M1 ALMA/ATAL/ORO
X8II110

Université de Nantes
2016/2017

# *Contrôle continu*
# Data Structures and Algorithms

Mars 2017

*Durée : 1h20. Documents de CM/TD autorisés. Présentation, clarté et orthographe seront pris en compte dans la note finale. Il est également important de bien justifier toutes vos réponses. Le barème est indicatif.*

**Exercice 1 (Algorithme glouton pour MIN VERTEX COVER - 4 points)**

On rappelle la définition du problème MIN-VERTEX-COVER (MIN-VC).

> MIN-VERTEX-COVER (MIN-VC)
> *Instance* : un graphe $G = (V, E)$, où $V = \{v_1, v_2 \dots v_n\}$
> *Solution* : un ensemble $V' \subseteq V$ qui couvre toutes les arêtes de $G$
> *Mesure* : le nombre $|V'|$ de sommets dans $V'$

Nous avons vu en CM un algorithme d'approximation de ratio 2 pour MIN-VC. Appelons cet algorithme $\mathcal{G}$ (comme glouton). On appelle $K_{n,n} = (V_1 \cup V_2, E)$ le graphe biparti dans lequel l'ensemble des sommets est partitionné en $V_1$ et $V_2$, tel que $|V_1| = |V_2| = n$, et où toutes les arêtes possibles existent entre $V_1$ et $V_2$.
On appelle $opt_n$ la taille d'un plus petit vertex cover dans $K_{n,n}$.

1. Soit $X$ un vertex cover quelconque de $K_{n,n}$. Est-il possible d'avoir $v_1 \in V_1$, $v_2 \in V_2$, $v_1 \notin X$ et $v_2 \notin X$ ? Pourquoi ?
2. En déduire que $opt_n \geq n$.
3. Montrer que $opt_n \leq n$. Illustrer votre réponse sur $K_{4,4}$.
4. Quelle solution est renvoyée par $\mathcal{G}$ sur $K_{n,n}$, et quelle est la taille de cette solution ? Justifier.
5. Quelle conclusion peut-on en tirer concernant $\mathcal{G}$ ?

**Exercice 2 (Programmation Linéaire et Approximation pour MIN-VC - 6 points)**

Voici la description d'un programme linéaire en nombre entiers (qu'on appellera (IP)), construit à partir d'un graphe $G$ :

$$\begin{aligned} \text{minimize } & x_1 + x_2 + \ldots + x_n & \\ \text{subject to } & x_i + x_j \geq 1 & \forall (v_i, v_j) \in E(G) \\ & x_i \in \{0,1\} & \forall v_i \in V(G) \end{aligned}$$

1. Indiquer comment, partant d'une solution de (IP), on peut construire une solution de MIN-VC qui a le même optimum (on attend ici une réponse précise et argumentée).
2. Indiquer comment, partant d'une solution de MIN-VC, on peut construire une solution de (IP) qui a le même optimum (on attend ici une réponse précise et argumentée).

Les Questions 1. et 2. ci-dessus montrent donc que les problèmes (IP) et MIN-VC sont équivalents.
Voici maintenant la description d'un programme linéaire (qu'on appellera (LP)). Cela veut dire qu'ici les $y_i$ ne sont pas nécessairement des entiers, mais *peuvent prendre n'importe quelles valeurs réelles entre 0 et 1*.

$$\begin{aligned} \text{minimize } & y_1 + y_2 + \ldots + y_n & \\ \text{subject to } & y_i + y_j \geq 1 & \forall (v_i, v_j) \in E(G) \\ & 0 \leq y_i \leq 1 & \forall v_i \in V(G) \end{aligned}$$

Les problèmes de type (LP) peuvent se résoudre en temps polynomial, alors que les problèmes de type (IP) sont NP-complets. L'idée est donc d'utiliser le problème (LP) pour obtenir une approximation du problème (IP), et donc une approximation du problème MIN-VC.

3. Montrer que, pour tout graphe $G$, $opt(LP) \leq opt(IP)$.

Appelons $y^* = (y_1^*, y_2^* \ldots y_n^*)$ une solution optimale obtenue pour (LP). On définit alors $x^A = (x_1^A, x_2^A \ldots x_n^A)$ de la manière suivante : pour tout $1 \leq i \leq n$, $x_i^A = 1$ si $y_i^* \geq \frac{1}{2}$, et $x_i^A = 0$ sinon.

4. Montrer que si $y^*$ est une solution optimale pour (LP), alors $x^A$ est une solution pour (IP) (cela revient à montrer que si $y_i^* + y_j^* \geq 1$, alors $x_i^A + x_j^A \geq 1$).
5. Montrer que pour tout $1 \leq i \leq n$, $x_i^A \leq 2y_i^*$.

Soit $opt(LP) = y_1^* + y_2^* + \ldots y_n^*$, et $sol(IP) = x_1^A + x_2^A + \ldots x_n^A$.

6. Montrer que $sol(IP) \leq r \cdot opt(LP)$, où $r$ est une valeur à déterminer.
7. Conclure quant à l'approximabilité de (IP) et de MIN-VC.
8. Indiquer (en français) les grandes étapes de l'algorithme d'approximation de ratio $r$ pour MIN-VC que nous venons d'étudier.

