École polytechnique — Majeure informatique

Promotion 2001

Cours de conception et analyse d'algorithmes

PC 8 – revisions

1. $\mathcal{NP}$ complétude du BIN PACKING

On se donne un ensemble de n objets de tailles respectives s_i, avec $0 < s_i < 1$, et un entier k. Le problème du BIN PACKING consiste à décider s'il est possible de ranger tous les objets dans au plus k boîtes de taille 1.

Montrer, par une réduction à partir de SUBSET-SUM, que ce problème est NP-complet.

2. Programmation dynamique.

On considère un mètre de menuisier, dont les n segments ont longueurs $\ell_1, \ell_2, \ldots, \ell_n$, qui dans cet exercice peuvent être des entiers quelconques (contrairement aux mètres habituels, dont tous les segments ont même longueur), mais tels que $\ell_1 + \ell_2 + \ldots + \ell_n = L \leq 10n$.

On souhaite ranger ce mètre dans une boîte de longueur entière k. Pour cela, il faut replier le mètre de façon que sa longueur totale, une fois replié, soit inférieure ou égale à k. Pour tout i, le i-ième segment peut être replié dans le même sens que le segment $i-1$ ou dans le sens contraire.

Proposer un algorithme basé sur la programmation dynamique et qui permette, étant donné k, n, et $\ell_1, \ldots, \ell_n$ tels que $\sum_i \ell_i \leq 10n$, de *décider* en temps polynomial s'il est possible de replier le mètre de façon à ce qu'il tienne dans la boîte. On souhaite ici que l'algorithme soit suffisamment détaillé (pseudo-code par exemple).

Exercice 3. MaxCut, Approximation

Dans la détermination de l'energie minimale du modèle d'Ising on est amené à résoudre le problème *MAXCUT* suivant :

Soit $G = (X, E)$ un graphe symétrique, déterminer une partition de X en

$$X = X_1 \cup X_2, \quad X_1 \cap X_2 = \emptyset$$

telle que la coupure $E(X_1, X_2)$ formée des arêtes ayant une extrémité dans X_1 et l'autre dans X_2 soit de taillle maximum.

On se propose de donner un algorithme approché pour cette question puis de montrer que le problème exact est NP-complet.

On considère l'algorithme qui consiste à initialiser X_1, X_2 de façon quelconque puis à chercher itérativement s'il existe un sommet dont le passage de l'une des classes à l'autre augmente la taille de la coupure. Tant que l'on trouve un tel sommet on effectue ce passage, puis une fois que l'on en trouve plus, on déclare que la coupure est maximale.

i) Soit X_1, X_2 la partition donnée par l'algorithme ci-dessus et soit Y_1, Y_2 la partition optimale. On note $V_1 = X_1 \cap Y_1$, $V_2 = X_1 \cap Y_2$, $V_3 = X_2 \cap Y_1$ et $V_4 = X_2 \cap Y_2$. Enfin pour tout couple i, j on note $m_{i,j}$ le nombre d'arêtes ayant une extrémité dans V_i et l'autre dans V_j. Montrer que pour tout sommet de V_1 le nombre de ses voisins dans V_2 est infe'rieur ou égal au nombre de ses voisins dans $V_3 \cup V_4$.

En déduire que

$$m_{1,2} \leq m_{1,3} + m_{1,4}$$

puis :

$$m_{1,2} + m_{3,4} + m_{1,4} + m_{2,3} \leq 2(m_{1,3} + m_{1,4} + m_{2,3} + m_{2,4})$$

ii) Déduire de ce qui précède que l'algorithme donne une coupure dont la taille est minorée par la moitié de la taille de la coupure maximum.

Soit $U = \wedge_{i=1,p} u_i$ un ensemble de clauses sur n variables x_j (où $u_i = t_i \vee z_i$). On suppose qu'elles sont toutes distinctes et qu'elles contiennent chacune deux variables différentes. On construit le graphe $G(U)$ ayant pour ensemble de sommets $X = \{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n, c_1, c_2, \ldots c_{2p}\}$ et dont les arêtes sont les paires suivantes associées aux clauses $u_i = t_i \vee z_i$

$$\{c_{2i-1}, t_i\}, \{c_{2i}, z_i\}, \{t_i, z_i\}$$

iii) Dessiner le graphe associé à la formule

$$(x_1 \vee \overline{x}_2) \wedge (x_3 \vee x_4) \wedge (x_2 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_4)$$

iv) Montrer que s'il existe k clauses satisfaisables dans U alors le graphe $G(U)$ admet une partition X_1, X_2 telle que X_1 contienne $c_1, c_2, \ldots c_{2p}$ et donnant une coupure de taille supérieure à $2k$.

v) Montrer que le probleme *MAX-2SAT* se réduit polynomialement au problème *MAXCUT-COMP* suivant :

Soit $G = (X, E)$ un graphe, K un entier, Y un sous ensemble de X et q paires a_i, b_i ($i = 1, \ldots, q$) de sommets de G disjointes ; déterminer s'i l existe une partition X_1, X_2 de X telle que :

– $Y \subset X_1$
– Pour tout $i = 1, \ldots, q$ $\{a_i, b_i\} \cap X_1 \neq \emptyset$ et $\{a_i, b_i\} \cap X_2 \neq \emptyset$
– La coupure $E(X_1, X_2)$ est de taille supérieure ou égale à K.

On admettra que MAXCUT-COMP se réduit polynomialement à *MAXCUT*, en déduire que *MAXCUT* est NP-complet.

Exercice 4. Taux de change On étudie dans cet exercice un problème lié aux taux de changes entre monnaies. On supposera que ces taux n'évoluent pas dans le temps et on négligera les frais de courtage pour les opérations de change.

On suppose données n monnaies numérotées de 1 à n, et les taux de change correspondants t_{ij}, c'est à dire que x unités de la monnaie i donnent $t_{ij}x$ unités de la monnaie j. Comme ces taux sont donnés avec une précision limitée, il est extrêmement improbable que $t_{ij}t_{ji}$ soit exactement égal à 1. En pratique, cette quantité est toujours au plus égale à 1. Cela dit, il reste possible qu'il existe une suite de monnaies $1, i_1, \ldots, i_k, 1$ telle que $t_{1i_1}t_{i_1i_2} \ldots t_{i_{k-1}i_k}t_{i_k1} \geq 1$, résultant en un profit.

1. Formuler le problème de trouver une suite d'opérations partant de la monnaie 1 et y revenant, produisant un profit optimal, en terme de recherche de cycles simples (i.e., passant une fois au plus par chaque sommet) dotés d'une certaine propriété dans un graphe valué que l'on précisera.

2. Montrer que le problème du cycle hamiltonien (décider si dans un graphe il existe un cycle passant une fois et une seule par chaque sommet) se réduit au problème de la recherche du plus long cycle simple dans un graphe valué. Que peut-on en déduire ?

3. Proposer une formalisation simple du problème du **1.** comme un problème de flot unitaire de coût minimal, dans un graphe dont les capacités sont unitaires, et où chaque arête a un certain coût par unité de flot. Cette formalisation fournit-elle un résultat optimal (on expliquera ce qui risque de se produire plutôt que de chercher à donner un contre-exemple) ?

Ce type de problème de min-cost flow peut être résolu par une variante de l'algorithme de Ford-Fulkerson, où l'on recherche dans le graphe résiduel non un chemin de la source au puits, mais un cycle de longueur négative.