Optimisation combinatoire
le 13 juillet 2005, durée 2 heures
Maitrise MIAGE

Exercice 0.1 Deux sacs à dos (16 points)

On considère un problème de sac à dos dans lequel il s'agit de remplir DEUX sacs, de poids maximum P et Q respectivement. On dispose de n objets, on note p_i le poids de l'objet i, et w_i son utilité. Il s'agit de déterminer pour chaque objet s'il est pris, et le cas échéant dans quel sac il se trouve, de sorte que la somme des utilités des objets pris soit maximale.

On notera Π une instance de ce problème

1 Modéliser le problème à l'aide un programme linéaire à variables bivalentes. On pourra définir les variables bivalentes suivantes : $x_i = 1$ si l'objet i se trouve dans le sac 1, 0 sinon. $y_i = 1$ si l'objet i se trouve dans le sac 2, 0 sinon.

2 Soit une instance du problème notée Π. Notons S^* la solution optimale de Π et notons $w(S^*)$ l'utilité des deux sacs de cette solution.

On considère le problème de sac à dos Π' constitué des mêmes objets, mais d'un seul sac de poids maximum $P1 + P2$. Soit X^* la solution optimale de ce problème, et notons $w(X^*)$ son utilité.

Montrer que $w(S^*) \leq w(X^*)$

3 Rappeler l'algorithme permettant de résoudre la relaxation continue de Π' (lorsque les objets peuvent être découpés). Notons $\hat{X}$ la solution optimals continue associée, et $w(\hat{X})$ son utilité. Montrer que $w(S^*) \leq w(\hat{X})$

4 On cherche à définir une méthode arborescente pour le problème Π. Pour cela, on considère qu'un noeud de l'arborescence, S, sera défini par trois ensembles d'objets $C_1(S)$, l'ensemble des objets choisis pour le sac 1, $C_2(S)$, l'ensemble des objets choisis pour le sac 2 et $R(S)$, l'ensemble des objets rejetés.

Proposer, en vous basant sur les questions précédentes, deux algorithmes permettant de définir une évaluation par excès du noeud S.

5 De même, proposer une heuristique permettant de définir une solution associée au noeud S, et donc une évaluation par défaut.

6 Le principe de séparation adopté consiste, au noeud S, à choisir un objet i non encore affecté dans S, et créer au plus trois fils de S : S_1,S_2,S_3 dont les ensembles caractéristiques sont définis comme suit :

$C_1(S_1) = C(S) \cup \{i\}, C_2(S_1) = C_2(S), R(S_1) = R(S)$ (à condition que le poids l'ensemble $C_1(S_1)$ ne dépasse pas P_1, ou sinon le noeud n'est pas créé).

$C_1(S_2) = C(S), C_2(S_2) = C_2(S) \cup \{i\}, R(S_2) = R(S)$ (à condition que le poids l'ensemble $C_2(S_2)$ ne dépasse pas P_2, ou sinon le noeud n'est pas créé).

$C_1(S_3) = C(S), C_2(S_3) = C_2(S), R(S_3) = R(S) \cup \{i\}$

Appliquer cette méthode arborescente à l'exemple suivant :

$P_1 = 4, P_2 = 3$

i	1	2	3	4	5	6
w_i	18	5	10	8	15	7
p_i	3	1	2	2	4	2

NB : Si vous manquez de temps, ne développez l'arborescence que partiellement (environ 10 noeuds), en indiquant surtout les décisions que vous pouvez prendre grâce aux évaluations.

7 On définit $F_k(u, v)$ la valeur maximale de l'utilité de deux sacs de poids respectivement inférieurs à u et v ne comportant que des objets parmi $\{k, \ldots, n\}$. Exprimer la solution recherchée du problème Π par rapport à cette notation.

8 Définir les valeurs de $F_n(u, v)$ pour les différentes valeurs de u et de v.

9 Etablir l'équation de récurrence qui lie les $F_k(u, v)$.

10 En déduire un algorithme de programmation dynamique permettant de résoudre Π, dont vous indiquerez la complexité.

Exercice 0.2 Algorithmes approchés (4 points)

Cet exercice a pour objectif d'étudier le problème du recouvrement minimum d'un graphe par un ensemble de sommets. Soit G=(S,E) un graphe non orienté. Un recouvrement de G est un sous-ensemble X de S tel que toute arête de E a au moins l'une de ses extrémités dans X. Etant donné un graphe G, on cherche un recouvrement de cardinalité minimale.

On rappelle que si l'on associe à chaque sommet i une variable bivalente x_i qui vaut 1 si X contient i et 0 sinon, le problème se modélise grâce au programme linéaire en nombres entiers suivant :

$$\begin{cases} Min & \sum_{i=1}^{n} & x_i \\ \forall (i,j) & \in E & x_i + x_j \geq 1 \\ \forall i & \in S & x_i \in \{0,1\} \end{cases}$$

Notons $u_i, i \in \{1, \ldots, n\}$ la solution optimale de la relaxation continue de ce programme linéaire (c'est à dire en considérant que les variables $x_i \in$

$[0, 1]$). On définit

$$\hat{x}_i = \begin{cases} 1 & si \quad\quad u_i \geq 1/2 \\ 0 & sinon \end{cases}$$

1 Montrer que $\hat{x}_i, i \in \{1, \ldots, n\}$ est une solution réalisable du programme linéaire en nombre entiers, et définit donc un recouvrement du graphe.

2 Déduire de l'inégalité $\hat{x}_i <= 2 * u_i$ une garantie de performance relative de 2 de la solution ainsi construite.