arête } e_j \text{ arrive au sommet } v_i \\ 0 & \text{sinon.} \end{cases}$$

On part également d'un entier $k$ correspondant à l'instance du problème de couverture du graphe sur laquelle on travaille (i.e. on cherche une couverture de taille $k$).

L'idée est de construire un ensemble $E$ d'entiers qui représentent le graphe. Ces entiers seront écrits en *base 4 modifiée* (B4M), ce qui signifie la chose suivante: les $|A|$ chiffres moins significatifs d'un nombre sont exprimés en base 4, et le chiffre le plus significatif prend une valeur comprise (au sens large) entre zéro et $k$. On définit ensuite $E = \{x_i\} \cup \{y_j\}$ de la manière suivante:

- pour chaque sommet $v_i$, on crée un entier positif $x_i$ dont la représentation B4M est constituée d'un 1 suivi de $|A|$ chiffres; ces derniers correspondent à la ligne de $v_i$ dans $B$. Autrement dit, on a:

$$x_i \;=\; 4^{|A|} + \sum_{0 \le j \le |A|-1} b_{ij} 4^j \,.$$

- Pour chaque arête $e_j \in A$, on crée un entier positif $y_j$ représentant une ligne de la matrice identité, i.e.

$$y_j \;=\; 4^j \,.$$

On se donne ensuite $t$ en posant:

$$t \;=\; k4^{|A|} + \sum_{0 \le j \le |A|-1} 2.\,4^j \,.$$

1. Supposons que le graphe $G$ possède une couverture de sommets $S' \subseteq S$ de taille $k$, avec $S' = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$. On lui associe

$$\begin{aligned} E' \;=\; & \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\} \cup \\ & \{y_j.\, e_j \text{ est incidente à un seul sommet de } S'\} \,. \end{aligned}$$

   Montrer que $\sum_{s \in E'} s = t$.

2. Réciproquement, soit $E' \subseteq E$ un ensemble dont la somme des éléments vaut $t$; on écrit $E' = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\} \cup \{y_{j_1}, y_{j_2}, \ldots, y_{j_p}\}$. Montrer que $m = k$ et que $S' = \{v_{i_1}, v_{i_2}, \ldots, v_{i_m}\}$ est une couverture de sommets de $G$.

3. Conclure.

**Question 2.2** ⋆ Algorithme en temps exponentiel ⋆

On va définir une solution optimale avec un algorithme en temps exponentiel. On écrit l'algorithme suivant (on suppose disposer d'une fonction *Fusionner* de fusion de listes):

Somme_Exacte($S$,$t$):=
$n$:=$|S|$
$L_0$:=[0]
`pour` $i$ `de` 1 `à` $n$ `faire`
$L_i$ := *Fusionner*$(L_{i-1}, L_{i-1} + x_i)$
supprimer de $L_i$ les éléments supérieurs à $t$
`finpour`
Retourner le max de $L_n$

Dans l'algorithme ci-dessus, $L_{i-1} + x_i$ désigne la liste d'entiers résultant de $L_{i-1}$ en ajoutant $x_i$ à tous les éléments de la liste.

1. Soit $P_i$ l'ensemble des valeurs obtenues en effectuant les sommes de tous les sous-ensembles de $x_1, \ldots, x_i$. Montrer que

$$P_i \;=\; P_{i-1} \cup (P_{i-1} + x_i)\,.$$

   Montrer que $L_i$ est une liste triée des éléments de $P_i$ de valeur inférieure à $t$.

2. Quelle est la taille maximale de $L_i$? Qu'en déduit-on eu égard au temps d'exécution de l'algorithme?

**Question 2.3** ⋆ Schéma d'approximation en temps entièrement polynomial ⋆

On s'attache à présent à limiter la taille de $L_i$, pour obtenir un schéma d'approximation en temps entièrement polynomial. On va pour ce faire introduire une notion de *seuil*, qui va être un réel $\delta$ appartenant à $]0;1]$. À ce seuil correspond une transformation d'une liste $L$ en $L'$ selon le principe suivant:

$$y \in L \;\wedge\; y \notin L' \;\Leftrightarrow\; \exists z \in L'.\, z \leq y \;\wedge\; \frac{(y-z)}{y} \leq \delta$$

(NB: la condition ci-dessus peut aussi s'écrire $(1-\delta)y \leq z \leq y$). L'idée est d'*écrêter* $L$ en $L'$.

1. Écrire l'algorithme de seuil, qui transforme une liste $L$ en une liste écrêtée $L'$; quelle en est la complexité?

2. Adapter l'algorithme exponentiel; le retour de l'algorithme ainsi obtenu est-il la somme d'un sous-ensemble de $S$?

   Soit maintenant $y^* \in P_n$ une solution optimale.

3. Montrer que pour tout $i$ entre 1 et $n$, et pour tout $y \in P_i$ au plus égal à $t$, il y a un $z \in L_i$ tel que

$$(1 - \frac{\epsilon}{n})^i y \ \leq \ z \ \leq \ y \, .$$

   En déduire que le résultat de l'approximation n'est pas inférieur de plus de $1 - \epsilon$ par rapport à $y^*$.

4. Établir une borne sur la longueur de $L_i$ en fonction de $n$ et $t$; qu'en déduit-on?

**Question 3** ⋆ Partition ⋆

Le problème de la partition consiste, étant donnés un ensemble fini $A$ et une fonction $s : A \to \mathbb{N}$, à chercher un ensemble $A' \subseteq A$ t.q.

$$\sum_{a \in A'} s(a) \ = \ \sum_{a \in A \setminus A'} s(a) \, .$$

Montrer que ce problème est NP complet.