Nom-prénom : Nbre de feuilles suppl. :

Examen de recherche opérationnelle

Durée=1h30 / Documents autorisés / Calculatrice interdite

1 Programmation linéaire en nombres entiers

Un charpentier fabrique des tables et des chaises dans son atelier pendant la semaine, qui sont vendues dans un dépot-vente pendant le week-end. Chaque table peut être vendue pour un profit de 30 euros et chaque chaise pour un profit de 10 euros. Le charpentier peut travailler 40 h par semaine. Il lui faut 6h pour fabriquer une table et 3h pour faire une chaise. La demande des clients exige qu'il fasse au moins 3 fois plus de chaises que de tables. Les tables nécessitent 4 fois plus d'espace en stockage que les chaises. La cave de l'atelier est telle qu'on peut y stocker l'équivalent de 4 tables par semaine.

1.1 PL

Dans un premier temps on suppose que le charpentier peut commencer un mobilier une semaine et le terminer la semaine suivante. Par exemple, fabriquer 1,5 chaises par semaine.

1. Modéliser ce problème sous forme d'un programme linéaire en spécifiant en "français" puis mathématiquement : les variables de décision, leur domaine de définition, la fonction-objectif et les contraintes.

2. Représenter le problème en 2D, puis le résoudre graphiquement

3. Appliquer la méthode du simplexe pour résoudre ce problème.

4. Préciser à quoi correspondent les solutions de bases successives par rapport au graphique 2D précédent (représenter sur le graphique si besoin)

5. Donner les intervalles de variation des coefficients de la function-objectif pour lesquels la base correspondant à la solution de la question 3 reste optimale.

2 PLNE

On suppose à présent que tout mobilier commencé doit être terminé la même semaine.

6. Quelle modification apporter au modèle mathématique précédent ?

7. Résoudre le problème résultant par une procédure de séparation et évaluation de votre choix dont vous aurez spécifié les paramètres
Suggestion : se servir des relaxations linéaires obtenu par simplexe de manière graphique pour calculer les bornes

3 Programmation dynamique

La figure ci-après montre l'image d'une chaussée comportant des fissures. Une telle image en niveaux de gris peut être considérée comme une matrice I, dont les éléments I(i,j) sont des entiers compris entre 0 et 255 : i désigne le numéro de ligne, j le numéro de colonne.

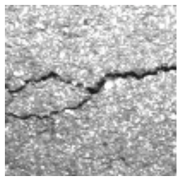

figure 1

0	1	2	1	0	2	2	0	1
2	0	1	2	1	2	1	0	1
0	2	0	0	1	1	0	1	2
2	0	1	1	2	0	2	4	**3**
0	1	2	0	2	2	1	2	1

tableau 1

Un chemin menant du point $D = (i_D, j_D)$ au point $A = (i_A, j_A)$ est défini comme une séquence de pixels comportant un seul pixel (i_j, j) par colonne j, de telle sorte que deux pixels consécutifs soient voisins, au sens des huit plus proches voisins. On suppose que les contraintes $j_D \leq j_A et |i_D - i_A| \leq j_A - j_D$ sont satisfaites. Le programme doit déterminer le chemin entre D et A qui minimise l'objectif suivant :

$$z = \sum_{j=j_D}^{j_A} I(i_j, j)$$

On s'intéresse à l'exemple du tableau 1 où les points D et A sont indiqués en gras :

1. Appliquez l'algorithme de programmation dynamique permettant de calculer, en chaque pixel atteignable, le score du chemin optimal menant de D à ce pixel, ainsi que ce chemin. En cas d'ambiguïté sur le pixel précédent, c'est le voisin de gauche qui doit être privilégié (c'est-à-dire la direction horizontale).

2. Déduisez de la question précédente le chemin optimal menant de D à A, en entourant sur la figure précédente les pixels parcourus

3. Dans l'algorithme de programmation dynamique, le problème d'optimisation locale s'écrit : $g(i, j) = \min\{g(i-1, j-1) + p1 * I(i, j), g(i, j-1) + p2 * I(i, j), g(i+1, j-1) + p3 * I(i, j)\}$.
   En modifiant les poids de telle sorte que (p1, p2, p3) = (1, 2, 1), effectuez la résolution de ce nouveau problème sur le même exemple que précédemment. Qu'observez-vous ?