Optimisation combinatoire

Interrogation du 13 mars – apprentis - durée 40 minutes – répondre sur le sujet (recto-verso)

Nom :	Prénom :

Exercice 1 Ordonnancement sur deux machines.

Modéliser le problème suivant :

On considère un ensemble de n tâches à faire. Les tâches peuvent être faites sur l'une ou l'autre de deux machines différentes. Chaque tâche i a une durée a_i si elle est faite sur la première machine, b_i sur la seconde. Chaque machine n'exécute qu'une seule tâche à la fois et réalise donc toutes les tâches qui lui sont affectées en séquence (dans n'importe quel ordre) sans s'arrêter. La durée totale de fonctionnement d'une machine ne doit pas dépasser la durée ouvrable de la journée O. Les deux machines fonctionnent en parallèle.

Malheureusement, il n'est pas toujours possible de faire toutes les tâches dans la durée impartie, et l'on est conduit à rejeter certaines tâches. Chaque tâche i, si on la rejette, engendre une pénalité notée p_i. On cherche donc à déterminer pour chaque tâche si on l'exécute ou non, et si oui sur quelle machine, de sorte que chaque machine fonctionne sur une durée qui n'excède pas O et que la somme des pénalités liées aux taches non-exécutées soit minimale.

Inconnues : pour chaque tâche i, x_i inconnue binaire qui vaut 1 lorsque la tâche i est faite par la machine 1, 0 sinon, et y_i qui vaut 1 lorsque la tâche i est faite sur la machine 2 et 0 sinon

Contraintes :

Une tâche ne peut pas être faite sur les deux machines en même temps :pour tout i, $x_i+y_i\leq 1$

La charge de la machine 1 ne dépasse pas O : $\sum_{i=1}^{n} a_i x_i \leq O$

La charge de la machine 2 ne dépasse pas O : $\sum_{i=1}^{n} b_i y_i \leq O$

Pour tout i, x_i , $y_i \in \{0,1\}$

Objectif :

Lorsqu'une tâche i n'est pas exécutée, x_i+y_i vaut 0, et donc $1-(x_i+y_i)$ vaut 1.

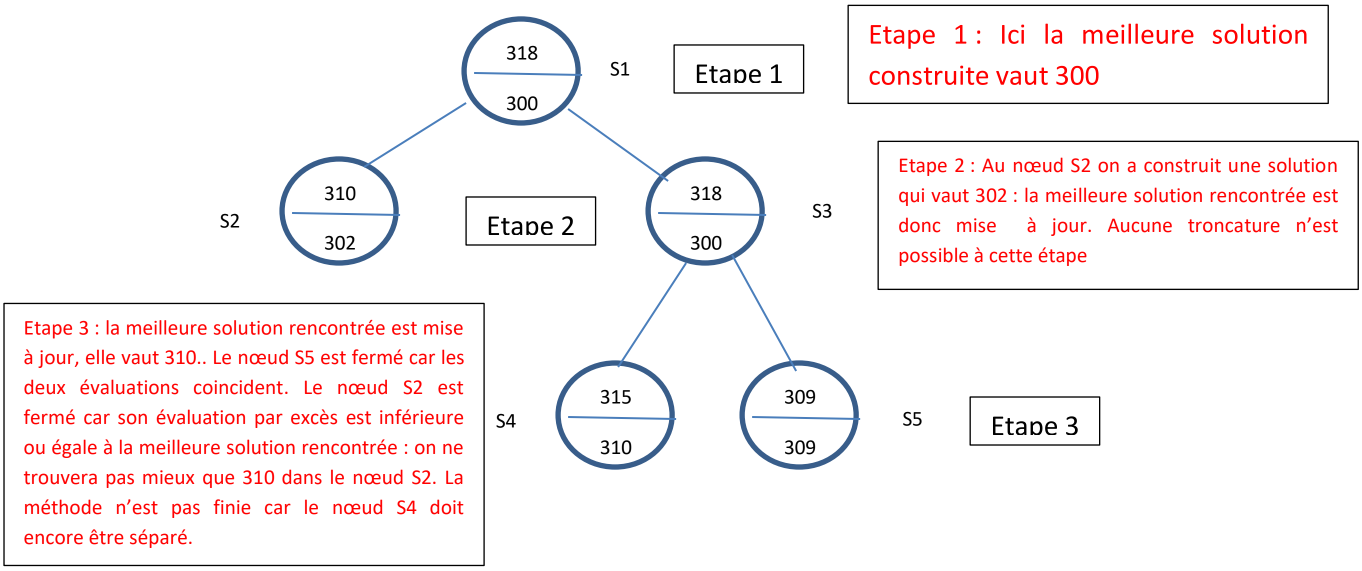
On doit donc minimiser :

$\sum_{i=1}^{n} p_i(1-(x_i+y_i))$

Exercice 2. Méthodes arborescentes

Question 1 : On considère l'arbre suivant correspondant à l'application d'une méthode arborescente vue en cours pour un problème de sac à dos. Préciser à chaque étape de la méthode la valeur de la meilleure solution rencontrée, et les troncatures qui peuvent être faites. Peut-on déduire ici la valeur de la solution optimale du problème (justifier) ?

Il s'agit ici du problème de sac à dos vu en cours : Maximiser la somme des utilités des objets du sac sachant que le sac ne pèse pas plus que P. Comme c'est un problème de maximisation, l'évaluation par excès est obtenue par relaxation, et l'évaluation par défaut au moyen d'un algorithme qui construit un solution particulière du nœud.



Question 2 : Rappeler les algorithmes permettant de calculer les évaluations par défaut et par excès d'un nœud.

Evaluation par excès : tri des objets dont le statut n'est pas encore défini par wi/pi décroissant, puis remplissage du sac jusqu'à ce qu'un objet dépasse, cet objet est découpé pour rentrer exactement dans le sac. L'évaluation est la somme des utilités des objets et morceaux d'objets dans le sac.

Evaluation par défaut : tri des objets dont le statut n'est pas encore défini par wi/pi décroissant puis remplissage du sac en examinant les objets dans cet ordre, en jetant les objets qui ne rentrent pas.