TD n^{o}9 : Algorithmes d'approximation

Un algorithme d'approximation avec un facteur $c \geqslant 1$ pour un problème de *minimisation* signifie que l'algorithme calcule une solutions S de valeur s tel que :

- la solution S est valide,
- s vérifie $s \leqslant c \cdot \mathrm{OPT}$, où OPT est la valeur d'une solution optimale.

Pour un problème de *maximisation*, le facteur d'approximation vérifie $c \leqslant 1$ et la seconde condition devient $s \geqslant c \cdot \mathrm{OPT}$.
Dans les deux cas, l'approximation est meilleure quand c est plus proche de 1, le cas limite $c = 1$ correspondant à un algorithme exact.

Exercice 1. – Couverture d'un graphe par des sommets.

Couverture de graphe par des sommets
Données : Un graphe $G = (V, E)$.
Objectif : Un sous-ensemble de sommets $V' \subseteq V$ tel que toute arête de G est incidente à V' (i.e. pour tout $(u, v) \in E$, $\{u, v\} \cap V' \neq \emptyset$), avec $|V'|$ minimal.

On propose l'algorithme d'approximation suivant pour la couverture : calculer un couplage maximal (pour l'inclusion) $A \subseteq E$, puis retourner l'ensemble des sommets incidents à A.

a) Montrer que l'algorithme ci-dessus renvoie bien une couverture du graphe.

b) Montrer qu'il est possible de calculer un couplage maximal (pour l'inclusion) en temps polynomial.

c) Soit $M \subseteq E$ un couplage quelconque et $S \subseteq V$ un ensemble de sommets couvrant G. Montrer que $|M| \leqslant |S|$.

d) En déduire que l'algorithme ci-dessus est un algorithme d'approximation avec un facteur 2.

e) Analyser le comportement de l'algorithme sur le graphe biparti complet $K_{n,n}$. En déduire que le facteur d'approximation 2 de cet algorithme ne peut être amélioré par une analyse plus fine (ce qui ne signifie pas qu'on ne peut pas faire mieux avec un autre algorithme).

Exercice 2. – Sous-graphe acyclique.

Sous-graphe acyclique
Données : Un graphe orienté $G = (V, E)$.
Objectif : Un sous-ensemble d'arcs $E' \subseteq E$ tel que (V, E') soit acyclique, avec $|E'|$ maximal.

a) Donner un algorithme d'approximation avec un facteur 1/2 pour le problème ci-dessus.

Indication. Numéroter les sommets de manière quelconque, puis partitionner les arcs en deux, selon que l'arc va du sommet de plus petit numéro au plus grand, ou l'inverse.

EXERCICE 3. – Affectation de tâches sur des machines.

AFFECTATION DE TÂCHES SUR DES MACHINES
Données : n tâches de durées respectives $t_1, t_2, \ldots, t_n$, le nombre m de machines.
Objectif : Une affectation des n tâches sur les m machines telle que le temps de complétion soit minimal.

On propose l'algorithme glouton suivant :

- Ordonner les tâches dans un ordre quelconque ;
- Considérer les tâches dans cet ordre, et assigner à chaque étape la tâche à la machine à laquelle on a donné le moins de travail (jusqu'ici).

a) Montrer que cet algorithme réalise un facteur d'approximation de 2.

b) Considérer l'instance composée de m machines et m^2+1 tâches avec $t_i = 1$ pour $1 \leqslant i \leqslant m^2$ et $t_{m^2+1} = m$. Que peut-on en déduire ?