Procédures arborescentes

Problème d'affectation

On a n tâches à affecter à n ouvriers avec une unique tâche par ouvrier. Le problème est spécifié par la donnée d'une matrice D de n lignes et n colonnes. Les lignes représentent les ouvriers, les colonnes les tâches. D_{ij} = coût d'affectation de l'ouvrier i à la tâche j. Le problème est donc d'affecter les tâches aux ouvriers de façon à minimiser les coûts engendrés. On peut modéliser ce problème par un programme linéaire puis résoudre ce programme linéaire à l'aide d'un « solveur ». Ici, on va le résoudre à l'aide d'une procédure arborescente définie de la façon suivante :

- Séparation n-aire : au niveau i, on fait toutes les affectations de tâches (non encore affectées) à l'ouvrier i. Cela crée n sommets (au plus)
- Evaluation (borne inférieure) : chaque tâche doit de toute façon être exécutée. Le coût minimum pour une tâche j est donc le coefficient D_{ij} minimum dans la colonne j de D. Ceci étant vrai pour toute tâche j, il suffit pour avoir une borne inférieure du coût total, d'ajouter les minimums de chaque colonne relative à chaque tâche. Cette borne inférieure se calcule donc facilement (algorithme polynomial) en chaque nœud de l'arborescence de recherche.
- Pour un sous-problème donné (en un nœud de l'arborescence de recherche) lorsque les minimums de chaque colonne, sont sur des lignes distinctes, on a dans ce cas une solution optimale pour le sous-problème traité et donc une solution (une borne supérieure) du problème. Utiliser la borne supérieure et la borne inférieure pour élaguer l'arbre de recherche.
- Faire une recherche en meilleur d'abord : explorer le sommet pendant de l'arborescence de recherche de meilleure borne inférieure c'est-à-dire la plus petite.

Traiter le cas suivant :

D (coûts)	Tâche 1	Tâche 2	Tâche 3	Tâche 4
Ouvrier 1	8	3	1	5
Ouvrier 2	11	7	1	6
Ouvrier 3	7	8	6	8
Ouvrier 4	11	6	4	9

Minimisation de la somme des retards dans un problème d'ordonnancement

Déterminer un ordre de passage de n tâches indépendantes, de durées $d_1,d_2,\ldots,d_n$ et de dates d'échéance $e_1,e_2,\ldots,e_n$, (e_i=date à laquelle la tâche i doit être terminée) de façon à minimiser la somme des retards où le retard d'une tâche i est défini par $R_i=\max(0, c_i-e_i)$, c_i étant la date de fin de la tâche i.

Le problème est donc de déterminer un ordre de passage des tâches et donc des dates de fin c_i des tâches i=1 à n de sorte que $\sum_{i=1}^{n} R_i$ est minimum.

Exemple :

n=5.

Tâches	1	2	3	4	5
Durée d	4	3	7	2	2
Echéances e	5	6	8	8	17

1-Donner le retard pour l'ordonnancement 1 2 3 4 5

2-Résoudre le problème par une procédure arborescente de caractéristiques suivantes :

- Séparation n-aire : au niveau 1, on fixe la tâche qui passe en dernier (n cas possibles), au niveau 2 la tâche qui passe en avant dernier (n-1 cas possibles) etc …
- Evaluation : somme des retards des tâches fixées.
- Borne supérieure : valeur d'une solution obtenue quand on arrive à une feuille de l'arborescence de recherche.
- Exploration en meilleur d'abord (sommet de l'arborescence de recherche d'évaluation minimum).