TD09 – NP-complétude et Approximation

Exercice 1. *Problème du k-centre*

On rappelle quelques définitions :
- Dans un graphe $G = (V, E)$, un *ensemble indépendant* est un sous-ensemble de sommets V' non reliés par des arêtes (si $u \in V'$ et $v \in V'$, alors $(u, v) \notin E$).
- Dans un graphe $G = (V, E)$, un *ensemble dominant* est un sous-ensemble de sommets V' tel que tout sommet de $V \setminus V'$ est adjacent à un sommet de V'. On note $dom(G)$ le cardinal minimal d'un ensemble dominant.

1. Montrer que trouver un ensemble indépendant de cardinal maximal est NP-difficile.

2. Montrer que trouver un ensemble dominant de cardinal minimal $dom(G)$ est NP-difficile.

Soit $G = (V, E)$ un graphe non-orienté, complet, dont les arêtes sont pondérées par une fonction de poids p qui vérifie l'inégalité triangulaire : $p(u, v) \leq p(u, w) + p(w, v)$ pour tout triplet de sommets (u, v, w). Soit aussi un entier $k \geq 1$.
Pour tout $S \subset V$ et tout $v \in V \setminus S$, on définit $connect(v, S)$ comme le poids minimal d'une arête reliant v à un sommet de S : $connect(v, S) = \min_{s \in S} \; p(v, s)$. Le problème est de trouver un k-centre, c'est à dire un sous-ensemble S de cardinal k et tel que

$$center(S) = \max_{v \in V \setminus S} connect(v, S)$$

soit minimal.

3. A quoi peut-il bien servir de déterminer un k-centre (donner un exemple d'application) ?

4. Montrer que trouver un k-centre est NP-difficile.

On va chercher une 2-approximation, i.e. un S de cardinal k tel que $center(S) \leq 2 \cdot OPT$, où $OPT = \min_{S \subset V, |S| = k} \; center(S)$.
On ordonne les arêtes de E par poids croissant : $p(e_1) \leq p(e_2) \leq \ldots \leq p(e_m)$, où $m = |E|$. On pose $G_i = (V, E_i)$ où $E_i = \{e_1, e_2, \ldots, e_i\}$ est l'ensemble des i premières arêtes.

5. Montrer que résoudre le problème du k-centre revient à trouver le plus petit indice i tel que G_i a un ensemble dominant de cardinal au plus k.

6. Étant donné un graphe H, soit I un ensemble indépendant du graphe carré $H^{(2)}$. Montrer que $|I| \leq dom(H)$.

Une dernière définition : le carré d'un graphe $G = (V, E)$, noté $G^{(2)} = (V, E^{(2)})$, contient les chemins de longueur au plus deux : $(u, v) \in E^{(2)}$ si $(u, v) \in E$ ou s'il existe $w \in V$ tel que $(u, w) \in E$ et $(w, v) \in E$
L'algorithme d'approximation du k-centre est le suivant :

1. Construire $G_1^{(2)}, G_2^{(2)}, \ldots, G_m^{(2)}$
2. Trouver de manière gloutonne un ensemble indépendant inextensible (auquel on ne peut pas rajouter des sommets) M_i dans chaque graphe $G_i^{(2)}$
3. Trouver le plus petit indice i tel que $|M_i| \leq k$, soit j cet indice
4. Renvoyer M_j

7. Montrer que :
(i) $p(e_j) \leq OPT$;
(ii) l'algorithme est bien une 2-approximation.

8. Montrer que la borne 2 est stricte : donner un exemple de graphe où l'algorithme réalise effectivement une 2-approximation.

9. Montrer que si $P \neq NP$, il n'existe pas de $(2-\epsilon)$-approximation au problème du k-centre, pour tout $\epsilon > 0$.

Exercice 2. *Sac à dos*

On s'intéresse au problème du sac-à-dos :
Instance : Un ensemble fini X d'objets ; pour chaque objet $x_i \in X$, une valeur $p_i \in \mathbb{N}$ et une taille $a_i \in \mathbb{N}$. Une capacité $B \in \mathbb{N}$.
Solution : Un sous-ensemble Y de X tel que $\sum_{x_i \in Y} a_i \leq B$
Mesure : La valeur totale des objets choisis, i.e. $\sum_{x_i \in Y} p_i$.
On notera $m^*(x)$ la mesure optimale (maximale).

1. Formuler le problème de décision associé et montrer qu'il est NP-complet.

2. Mais alors, comment a-t-on pu résoudre ce problème en cours par programmation dynamique ?

Dans l'algorithme glouton, on trie les éléments par rapport p_i/a_i décroissant, et on choisit chaque élément examiné dans cet ordre s'il y a la place pour lui. Soit $m_g(x)$ la mesure de l'algorithme glouton.

3. Soit K un entier arbitrairement grand. Construire une instance x telle que $m^*(x)/m_g(x) > K$.

4. Soit p_{max} la valeur maximale d'un objet. Montrer que $m^*(x)/\max\{m_g(x), p_{max}\} < 2$.
Indication : soit j l'indice du premier élément que l'algorithme glouton ne prend pas ; montrer que $m^*(x) \leq \sum_{i=1}^{j} p_i$.

On essaie maintenant de résoudre le problème par programmation dynamique. Pour $1 \leq k \leq n$ et $0 \leq p \leq \sum_{i=1}^{n} p_i$, on cherche, parmi les sous-ensembles de $\{x_1, x_2, \ldots, x_k\}$ de valeur totale égale à p et de taille majorée par B, un qui soit de taille minimale. On note $M^*(k,p)$ une solution optimale de ce problème et $S^*(k,p)$ la taille correspondante.

5. Expliquer comment calculer les $M^*(k,p)$ par programmation dynamique. Quelle est la complexité de la résolution du problème du sac-à-dos ?

Pout tout rationnel $r > 1$, on considère le schéma d'approximation suivant : soit p_{max} la valeur maximale d'un objet et $t = \lfloor \log(\frac{r-1}{r} \frac{p_{max}}{n}) \rfloor$. On résout par programmation dynamique comme plus haut une instance modifiée du problème original : les tailles des n objets sont toujours a_i mais les valeurs sont $p'_i = \lfloor \frac{p_i}{2^t} \rfloor$. Soit $m_{AS}(x,r)$ la mesure de la solution ainsi obtenue.

6. Montrer que la complexité du schéma d'approximation est $O(\frac{r}{r-1} n^3)$.

7. Montrer que $m^*(x)/m_{AS}(x,r) \leq r$.
Indication : montrer que $\frac{m^*(x) - m_{AS}(x,r)}{m^*(x)} \leq \frac{n 2^t}{p_{max}}$.