



**Forum: WD 10**

**Topic: Algorithme de recherche de combinaisons possibles...**

**Subject: Re: Algorithme de recherche de combinaisons possibles...**

Posté par: Bebe

Contribution le : 6/9/2006 13:46:48

Ton problème ressemble à celui du voyageur de commerce (un grand classique).  
Les algo qui permettent de résoudre ce type de problème sont tous, plus ou moins, heuristiques.  
Le temp nécessaire à l'exécution augmente de façon exponentielle en fonction du nombre de noeuds.  
Il est donc important de ne pas vouloir calculer tous les cas,  
et de concentrer son effort sur ceux qui sont potentiellement les plus favorables.

**Un début de solution (trouvé dans un livre):**

Le problème peut être "visualiser" sous la forme d'un graphe.  
Soit N villes à visiter (les Noeuds).  
On connaît la distance entre chaque ville (les arcs)  
exemple  $d(I,J)$  pour la distance entre les villes I et J  
Il faut visiter chaque ville 1 seule fois et en parcourant le minimum de distance.

L'algorithme proposé est approximativement celui établi par LITTLE en 1955

L'idée fondamentale est de supprimer les arcs coûteux  
et de ne supprimer un arc que si on est assuré de faire moins cher.  
Le noeud I et le noeud J étant donné, pour supprimer l'arc IJ  
il faut s'assurer au préalable qu'on peut arriver à J en partant de I  
et en passant par d'autres arcs intermédiaires.

Pour cela on utilise une matrice  $N \times N$ ,  $(A(i,j))$  pour stocker les valeurs des arcs.  
On élague ensuite la matrice des arcs trop coûteux.  
Une fois la matrice élaguée on effectue des choix parmi les arcs restants en gérant au fur et à mesure  
l'ensemble des arcs, leurs liens respectifs tenant compte des choix faits.  
Parfois, si la suite des choix faits ne permet pas de conclure, il faut être capable  
d'effectuer des "retour-arrière" (back-track) et refaire de nouveaux choix plus favorables.

**Algorithme de parcours en pseudo-code**

on appelle rangée une ligne ou une colonne et on pourra choisir de se déplacer à ligne ou à colonne constante.

POUR  $i=1$  à N FAIRE

E1: SI il existe une rangée interdite ALORS ALLER A BT;

SINON choisir la rangée avec le moins de choix possibles  
faire un choix  
générer les interdictions relatives à ce choix  
SI  $i=N-1$  ALLER A E2  
SINON ALLER A FIN POUR;

BT: SI  $i=1$  ALORS FIN;  
SINON  $i=i-1$ ;

E2: éliminer les interdictions relatives à ce choix  
ALLER A E1

### **Heuristique d'élagage de la matrice**

On calcule en quelque sorte une contrainte DL  
et on effectue le parcours de la matrice restante suivant l'algorithme ci-dessus.  
Tout d'abord on calcule MI, somme des minimums de la matrice selon les lignes et MA pour les maximums.

On en déduit  $DL = (MA-MI)/(2*N*II)$

avec II initialisé à environ 6 (je ne sais pas pourquoi mais c'est fonction de la taille de la matrice )

et qu'on décrémentera si nécessaire.

si on augmente II on est plus sélectif - si on le décrémente on "relache" les contraintes

En espérant avoir aidé ...