07/12/2016

Maxime LAVASTE

[nom de la société]

Projet de MOGPL

Résolution d’un casse-tête Rush Hour

1. Résolution par programmation linéaire en variables binaires
2. min RHM = k

min RHC = ??

1. A. La voiture rouge est positionné devant la sortie au terme du dernier mouvement.

B. Au plus un véhicule est déplacé par tour.

C. La position du marqueur d’un véhicule i est bien mise à jour si le véhicule i se déplace.

 !!! Vraiment pas sur !!

II

8. (RHC)

Le poids d’une arrête entre 2 sommets représentent le nombre de cases d’un mouvement entre 2 configurations. En appliquant l’algorithme de Dijkstra, on obtient le plus court chemin et donc un chemin avec le moins de cases de déplacements.

A chaque itération de l’algorithme, on marque le sommet (non marqué) qu’on peut atteindre avec la plus petite distance.

Ainsi, la condition d’arrêt pour Dijkstra est le marquage d’un sommet correspondant à une configuration-but, c’est-à-dire quand la voiture rouge est devant la sortie.

9.

En stockant les prédecesseurs de chaque sommet lors de l’exécution de l’algorithme de Dijkstra, on est capable de recréer la séquence de déplacements. En effet, un sommet équivaut à une configuration. Il suffit alors de partir de la solution et de remonter les prédécesseurs jusqu’à retrouver la configuration de départ. On a au final la séquence de déplacements de la solution vers la configuration initiale, il suffit alors de l’inverser pour avoir cette séquence dans le bon ordre.

NB : Il aurait été plus difficile d’utiliser un tableau de successeurs vu qu’un sommet du sous-graphe à la fin de l’algorithme peut avoir plusieurs successeurs alors qu’il n’a qu’un seul prédécesseur.

10. (RHM)

S’il existe une arrête entre deux sommets du graphe des configurations, il existe alors un déplacement (d’un véhicule) permettant de passer de l’un à l’autre. Dans un problème RHM, on cherche à minimiser le nombre de ces déplacements. Lorsque l’on déplace un véhicule de 1,2,3,etc. cases, on n’effectue au final qu’un seul déplacement.

Le poids de chaque arrête est donc fixée à 1 et on applique Dijkstra pour un chemin de poids minimal (et donc effectuant le moins de mouvement) vers un sommet représentant une configuration-but.

11.

nombre de config réalisable a.k.a

soit le nb total de config (config.size()) ?

soit compter le nb de config pour N mv donné

2e possibilité :

On envoie un paramètre supplémentaire à la fonction de Dijkstra pour le nombre de mouvements. On utilise deux compteurs internes : un pour compter le nombre de configurations réalisables et un autre pour savoir si on a atteint le nombre de mouvements passé en paramètre.

Initialisation du compteur mv à 1 ?

Lors d’un marquage de sommet, on incrèmente de 1 le compteur interne de mouvement.

Pour chaque sommet marqué, on mettait à jour la distance pour atteindre ses successeurs, il suffit d’incrémenter de 1 le compteur autant de fois qu’il y a de successeurs. Cette incrémentation ne s’effectue pas si on atteint le nombre de mouvement demandé.