TD du TP 12

On s'intéresse au problème d'optimisation BIN PACKING défini comme suit :

BIN PACKING

Entrée : n objets de volume $v_1, ..., v_n$ dans $\mathbb{N}$ et un volume maximal $V \in \mathbb{N}^*$

Sortie : Un entier m et une fonction $f : [\![|1, n]\!] \to [\![1, m]\!]$ tels que $\forall i \in [\![1, m]\!], \sum_{f(v_j)=i} v_j \leq V$ qui minimise m.

Exercice 1 : NP-Complétude

On admet que le problème suivant est NP-Complet :

PARTITION

Entrée : $E = \{c_1, ..., c_n\} \in \mathbb{N}^n$

Sortie : Vrai s'il existe un sous-ensemble $C \subseteq [\![1, n]\!]$ tel que $\sum_{i \in S} c_i = \sum_{i \notin S} c_i$ et faux sinon.

1) Décrire le problème de décision BPD associé au problème d'optimisation BIN PACKING.

2) Montrer que BPD $\in \mathrm{NP}$.

3) Étant donnée une instance $c_1, ..., c_n$ de PARTITION, on construit l'instance suivante de BPD : les volumes sont $2c_1, ...2c_n$, le volume total est $V = \sum_{i=1}^{n} c_i$ et le seuil (donc le nombre maximum de boîtes) est égal à 2. Montrer que cette transformation prouve que PARTITION se réduit polynomialement à BPD et conclure.

Exercice 2 : 2-approximation

On cherche à approcher les solutions pour BIN PACKING via l'algorithme *next-fit*. Son principe est le suivant : il maintient à jour une boîte courante. Pour chaque objet, *next-fit* détermine s'il peut rentrer dans la boîte courante : si oui, il l'y place, si non, il ferme définitivement la boîte courante, ouvre une nouvelle boîte qui devient la nouvelle boîte courante et y place l'objet. On note m le nombre de boîtes déterminé via la stratégie *next-fit* sur une instance donnée de BIN PACKING et m^* le nombre optimal de boîtes pour cette même instance.

4) Montrer que la somme des volumes occupés par des objets de deux boîtes consécutives selon la stratégie *next-fit* est strictement supérieure à V.

5) Montrer que $m^* V \geq \sum_{i=1}^{n} v_i$.

6) En déduire que *next-fit* est une 2-approximation de BIN PACKING.

7) Déterminer la complexité de *next-fit* et commenter.

8) Existe-t-il un $\alpha < 2$ tel que *next-fit* soit une α-approximation de BIN PACKING ?

Soit à présent $\varepsilon > 0$ et supposons qu'il existe un algorithme polynomial qui soit une $(\frac{3}{2} - \varepsilon)$-approximation de BIN PACKING.

9) En s'inspirant de la réduction de la question 3, montrer que dans ces conditions il existe un algorithme polynomial permettant de résoudre PARTITION. Que peut-on en conclure ?

Exercice 3 : NP-Complétude de PARTITION (bonus)

On admet temporairement que le problème suivant est NP-Complet (on le montrera dans un : deuxième temps) :

SUBSET SUM

Entrée : Un multi-ensemble A d'entiers naturels et un entier s

Sortie : Vrai s'il existe $B \subseteq A$ tel que $\sum_{x \in B} x = s$ et faux sinon.

10) À l'aide d'une réduction de SUBSET SUM à PARTITION (ajoutant deux éléments bien choisis à une instance de SUBSET SUM), montrer que PARTITION est NP-Complet.

On montre maintenant que SUBSET SUM est NP-Complet.

11) Montrer que SUBSET SUM est dans NP.

12) Donner un algorithme permettant de résoudre le problème en un temps polynomial en $n + s$, où n est le cardinal de A. Pourquoi cela ne montre-t-il pas que SUBSET SUM appartient à P ?

On va montrer que SUBSET SUM est NP-Complet par réduction de 3-SAT. On considère pour cela une instance φ de 3-SAT, constituée de m clauses $C_0, ..., C_{m-1}$ utilisant n variables $x_0, ..., x_{n-1}$.

Nous allons travailler avec des nombres écrits en base 4, en donnant leurs chiffres.

- Pour chaque variable x_i, on définit deux nombres t_i et f_i ayant $n + m$ chiffres chacun.
  - Le i-ième chiffre (chiffre de poids 4^i) de t_i et de f_i vaut 1.
  - Pour $0 \leq j < m$, le $j + n$-ième chiffre de t_i vaut 1 si x_i apparaît (positivement) dans la clause C_j.
  - Pour $0 \leq j < m$, le $j + n$-ième chiffre de f_i vaut 1 si $\neg x_i$ apparaît dans la clause C_j.
  - Tous les autres chiffres de t_i et de f_i valent 0.
- Pour chaque clause C_j, on définit deux nombres a_j et b_j, de $n + m$ chiffres.
  - Le $j + n$-ième chiffre de a_j et de b_j valent 1.
  - Tous leurs autres chiffres sont nuls.

  On a donc $a_j = b_j$.

On définit A comme le multi-ensemble constitué des t_i et f_i pour $0 \leq i < n$ et des a_j et b_j pour $0 \leq j < m$.

13) Pour $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)$, déterminer les nombres t_i, f_i, a_j et b_j (on les écrira en base 4).

14) Montrer que l'on peut choisir la cible de s de manière à ce que l'instance (A, s) soit positive si et seulement si φ est satisfiable.

15) Conclure.