TD 11 – Deux problèmes et leur algorithme d'approximation

Margot Catinaud margot.catinaud@lmf.cnrs.fr
Valentin Dardilhac valentin.dardilhac87@gmail.com

I Voyageur de commerce revisité

On suppose disposer d'un algorithme de calcul du couplage parfait minimal d'un graphe pondéré. Dans les questions suivantes, on va s'intéresser à la démonstration de la correction de l'algorithme donné dans les notes de cours (voir Section 4.4.3) pour ce problème dans le cas des graphes vérifiant l'inégalité triangulaire.

Question 1 *Soit $\mathcal{G}' = (\mathcal{V}, \mathcal{E})$ un multi-graphe connexe. Montrer qu'il existe un cycle eulérien*[1] *si et seulement si le degré*[2] *de tout sommet est pair.*

Soit $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ un graphe complet pondéré par une fonction de poids $\omega : \mathcal{E} \longrightarrow \mathbb{R}_+^*$ vérifiant l'inégalité triangulaire :

$$\forall\, s, t, u \in \mathcal{V},\ \omega(s,u) \leqslant \omega(s,t) + \omega(t,u).$$

De plus, on note $\mathcal{T} = (\mathcal{V}, \mathcal{E}')$ un arbre couvrant de poids minimal. On note $\mathcal{V}_{\mathsf{odd}} \subseteq \mathcal{V}$ l'ensemble des sommets de degré impair dans $\mathcal{T}$.

Question 2 *Montrer que* $\mathsf{Card}\big(\mathcal{V}_{\mathsf{odd}}\big)$ *est pair.*

Question 3 *On considère le sou-graphe complet $\mathcal{G}_{\mathsf{odd}} = (\mathcal{V}_{\mathsf{odd}}, \mathcal{V}_{\mathsf{odd}}^2)$ dans G. Montrer qu'il admet un couplage parfait. On notera $\mathcal{M}_{\mathsf{odd}}$ un couplage parfait de coût minimal dans $\mathcal{G}_{\mathsf{odd}}$.*

Question 4 *On considère une solution optimale $\pi = \Big(\big((v_i, \gamma_i)\big)_{i=1}^{2n}, v_1\Big)$, où $\mathcal{V}_{\mathsf{odd}} = \{v_i\}_{i=1}^{2n}$ et $\Gamma = (\sigma_i)_{i=1}^{2n}$ est un chemin. Montrer que tout couplage parfait $\mathcal{M}_{\mathsf{odd}}$ de coût minimal dans $\mathcal{G}_{\mathsf{odd}}$ vérifie l'inégalité*

$$\omega(\mathcal{M}_{\mathsf{odd}}) \leqslant \frac{1}{2}\omega(\pi).$$

Question 5 *Soit le multi-graphe $\mathcal{G}''$ constitué de l'union de l'arbre $\mathcal{T}$ et du couplage parfait $\mathcal{M}_{\mathsf{odd}}$,* i.e. *$\mathcal{G}'' = (\mathcal{V}, \mathcal{E}' \oplus \mathcal{M}_{\mathsf{odd}})$, où $\oplus$ est la somme de multi-ensembles (voir TD 9). Montrer que tout sommet de $\mathcal{V}$ est de degré pair dans $\mathcal{G}''$.*

Question 6 *On considère un cycle de $\mathcal{G}''$ empruntant toutes les arêtes. Montrer que l'on peut en extraire un chemin hamiltonien dans $\mathcal{G}$ de coût inférieur à $\omega(\mathcal{T}) + \omega(\mathcal{M}_{\mathsf{odd}})$.*

II *Bin packing*

Une instance du problème *bin packing* est donnée par n rationnels $(x_i)_{i=1}^n$ avec, pour tout $i \in [\![1; n]\!]$, $x_i \in \mathbb{Q} \cap]0, 1]$. On cherche le plus petit entier $p \in \mathbb{N}$ tel que l'on peut ranger les n rationnels en p paquets de somme inférieure à 1, *i.e.* tel qu'il existe une fonction que l'on suppose surjective

$$\varphi : [\![1; n]\!] \longrightarrow [\![1; p]\!],\ \text{avec} : \forall\, b \in [\![1; p]\!],\ \sum_{i \in \varphi^{-1}(b)} x_i \leqslant 1.$$

1. **Rappel :** un cycle eulérien est un chemin $\Gamma \overset{\text{def}}{=} (\gamma_i)_{i=0}^p$ de sommets de $\mathcal{V}$ tel que $p = \mathsf{Card}\big(\mathcal{E}\big)$ et $\Big\{(\gamma_{i-1}, \gamma_i) \mid i \in [\![1; p]\!]\Big\} = \mathcal{E}$: le chemin passe une et une seule fois par chaque arête de $\mathcal{E}$.

2. **Rappel :** Le degré $\mathsf{deg}(s)$ d'un sommet $s \in \mathcal{V}$ d'un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est le nombre de liens reliant ce sommet, *i.e.* $\mathsf{deg}(s) \overset{\text{def}}{=} \mathsf{Card}\Big(\Big\{(u, v) \in \mathcal{E} \mid u = s \vee v = s\Big\}\Big)$, avec les boucles comptées deux fois.

II.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 p le résultat et φ l'assignation correspondante fournie par l'algorithme.

