Examen partiel d'optimisation combinatoire
Durée 2h, le 27 juin 2006
Le barème est indicatif

Exercice 1 Ordonnancement (8 points)

On se donne un ensemble de n tâches à effectuer sur une machine. On connaît pour chaque tâche i sa durée p_i et son échéance d_i. Lorsqu'elle est en retard cette tâche à un coût c_i. On s'intéresse à la détermination d'un ordonnancement des tâches sur la machine qui soit de coût minimum (le coût d'un ordonnancement étant la somme des coûts des tâches en retard).

On admettra que comme dans le problème étudié en cours, il existe un ordonnancement optimal où toutes les tâches à l'heure sont placées avant les tâches en retard, et où toutes les tâches à l'heure sont exécutées dans l'ordre croissant de leurs dates d'échéance. On supposera dans la suite que $d_1 \leq d_2 \leq \ldots \leq d_n$ et on appellera séquence à l'heure d'un ensemble de tâche T tout sous-ensemble de tâches S de T, qui, placées dans l'ordre croissant, sont à l'heure. La durée de S est la somme des durées des tâches qui la composent. Le coût de S est la somme des coûts des tâches de $T - S$.

Notons $F_k(E)$ le coût minimal d'une séquence à l'heure de l'ensemble de tâches $\{k \ldots n\}$ commençant à la date E.

1 Que vaut $F_n(E)$ pour les différentes valeurs de E ?

2 Montrer que si $E+p_k \leq d_k$ alors $F_k(E) = \min(c_k+F_{k+1}(E), F_{k+1}(E+p_k))$ sinon $F_k(E) = c_k + F_{k+1}(E)$

3 En déduire un algorithme permettant de résoudre le problème dont vous indiquerez la complexité

4 Calculer la solution du problème suivant :

i	1	2	3	4	5	6
d_i	4	4	4	5	6	8
p_i	3	1	2	2	4	6
c_i	10	5	10	8	12	18

Exercice 2 Méthode arborescente (5 points)

Développez la dernière méthode arborescente vue en cours pour le problème de sac à dos ci-dessous avec $P = 10$. On précisera les différentes étapes de construction de l'arborescence ainsi que les décisions prises (troncature, séparation,...etc).

i	1	2	3	4	5	6	7	8	9
w_i	14	18	10	5	16	18	9	4	3
p_i	2	3	2	1	4	5	3	2	3

Exercice 3 Recouvrement et stable (7 points)

On rappelle qu'un recouvrement d'un graphe non orienté G est un sous-ensemble X des sommets de ce graphe, tel que toute arête de G a au moins une extrémité dans X. On s'intéresse à la construction d'un recouvrement de cardinalité minimale. On notera X^* un recouvrement optimal.

1 Rappeler la modélisation du problème.

2 Dessiner un graphe non orienté à 10 sommets et donnez-en un recouvrement.

3 Soit U un ensemble d'arêtes de G qui n'ont aucune extrémité commune. Montrer que $|X^*| \geq |U|$

L'algorithme suivant construit un sous-ensemble X : au départ X est vide, et G' est une copie de G.

Tant qu'il existe une arête dans G' :
a)-Choisir une arête $a = \{i, j\}$ et ajouter les sommets i et j à X
b)-supprimer de G' l'arête a et toutes les arêtes adjacentes à i et j

4 Appliquer cet algorithme à votre graphe.

5 Notons $\hat{U}$ l'ensemble des arêtes de G sélectionnées lors de l'étape a) des itérations de la boucle. Montrer que les arêtes de $\hat{U}$ n'ont pas d'extrémité commune.

6 En déduire que cet algorithme fournit une 2-approximation relative de X^*.

On appelle *stable* d'un graphe non orienté un sous-ensemble de sommets Y tel qu'il n'existe pas d'arêtes entre deux sommets de Y.

7 Modéliser le problème de la recherche d'un stable de cardinalité maximum dans un graphe.