TP Techniques d'Intelligence Artificielle

EL HADJ MUSTAPHA Mohamed Kais

LUSSIEZ Corentin

1- Grid

The project is based initially on a 3x3 grid, since the number of states the grid could take isn’t that wide, we managed to solve the problem using A\*, BFS and IDS.

DFS took so much time (more than 40 mins) and yet it couldn’t find the solution, so it stopped the execution, but it eventually found the solution, if there is one.

Naturally, we can change the size of the grid and everything would still work, nonetheless the state space gets a lot wider just by going from 3x3 to 4x4, to the point where we managed to solve a 4x4 grid only using BFS (while taking more than 30 minutes), while the others took so much time we stopped execution.

1- Agents

Assuming that we have a grid of size 3, we designed the structure of the grid to have 3\*3-1 agents, each one of them disposing of several properties such as:

Id

Current position

Goal position

Previous position

Status (active, arrived)

Remaining Distance to the goal position

Steps taken so far

Each agent is capable of moving on the grid according to the validity of the move to take and the strategies used to determine that move.

We added one more agent, which is a master agent. This one will be responsible for finding out which one of the previous agents should move and requesting them to do so, in order to resolve the problem and send each agent to its right position.

The agent communication established goes through the master agent and the regular agents.

The master agent determines the next agent to move, then it requests it to do so.

We could’ve designed the system in a way where each regular agent determines the next agent to move and waits for its turn through communications with the other agents, but this is a simpler way to do it.

2- strategies

The master agent determines the next move through one of the following strategies: BFS, DFS, IDS, A\*.

For the A\* algorithm,It uses the euclidean distance as a heuristic function, more specifically, it uses the sum of the remaining distances for each normal agent, which is noted as the h() function. Secondly we use a g() function, which is the actual used heuristic function to find the shortest path.

When multiple agents have moves with equal f() function values, we simply choose the agent that appeared first.

The other algorithms don’t require any kind of heuristic functions since they are blind search algorithms.

On a 3x3 grid, the best performing algorithm was A\* followed by IDS then BFS and DFS lastly.

FR

TP Techniques d'Intelligence Artificielle

EL HADJ MUSTAPHA Mohamed Kais

LUSSIEZ Corentin

1- Grille

Le projet est basé initialement sur une grille 3x3, puisque le nombre d'états que la grille peut prendre n'est pas très grand, nous avons réussi à résoudre le problème en utilisant A\*, BFS et IDS.

DFS a pris tellement de temps (plus de 40 mins) et pourtant il n'a pas pu trouver la solution, donc il a arrêté l'exécution, mais il a finalement trouvé la solution, s'il y en a une.

Naturellement, nous pouvons changer la taille de la grille et tout fonctionnerait encore, néanmoins l'espace d'état devient beaucoup plus large juste en passant de 3x3 à 4x4, au point que nous avons réussi à résoudre une grille 4x4 seulement en utilisant BFS (tout en prenant plus de 30 minutes), alors que les autres ont pris tellement de temps que nous avons arrêté l'exécution.

1- Agents

En supposant que nous avons une grille de taille 3, nous avons conçu la structure de la grille pour avoir 3\*3-1 agents, chacun d'entre eux disposant de plusieurs propriétés telles que :

Id

Position actuelle

Position cible

Position précédente

Statut (actif, arrivé)

Distance restante jusqu'à la position de but

Pas effectués jusqu'à présent

Chaque agent est capable de se déplacer sur la grille en fonction de la validité du déplacement à effectuer et des stratégies utilisées pour déterminer ce déplacement.

Nous avons ajouté un agent supplémentaire, qui est un agent maître. Celui-ci sera chargé de trouver lequel des agents précédents doit se déplacer et de leur demander de le faire, afin de résoudre le problème et d'envoyer chaque agent à sa bonne position.

La communication établie entre les agents passe par l'agent maître et les agents réguliers.

L'agent maître détermine le prochain agent à déplacer, puis il lui demande de le faire.

Nous aurions pu concevoir le système de manière à ce que chaque agent régulier détermine le prochain agent à déplacer et attende son tour en communiquant avec les autres agents, mais ceci est une manière plus simple de procéder.

2- stratégies

L'agent maître détermine le prochain mouvement par l'une des stratégies suivantes : BFS, DFS, IDS, A\*.

Pour l'algorithme A\*, il utilise la distance euclidienne comme fonction heuristique, plus précisément, il utilise la somme des distances restantes pour chaque agent normal, ce qui est noté comme la fonction h(). Ensuite, nous utilisons une fonction g(), qui est la fonction heuristique réellement utilisée pour trouver le chemin le plus court.

Lorsque plusieurs agents ont des déplacements avec des valeurs de fonction f() égales, nous choisissons simplement l'agent qui est apparu en premier.

Les autres algorithmes ne nécessitent aucune fonction heuristique puisqu'il s'agit d'algorithmes de recherche aveugle.

Sur une grille 3x3, l'algorithme le plus performant est A\*, suivi de IDS, puis de BFS et enfin de DFS.