

Documentation

Groupe 3 Bleu : Sokoban

Goron Nathan, De La Rosa Louis-David, Basset Emilien, Demé Quentin

Algorithme A*, fonctionnement

1 Principes de fonctionnement.

1.1 Présentation et objectif :

L'algorithme A* est un algorithme souvent utilisé pour trouver des chemins dans la résolution de jeux. Il est dans notre cas implémenté sur une grille de Sokoban. En dehors du sokoban où cela nous importe peu, il peut être interrompu pendant son utilisation est relancé plus tard. C'est un algorithme de type "anytime". Le but de A* est de trouver le chemin *le plus court* pour aller d'un point de départ, à un point d'arrivée. Pour cela l'algorithme garde une liste de toutes les étapes futures que l'on appelle "liste ouverte". Cette liste est établie par la fonction $g(n)$. Ensuite, il choisit parmi cette liste une étape future qui soit "à priori" la meilleure pour nous mener à l'objectif en un temps qui soit le plus court possible. Pour que cela puisse se faire, nous avons besoin d'une heuristique qui puisse nous aider à déterminer quelle est "à priori" la meilleure option. Une fois que cette option est choisie, l'algorithme passe à la "liste fermée".

2 Explication détaillée avec exemples.

2.1 Fonctions g et h :

A chaque étape de l'algorithme, les fonctions $g(n)$ et $h(n)$ sont exécutées pour n étant un point d'arrivée. Cela nous donne les informations suivantes.

- $g(n)$ = Le nombre d'étapes pour aller du point de départ à l'arrivée n .
- $h(n)$ = L'heuristique qui estime le coup pour aller de n à l'objectif final.
- $f(n) = g(n) + h(n)$: Le nombre minimum d'étapes si on choisit le point n .

2.2 Etapes détaillées de l'algorithme :

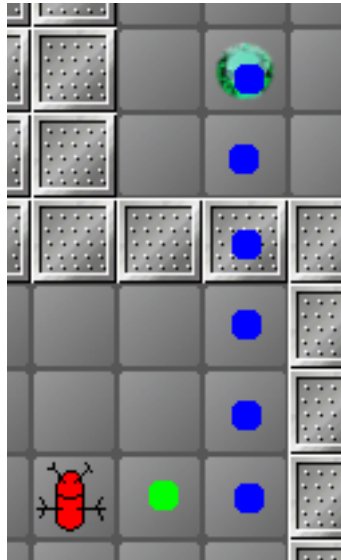
Voici les étapes de l'algorithme en langage algorithmique :
Soient : OPEN la liste ouverte
(La partie ci-dessous n'a pas pu être indentée correctement, attention donc!)
Ajouter DEPART à OPEN
Tant que OPEN n'est pas vide :
 Choisir le N ayant le plus petit $f(n)$
 Si N est l'objectif alors PASS
 Déplacer N dans CLOSED
 Pour chaque $N' = \text{PeuBouger}(N, \text{direction})$:
 $g(N') = g(N) + 1$
 $h(N')$
 Si N' dans OPEN et nouveau N' n'est pas mieux, CONTINUE
 Si N' dans CLOSED and nouveau N' n'est pas mieux, CONTINUE
 supprimer tout N' étant dans OPEN et CLOSED (en même temps)
 Ajouter N comme parent de N'
 Ajouter N' à OPEN

FIN POUR
FIN TANT QUE
Si on arrive ici, alors il n'y a pas de solutions.

3 Le cas du Sokoban :

3.1 Cas général :

Dans le cas du sokoban, l'heuristique sera le minimum de déplacements pour atteindre l'objectif si un chemin direct est possible. Or puisque l'on se déplace uniquement à la verticale ou à l'horizontale, cette donnée sera la somme des déplacements horizontaux et verticaux à réaliser.



Ce schéma présente le minimum de déplacements pour aller du point rouge au point d'arrivée.

Ainsi pour l'étape en vert :

$$g(n) = 1$$

$$h(n) = 6$$

$$f(n) = 1 + 6 = 7$$