Université Paris-Sud
Master d'Informatique (M1)
Année 2012-2013

Algorithmique Avancée : Programmation Linéaire
Responsables : A. Lisser, P. Valicov
Nombre de pages : 3

Théorie de la NP-complétude

Pour montrer qu'un problème de décision A est NP-complet il faut :

- montrer que A appartient à la classe NP. C'est à dire que le certificat (ou la solution) peut être vérifié en temps polynomial. Généralement pour ce faire il suffit de proposer algorithme polynomial.
- montrer que $\mathsf{B} \leq_P \mathsf{A}$, où B est un autre problème NP-complet. Le symbole $\leq_P$ signifie "B se réduit en temps polynomial à A". Autrement dit, pour toute instance I_{B} du problème B on peut construire **en temps polynomial** une instance I_{A} du problème A tel que I_{B} admet comme réponse OUI si et seulement si I_{A} admet comme réponse OUI.

Exercice I - 3-Coloration

On rappelle qu'une coloration de sommets d'un graphe est une application qui associe à chaque sommet une couleur (un nombre) tel que deux sommets adjacents n'ont pas la même couleur. Le problème de décision k-COLORATION (ou k-COL en plus simple) est le suivant :

INSTANCE : Un graphe G.
QUESTION : Est-ce que G peut être colorié avec au plus k couleurs ?

1. Prouver que k-COL appartient à la classe NP.

   Remarquez que 1-COL et 2-COL sont des problèmes qui peuvent être résolus en temps polynomial, voire linéaire...
   On souhaite prouver que 3-COL est un problème NP-complet. Pour ce faire nous allons utiliser 3-SAT.
   SAT est un problème qui consiste en un ensemble $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ de variables booléennes et un ensemble $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ de clauses. Chaque clause c_i contient un ensemble de littéraux x_j ou $\overline{x}_j$ avec un OU entre eux. Voici un exemple de clause : $c_i = (x_j \vee x_k \vee \overline{x}_l)$. Les clauses sont reliées entre elles avec un ET : $(c_1 \wedge c_2 \wedge \ldots \wedge c_m)$.
   Le problème de décision est le suivant :

   INSTANCE : Un ensemble de variables booléennes $\mathcal{X}$ et un ensemble de clauses $\mathcal{C}$.
   QUESTION : Est-ce que $\mathcal{C}$ est satisfiable ? Plus précisément est-ce qu'il existe une assignation de variables tel que chaque clause de $\mathcal{C}$ est VRAI.

   Dans le cas de 3-SAT la restriction est qu'il y a **au plus 3** littéraux par clause.

   Maintenant nous allons montrer comment on peut *encoder* l'instance I de 3-SAT dans un graphe G tel que I est satisfiable si et seulement si G admet une coloration avec au plus 3 couleurs.

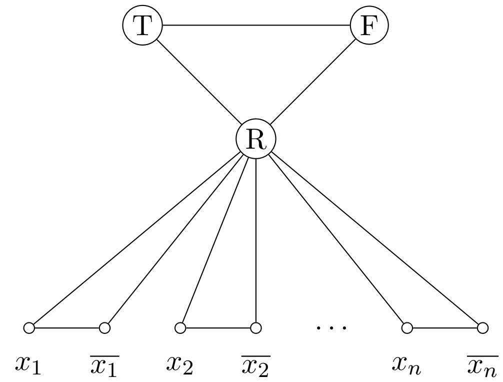

Figure 1 – Le gadget de variables

2. L'ensemble des variables de I va être représenté par le graphe donné dans la Figure 1. Sans perte de généralité, nous allons supposer que le triangle est colorié tel que c'est indiqué dans la figure. Quelles peuvent être les couleurs de $x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots, x_n, \overline{x_n}$?
3. Pour chaque clause dans l'instance de 3-SAT on va construire le graph donné dans la Figure 2. Dans l'exemple de la figure la clause est $(x_j \vee, x_k \vee \overline{x_l})$. Les sommets x_i ou $\overline{x_i}$ ainsi que le sommet colorié avec T sont les mêmes que ceux dans la Figure 1.

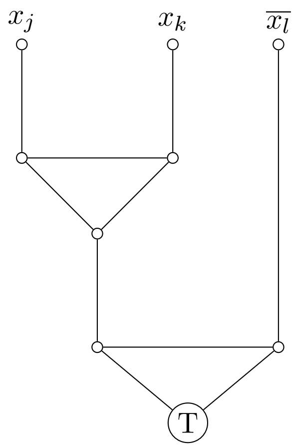

Figure 2 – Le gadget de clause

En utilisant la réponse à la question précédente qu'en déduisez-vous ?
4. Montrer que si I est satisfiable alors il existe une coloration de G avec au plus 3 couleurs.
5. Montrer que si G admet une coloration avec au plus 3 couleurs alors I est satisfiable.

Exercice II - Cycle Hamiltonien vs TSP

Un cycle hamiltonien d'un graphe est un cycle qui passe par tous les sommets du graphe **une et une seule** fois. Le problème de décision associé (CH) est le suivant :

INSTANCE : Un graphe G.
QUESTION : Est-ce que G possède un cycle hamiltonien ?

Avec une réduction à partir de 3-SAT il est possible de montrer que CH est NP-complet.

Le problème du voyageur du commerce (TSP) est le suivant :

INSTANCE : Un ensemble $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ de villes, une distance $d(c_i, c_j) \in \mathbb{Z}^+$ entre chaque paire de villes c_i, c_j et un entier B.

QUESTION : Est-ce qu'il existe un "tour" passant par toutes les villes de $\mathcal{C}$ ayant une longueur au plus égale à B ?

1. Montrer que TSP est dans NP.
2. Montrer que TSP est NP-complet en faisant une réduction à partir de CH.

Exercice III - Sac-à-Dos

Le problème du Sac-à-Dos est défini comme suit :

INSTANCE : Un ensemble d'items U, une "taille" $s(u) \in \mathbb{Z}^+$ et une valeur $v(u) \in \mathbb{Z}^+$ pour tout $u \in U$, une contrainte de taille $B \in \mathbb{Z}^+$ et une valeur objectif $K \in \mathbb{Z}^+$.

QUESTION : Est-ce qu'il existe un sous-ensemble $U' \subseteq U$ tel que

$$\sum_{u \in U'} s(u) \leq B \ \text{ et } \sum_{u \in U'} v(u) \geq K$$

Montrer que le problème du Sac-à-Dos est un problème NP-complet. Pour cela il faudrait réduire le problème PARTITION :

INSTANCE : Un ensemble d'entiers E et une "taille" $s(e) \in \mathbb{Z}^+$ pour tout $e \in E$.

QUESTION : Est-ce qu'il existe un sous-ensemble $E' \subseteq E$ tel que :

$$\sum_{e \in E'} s(e) = \sum_{e \in U - U'} v(e)$$