TD 11 – Approximations et NP-complétude

Exercice 1. *Approximation algorithms for BP*

We introduce a new classical problem that is the bin packing problem.

Définition 1 (BP - Bin Packing)**.** *Given n rational numbers (also called objects) $a_1, \dots, a_n$, with $0 < a_i \leq 1$, for $1 \leq i \leq n$, can we partition them in k bins $B_1, \dots, B_k$ of capacity 1, i.e., for each $1 \leq j \leq k$, $\sum_{i \in B_j} a_i \leq 1$?*

1. Prove that BP is NP-complete.
2. Prove that for all $\epsilon > 0$, there does not exist any $\left(\frac{3}{2} - \epsilon\right)$-approximation algorithm for BP unless P = NP.

We look now at approximation algorithms for BP. We start with a simple greedy algorithm in which we select objects in a random order, and, at each step, we place the object either in the last used bin where it fits (**next-fit** algorithm) or in the first used bin where it fits (**first-fit** algorithm) ; otherwise (i.e., the object is not fitting in any used bin), we create a new bin and place the object in this new bin.
We prove below that **next-fit** (and, hence, **first-fit**) is a 2-approximation algorithm for the BP problem.

3. Prove that **Next-fit** is a 2-approximation algorithm for BP.
   Hint : Compare the cost of a solution with $A = \sum_{i=1}^{n} a_i$ and think about the content of consecutive bins.
4. Show this approximation ratio is tight.

The previous algorithms can be qualified as *online* algorithms because no sorting is done on the objects, and we can pack them in the bins when they arrive, on the fly. If we have the knowledge of all objects before executing the algorithm, we can refine the algorithm by sorting the objects beforehand. Such algorithms are called *offline* algorithms. The **first-fit-dec** algorithm sorts the objects by nonincreasing size (**dec** stands for decreasing), and then it applies the **first-fit** rule : The object is placed in the first used bin in which it fits ; otherwise, a new bin is created.

5. Prove that $C_{\textbf{first-fit-dec}} \leq \frac{3}{2} C_{opt} + 1$, where $C_{\textbf{first-fit-dec}}$ is the cost returned by the **first-fit-dec** algorithm, and C_{opt} is the optimal cost. To do so, you can split the a_i into categories according to the values $\frac{1}{3}$, $\frac{1}{2}$ and $\frac{2}{3}$ and examine how they form bins.

Remarque 1. *Note that this is not an approximation algorithm as defined in the course, because of the "+1" in the expression, which corresponds to one extra bin that the* **first-fit-dec** *algorithm may use. This is rather an* asymptotic *approximation algorithm, which is similar to an A(F)PTAS scheme. Indeed, the constant 1 is independent of the problem size, and the algorithm is asymptotically a $\frac{3}{2}$-approximation.*
It is also possible to prove that $C_{\textbf{first-fit-dec}} \leq \frac{11}{9} C_{opt} + 1$. The idea of the proof is similar, but the algorithm turns out to be much more complex.

Without allowing an extra bin, we can finally prove that **first-fit-dec** is a $\frac{3}{2}$-approximation algorithm.

6. Show that **First-fit-dec** is a $\frac{3}{2}$-approximation algorithm for the bin packing problem. To do so, you can examine the content of the bin created $\frac{2}{3}$ along the way.

Exercice 2.

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.