TD 11 – Approximations et NP-complétude

Exercice 1. *Algorithmes d'approximation de BP*

On va s'intéresser dans cet exercice au problème *Bin Packing* :

Définition 1 (BP - Bin Packing)**.** *Soient n nombres rationnels (ou objets) $a_1, \dots, a_n$, avec $0 < a_i \leq 1$, pour $1 \leq i \leq n$. Peut-on les partitionner en k boîtes $B_1, \dots, B_k$ de capacité 1, i.e., pour chaque $1 \leq j \leq k, \sum_{i \in B_j} a_i \leq 1$?*

1. Prouvez que BP est NP-complet.

2. Prouvez que pour tout $\varepsilon > 0$, il n'existe pas de $\left(\frac{3}{2} - \varepsilon\right)$-approximation de BP si P $\neq$ NP.

On s'intéresse à présent à des algorithmes d'approximation de BP. On commence avec un algorithme glouton dans lequel on choisit des objets dans un ordre aléatoire, et, à chaque étape, on place l'objet soit dans la dernière boîte utilisée où il rentre (algorithme **next-fit**), soit dans la première boîte utilisée où il rentre (algorithme **first-fit**) ; sinon (i.e. l'objet ne rentre dans aucune boîte utilisée), on crée une nouvelle boîte et on place l'objet dedans.
Nous allons prouver que **next-fit** (et ainsi **first-fit**) est une 2-approximation de BP.

3. Prouvez que **next-fit** est une 2-approximation de BP.
Indice : Comparez le coût (i.e. le nombre de boîtes k) d'une solution avec $S = \sum_{i=1}^{n} a_i$, puis examinez le contenu de deux boîtes consécutives.

4. Prouvez que ce ratio d'approximation ne peut être amélioré pour cet algorithme.

Les algorithmes précédents peuvent être qualifés d'*online* car aucun tri des objets n'est fait, on peut donc les placer dans des boîtes quand ils arrivent, à la volée. Si on connaît tous les objets avant d'exécuter l'algorithme, on peut améliorer l'algorithme en commençant par trier les objets. Ces algorithmes sont qualifiés d'*offline*. L'algorithme **first-fit-dec** trie les objets par taille décroissante, et ensuite il applique la règle **first-fit** : l'objet est placé dans la première boîte utilisée où il rentre ; sinon, une nouvelle boîte est créée.

5. On va prouver que $C_{\textbf{first-fit-dec}} \leq \frac{3}{2} C_{opt} + 1$, où $C_{\textbf{first-fit-dec}}$ est le coût renvoyé par l'algorithme **first-fit-dec** et C_{opt} est le coût optimal. Pour cela, on sépare les a_i dans les 4 catégories suivantes :

$$A = \left\{ i : a_i > \frac{2}{3} \right\} \quad B = \left\{ i : \frac{2}{3} \geq a_i > \frac{1}{2} \right\} \quad C = \left\{ i : \frac{1}{2} \geq a_i > \frac{1}{3} \right\} \quad D = \left\{ i : \frac{1}{3} \geq a_i \right\}$$

   a. Montrez que si au moins une boîte contient uniquement des éléments de D, alors on a $C_{\textbf{first-fit-dec}} \leq \frac{3}{2} C_{opt} + 1$.

   b. Montrez que si tous les a_i sont dans A, B ou C, alors l'algorithme est optimal.

   c. Conclure.

6. On va finalement prouver que **First-fit-dec** est une $\frac{3}{2}$-approximation de BP. Pour cela, on examine le contenu de la boîte $j = \lceil \frac{2}{3} k \rceil$ où $k = C_{\textbf{first-fit-dec}}$:

   a. Montrez que si la boîte j contient un a_i tel que $a_i > \frac{1}{2}$, alors $C_{\text{opt}} \geq j$.

   b. Montrez que $2(k - j) + 1 \geq j - 1$, puis montrez que si on n'est pas dans le cas de la question **a.**, on a alors $S = \sum_i a_i > j - 1$.

   c. Conclure.

Exercice 2. *Diamètre et nuage de points*

On considère n points dans un espace métrique (les distances entre les points satisfont l'inégalité triangulaire). On veut partitionner les points en k groupes de manière à minimiser le plus grand diamètre d'un groupe. Le diamètre d'un groupe est la distance maximale entre deux points de ce groupe. Noter que n et k sont fixés dans l'énoncé du problème.

1. On suppose que le diamètre optimal d est connu. Trouver une 2-approximation pour le problème.
2. On considère l'algorithme qui, k fois, choisit comme "centre" le point à distance maximale de tous les centres déjà choisis, puis alloue chaque point au centre le plus proche. En faisant le lien avec la question précédente, montrer que cet algorithme est une 2-approximation pour le problème.