---

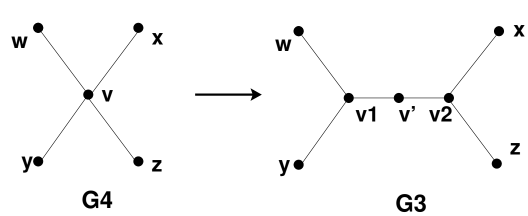


FIGURE 1 – Exercice 3 : transformation de $G_4$ vers $G_3$

**Exercice 3 (MIN-VC3 est APX-dur – 10 points)**

On appelle MIN-VC3 (resp. MIN-VC4) le problème MIN-VC pour lequel les instances d'entrée sont les graphes de degré maximum 3 (resp. de degré maximum 4). Le but de cet exercice est de démontrer que MIN-VC3 est APX-dur, en se basant sur le fait que MIN-VC4 est APX-dur (ce que nous avons vu en TD, voir Exercice 1.1).

1. Montrer que dans un graphe $G_d$ de degré maximum $d$ et possédant $n$ sommets, tout vertex cover de $G_d$ est de taille au moins égale à $\frac{n}{d+1}$.

Partant de n'importe quel graphe $G_4$ de degré maximum 4, on le transforme en un graphe $G_3$ de degré maximum 3 en transformant tous les sommets de degré 4 de la manière décrite dans la Figure 1, page précédente.

On appellera $V_4$ l'ensemble des sommets de $G_4$ qui sont de degré exactement 4, et $n_4 = |V_4|$ le nombre de ces sommets.

Pour commencer, partons d'un vertex cover de $G_4$ qu'on appelle $C_4$, et construisons à partir de lui un vertex cover $C_3$ de $G_3$. Pour cela, on observe un sommet $v$ de $G_4$.

2. Supposons que $v$ est dans $V_4$, et qu'il est aussi dans $C_4$. Montrer que mettre $v_1$ et $v_2$ dans $C_3$ couvre dans $G_3$ les arêtes que couvre $v$ dans $G_4$, ainsi que les arêtes $\{v_1, v'\}$ et $\{v', v_2\}$.
3. Supposons que $v$ est dans $V_4$, et qu'il n'est pas dans $C_4$. Montrer que mettre $v'$ dans $C_3$ couvre dans $G_3$ les arêtes que couvre $v$ dans $G_4$, ainsi que les arêtes $\{v_1, v'\}$ et $\{v', v_2\}$.

Par ailleurs, si $v$ n'est pas dans $V_4$, on ne le met dans $C_3$ que s'il est présent dans $C_4$.

4. Montrer que l'ensemble $C_3$ ainsi obtenu est un vertex cover de $G_3$.

Soit $c_3$ la taille de $C_3$ et $c_4$ la taille de $C_4$.

5. Montrer que $c_3 = c_4 + n_4$.

On admettra que la réciproque est vraie, c'est-à-dire que s'il existe un vertex cover $C'_3$ de taille $c'_3$ dans $G_3$, alors on peut construire à partir de lui un vertex cover $C'_4$ de taille $c'_4 = c'_3 - n_4$.

On appelle $opt(G_3)$ (resp. $opt(G_4)$) la taille du plus petit vertex cover de $G_3$ (resp. $G_4$).

6. Montrer que $opt(G_3) = opt(G_4) + n_4$.

Nous allons maintenant supposer que MIN-VC3 est dans PTAS, c'est-à-dire qu'il existe un algorithme $\mathcal{A}_3$ qui, pour tout graphe $G_3$ de degré maximum 3, détermine un vertex cover de taille $a_3(G_3) \leq (1 + \varepsilon) \cdot opt(G_3)$.

Partant d'un graphe $G_4$ de degré maximum 4, on propose l'algorithme suivant, que l'on appellera $\mathcal{A}_4$ :
— transformer $G_4$ en $G_3$ comme décrit à la Figure 1
— appeler l'algorithme $\mathcal{A}_3$
— transformer le vertex cover obtenu par $\mathcal{A}_3$ pour $G_3$ en un vertex cover pour $G_4$, comme décrit ci-dessus

On appelle $a_3(G_3)$ (resp. $a_4(G_4)$) la taille du vertex cover obtenu par $\mathcal{A}_3$ sur $G_3$ (resp. $\mathcal{A}_4$ sur $G_4$).

7. Montrer que $a_4(G_4) = a_3(G_3) - n_4$.
8. Montrer que $a_4(G_4) \leq (1 + \varepsilon) \cdot opt(G_4) + \varepsilon \cdot n_4$.
9. En utilisant la Question 1., montrer que $a_4(G_4) \leq (1 + 6\varepsilon) \cdot opt(G_4)$.
10. Que signifie le résultat de la question précédente concernant le problème MIN-VC4 ?
11. Sachant que le problème MIN-VC4 est APX-dur, que peut-on en déduire pour MIN-VC3 ? (on attend ici une réponse précise et argumentée)