TD10 : Encore des algorithmes d'approximation

1 Voyageur de commerce révisé

On suppose disposer d'un algorithme de calcul du couplage parfait minimal d'un graphe pondéré.

Question 1 : Soit G' un multi-graphe connexe. Montrer qu'il existe un cycle empruntant chaque arête exactement une fois si, et seulement si, le degré de tout sommet est pair.

Question 2 : Soit $G = (V, E, c)$ un graphe connexe pondéré vérifiant l'inégalité triangulaire (i.e. G est complet et pour tout $v, v', v'' \in V$, on a $c(v, v') \leq c(v, v') + c(v', v''))$. On note $T = (V, E')$ un arbre couvrant de poids minimal. On note $V_p \subseteq V$ l'ensemble des sommets de degrés impairs dans T. Montrer que $|V_p|$ est pair.

Question 3 : On considère le sous-graphe complet $G' = (V_p, (V_p)^2)$. Montrer qu'il admet un couplage parfait. On notera M un couplage parfait minimal.

Question 4 : On considère une solution optimale $\pi = v_1, \sigma_1, v_2, \sigma_2 \ldots v_{2m}, \sigma_{2m}, v_1$ au problème du voyageur de commerce du graphe G, en notant $\{v_1, \ldots, v_{2m}\} = V_p$. Montrer que $c(M) \leq c(\pi)/2$.

Question 5 : Soit le multigraphe G'' constitué de l'union de T et M. Montrer que tout sommet est de degré pair.

Question 6 : On considère un cycle de G'' empruntant toutes les arêtes, montrer que l'on peut en extraire un chemin hamiltonien de coût inférieur à $c(T) + c(M)$.

2 Bin packing

Une instance du problème BIN PACKING est donnée par n rationnels $x_1 \ldots x_n$ avec $\forall i\ 0 < x_i \leq 1$. On cherche le plus petit entier m tel que l'on peut ranger tous les éléments en m paquets de taille inférieure à 1, c'est-à-dire tel qu'il existe une fonction $f : [1, n] \to [1, m]$ avec $\forall b \in [1, m]\ \sum_{i \in f^{-1}(b)} x_i \leq 1$.

2.1 Algorithme glouton

On considère l'algorithme itératif plaçant chaque élément dans le premier paquet disponible dans lequel il peut rentrer (l'algorithme crée un nouveau paquet vide si l'élément ne peut rentrer nulle part). On note m le résultat et f l'assignation correspondante fournie par cet algorithme.

Question 1 : Montrer qu'au moins $m - 1$ paquets sont remplis (strictement) à plus de la moitié de leur capacité (c'est-à-dire qu'il existe $m - 1$ valeurs de b telles que $\sum_{i \in f^{-1}(b)} x_i > \frac{1}{2}$).

Question 2 : Montrer que toute solution au problème du bin packing est au moins égale à $S = \sum_i x_i$.

Question 3 : En déduire que l'algorithme glouton présente une garantie de performance de 2.

2.2 Algorithme glouton avec données ordonnées

On considère l'algorithme suivant : 1)On trie les éléments par ordre décroissant : $x_1 \geq x_2 \geq \ldots \geq x_n$. 2) On applique l'algorithme glouton précédent.

Question 4 : Supposons que si le paquet numéro b contient un élément $x_i > 1/2$ (ie $i \in f^{-1}(b)$). Montrer que la solution optimale m^* vérifie $m^* \geq b$.

Question 5 : On suppose dans cette question que le paquet b ne contient aucun élément de poids supérieur à $1/2$.

1. En déduire que les conteneurs $b' \in [b, m-1]$ contiennent au total plus de $2(m-b)$ éléments.
2. Sous-cas 1 : on suppose de plus que $b \leq 2(m-b)$. Montrer que $S > b-1$.
3. Sous-cas 2 : on suppose cette fois $b > 2(m-b)$. Montrer que $S > 2(m-b)$.

Question 6 : En déduire que l'algorithme possède une garantie de performance égale à $3/2$.

2.3 Non-approximabilité

On suppose dans cette section disposer d'un algorithme d'approximation résolvant le problème de bin packing en temps polynomial avec une garantie de performance égale à k.

On introduit le problème PARTITION défini comme suit :
— Entrée : $a_1 \dots a_n \in \mathbb{N}$
— Sortie : Existe-t-il $A \subseteq [1, n]$ tel que $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$?
On admet que ce problème est NP-complet.

Question 7 : Montrer que PARTITION n'admet pas de solution lorsqu'il existe i tel que $a_i > \sum_{j \neq i} a_j$.

Question 8 : Convertir toute instance I de PARTITION en une instance du problème de BIN PACKING I', telle que I admet une solution, si et seulement si, I' admet une solution à 2 paquets.

Question 9 : En déduire que $k \geq \frac{3}{2}$ (à moins que $P = NP$).

2.4 Schéma d'approximation asymptotique

Dans cet exercice, nous allons voir qu'il est toujours possible de concevoir un algorithme d'approximation dont la garantie de performance asymptotique reste bonne.

Question 10 : On fixe $\varepsilon > 0$ et $d \in \mathbb{N}$. Montrer qu'il existe un algorithme polynomial résolvant BIN PACKING dans le cas d'instances contenant moins de d valeurs différentes toutes supérieures à ε. (Indication : dénombrer le nombre de solutions).

Pour toute instance I de taille n, on note $m^*(I)$ la valeur optimale de BIN PACKING.

On suppose I triée par poids croissants, tous supérieurs à $\varepsilon > 0$. On fixe $P > 0$ et on note $Q = \left\lfloor \frac{n}{P} \right\rfloor$. On construit les instances H et J de taille n définies par

$$H = (\underbrace{\min(x_1 \dots x_Q) \dots}_{Q \text{ fois}}, \underbrace{\min(x_{Q+1} \dots x_{2Q}) \dots}_{Q \text{ fois}}, \dots \underbrace{\min(x_{\lfloor (n-1)/Q \rfloor Q+1} \dots x_n) \dots}_{n \mod Q \text{ fois}})$$

$$J = (\underbrace{\max(x_1 \dots x_Q) \dots}_{Q \text{ fois}}, \underbrace{\max(x_{Q+1} \dots x_{2Q}) \dots}_{Q \text{ fois}}, \dots \underbrace{\max(x_{\lfloor (n-1)/Q \rfloor Q+1} \dots x_n) \dots}_{n \mod Q \text{ fois}})$$

Intuitivement, on regroupe I par groupe de taille Q et on donne le même poids à tous les objets d'un même groupe : le poids le plus faible parmis les éléments du groupe (respectivement le plus élevé) dans le cas de H (respectivement J).

Question 11 : Montrer que $m^*(H) \leq m^*(I) \leq m^*(J)$.

Question 12 : Montrer que $m^*(J) \leq m^*(H) + Q \leq m^*(I) + Q$.

Question 13 : En déduire un algorithme d'approximation polynomiale à garantie de performance $1 + \varepsilon$ (dans le cas $\forall i \; x_i \geq \varepsilon > 0$).