Question 7 *Montrer qu'au moins $p-1$ paquets sont remplis (strictement) à plus de la moitié de leur capacité : il existe au moins $p-1$ valeurs de b telles que $\sum_{i\in\varphi^{-1}(b)} x_i > \frac{1}{2}$.*

Question 8 *Montrer que toute solution p du problème* bin packing *est au moins égale à $S = \sum_{i=1}^{p} x_i$.*

Question 9 *En déduire que l'algorithme glouton présente une garantie de performance de 2.*

II.2 Algorithme glouton avec données ordonnées

On considère l'algorithme suivant :

(1) On trie les éléments par ordre décroissant : $\forall\, i \in [\![1; n-1]\!],\ x_i \geqslant x_{i+1}$;

(2) On applique l'algorithme glouton précédent en considérant les rationnels dans cet ordre.

Question 10 *Supposons que le paquet numéro b contient un élément x_i,* i.e. *$i \in \varphi^{-1}(b)$, tel que $x_i > \frac{1}{2}$. Montrer que la solution optimale p^* vérifie $p^* \geqslant b$.*

Question 11 *On suppose dans cette question que le paquet b ne contient aucun élément de poids supérieur à $\frac{1}{2}$.*

1. *En déduire que les paquets suivants $b' \in [\![b; p-1]\!]$ contiennent au total plus de $2(p-b)$ éléments.*
2. ***Sous-cas 1 :** On suppose de plus que $b \leqslant 2(p-b)$. Montrer que $S > b-1$;*
3. ***Sous-cas 2 :** On suppose cette fois $b > 2(p-b)$. Montrer que $S > 2(p-b)$.*

Question 12 *En déduire que l'algorithme possède cette fois-ci une garantie de performance égale à $\frac{3}{2}$.*

II.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 δ. On introduit le problème partition défini comme suit :

- **Entrée :** $n \in \mathbb{N}^*$ entiers $(a_i)_{i=1}^n$;
- **Sortie :** Existe-t-il une partie $A \subseteq [\![1; n]\!]$ tel que $\sum_{i\in A} a_i = \sum_{i\notin A} a_i$?

On admet que ce problème est NP-complet.

Question 13 *Montrer que le problème* partition *n'admet pas de solution lorsque :*

$$\exists\, i \in [\![1; n]\!],\ a_i > \sum_{j=1,\, j\neq i}^{n} a_j.$$

Question 14 *Convertir toute instance $\mathcal{I}$ de* partition *en une instance $\mathcal{I}'$ du problème de* bin packing*, telle que $\mathcal{I}$ admet une solution si et seulement si $\mathcal{I}'$ admet une solution à 2 paquets.*

Question 15 *En déduire que $\delta \geqslant \frac{3}{2}$ (à moins que $\mathsf{P} = \mathsf{NP}$).*

II.4 Schéma d'approximation asymptotique

Dans cet section, on va montrer qu'il est toujours possible de concevoir un algorithme d'approximation pour *bin packing* dont la garantie de performance asymptotique reste "bonne".

Question 16 *On fixe $\varepsilon > 0$ et $d \in \mathbb{N}$. Montrer qu'il existe un algorithme polynomial résolvant* bin packing *de taille n dans le cas d'instances contenant moins de d valeurs différentes, toutes supérieures à ε. Pour cela, on pourra dénombrer le nombre de solutions.*

On suppose l'instance $\mathcal{I}$ triée par poids croissants, tous supérieurs à $\varepsilon > 0$. On fixe un entier $q \in \mathbb{N}^*$ et on note $Q \overset{\text{def}}{=} \left\lfloor \frac{n}{q} \right\rfloor$. On construit les instances $\mathcal{H}$ et $\mathcal{J}$ de *bin packing* de taille n définies comme suit :

$$\mathcal{H} \overset{\text{def}}{=} \left(\left(\min_{i=jQ+1}^{(j+1)Q} x_i \right)_{j=0}^{\lfloor (n-1)/Q \rfloor - 1}, \min_{i=\lfloor (n-1)/Q \rfloor Q+1}^{n} x_i \right).$$

$$\mathcal{J} \overset{\text{def}}{=} \left(\left(\max_{i=jQ+1}^{(j+1)Q} x_i \right)_{j=0}^{\lfloor (n-1)/Q \rfloor - 1}, \max_{i=\lfloor (n-1)/Q \rfloor Q+1}^{n} x_i \right).$$

Intuitivement, on regroupe $\mathcal{I}$ par groupe de taille Q et on donne le même poids à tous les objets d'un même groupe : le poids le plus faible (resp. le plus élevé) parmi les éléments du groupe dans le cas de $\mathcal{H}$ (resp. $\mathcal{J}$).

Devoir maison 11

1. *Montrer que $p^*(\mathcal{H}) \leqslant p^*(\mathcal{I}) \leqslant p^*(\mathcal{J})$;*
2. *Montrer que $p^*(\mathcal{J}) \leqslant p^*(\mathcal{H}) + Q \leqslant p^*(\mathcal{I}) + Q$;*
3. *En déduire un algorithme d'approximation polynomial à garantie de performance $1 + \varepsilon$ dans le cas :*

$$\forall\, i \in [\![1; n]\!],\ x_i \geqslant \varepsilon > 0.$$