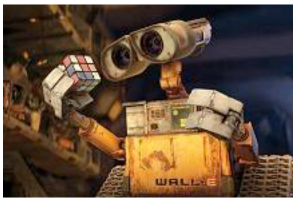

Algorithmique 2 – Esiee Paris

TD3 et TP3 – semaines 23 et 24

Les rapports sont à remettre lors de la séance de TP

R. Natowicz, A. Çela, I. Alamé

Un petit robot se déplace sur une grille carrée à n lignes et n colonnes. Son point de départ est la case (0,0), son point d'arrivée est la case $(n-1, n-1)$. Ses déplacements possibles sont :

- déplacement horizontal, $(i, j) \longrightarrow (i, j+1)$, dont le coût est $E(i, j, n)$; (E = Est)
- déplacement vertical, $(i, j) \longrightarrow (i+1, j)$, dont le coût est $N(i, j, n)$; (N = Nord)
- déplacement diagonal, $(i, j) \longrightarrow (i+1, j+1)$, dont le coût est $NE(i, j, n)$ (NE = Nord-Est.)

Tout déplacement qui ferait sortir le robot de la grille a un coût infini. Exemple : $NE(n-1, 0, n) = \infty$.

Minimisation locale.

Étant sur la case (i, j), le robot choisit de se déplacer dans une des directions dont le coût est localement minimum.

1. Écrire le programme correspondant. Ce programme affichera les cases du chemin suivi, ainsi que le coût du chemin.

   Invariant $I(i, j, c')$: un chemin de coût c obtenu par minimisation locale depuis la case $(0, 0)$ jusqu'à la case $(n-1, n-1)$ est un chemin de coût c' obtenu par minimisation locale depuis la case $(0, 0)$ jusqu'à la case (i, j) suivi d'un chemin de coût $c - c'$ obtenu par minimisation locale depuis la case (i, j) jusqu'à la case $(n-1, n-1)$. Le chemin de $(0, 0)$ à (i, j) a été affiché et le chemin de (i, j) à $(n-1, n-1)$ reste à afficher.

   $$(0,0) \xrightarrow{c'} (i,j) \xrightarrow{c-c'} (n\text{-}1, n\text{-}1)$$

   Choisir des fonctions de coût Est, Nord et Nord-Est qui montrent que le chemin obtenu par minimisation locale n'est pas de coût minimum dans le cas général (en fait, sauf cas particulier, il n'est pas de coût minimum.) Exemple de fonctions : $E(i, j, n) = i$, $N(i, j, n) = j^2$, $NE(i, j, n) = i^j$ pour tout déplacement qui ne fait pas sortir le robot de la grille (sinon : $+\infty$.)

Minimisation globale.

On note $m(\ (i, j)\ )$ le coût minimum d'un chemin allant de la case $(0, 0)$ à la case (i, j). On calculera tous les coûts minimum dans une matrice $M[0 : n][0 : n]$ de terme général $M[i][j] = m(\ (i, j)\ )$. En particulier $M[n-1][n-1] = m(\ (n-1, n-1)\ )$ sera le coût minimum d'un chemin allant de la case origine $(0, 0)$ à la case destination $(n-1, n-1)$. On calculera également, à la volée, une matrice $A[0 : n][0 : n]$ de terme général $A[i][j] = \arg\min m(\ (i, j)\ )$, où $A[i][j]$ est la direction d'arrivée sur la case (i, j) : 0, 1 ou 2, respectivement Est, Nord, ou Nord-Est. On posera $A[0][0] = \arg\min m(\ (0, 0)\ ) = 2$.

Nous définissons la taille du problème $m(\ (i, j)\ )$ comme étant la valeur i (la ligne i de la grille.)

2. Base : on pose $m(\ (0, 0)\ ) = 0$. Donner les valeurs $m(\ (0, j)\ )$, $1 \leq j < n$, et $m(\ (i, 0)\ )$, $1 \leq i < n$.
3. Cas général : pour toute case (i, j), $1 \leq i < n$ et $1 \leq j < n$, donner l'expression du coût minimum $m(\ (i, j)\ )$ en fonction des coûts minimum $m(\ (i, j-1)\ )$, $m(\ (i-1, j)\ )$, $m(\ (i-1, j-1)\ )$ et des coûts de déplacement Est, Nord, Nord-Est, $E(i, j-1, n)$, $N(i-1, j, n)$, $NE(i-1, j-1, n)$;
4. écrire le programme calculant la matrice $M[0 : n][0 : n]$ de terme général $M[i][j] = m(\ (i, j)\ )$ et la matrice $A[0 : n][0 : n]$ de terme général $A[i][j] = \arg\min m(\ (i, j)\ )$. Quelle est sa complexité ?
5. écrire un programme d'affichage d'un chemin de coût minimum allant de la case origine $(0, 0)$ à la case destination $(n-1, n-1)$. Le chemin sera affiché à l'envers. Pour cet affichage, le chemin de coût minimum sera décomposé en deux parties : le chemin allant de la case $(0, 0)$ à la case (i, j), chemin restant à afficher, et le chemin allant de la case (i, j) à la case $(n-1, n-1)$, déjà affiché.
6. comparer les coûts des chemins obtenus par minimisation locale et par minimisation globale avec le jeu fonctions E, N et NE proposé ci-dessus; et avec un autre jeu de fonctions que vous choisirez et décrirez.