

Cours d'Intelligence Artificielle : IA__Solve

(À remettre le 24 octobre 2024)

Ce devoir à la maison est noté sur 20 points.

1. Recherche d'un chemin optimal par l'algorithme A* (3 points)

Soit le graphe correspondant à la figure 1 dans lequel les nombres sur les arcs indiquent le coût associé à cet arc, et les nombres donnés pour chaque nœud dans le tableau en haut à droite indiquent la valeur retournée par l'heuristique pour ce nœud et le but « G ».

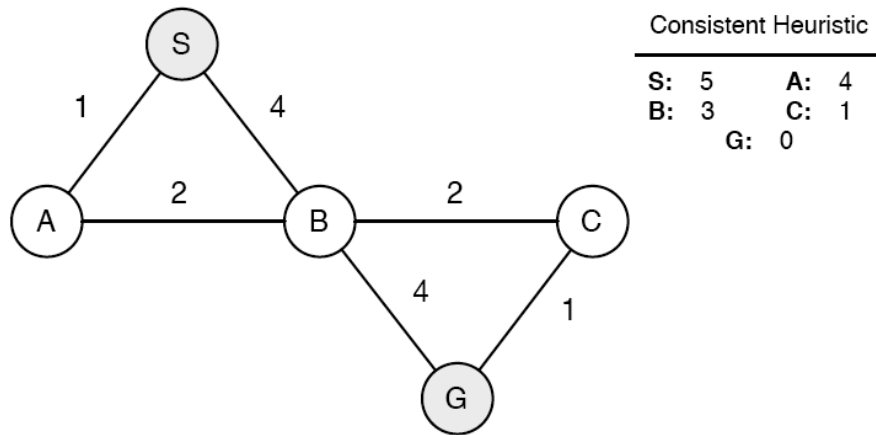


FIGURE 1 – Graphe.

Donnez l'ordre d'exploration des nœuds développés par une recherche en meilleur d'abord commençant par le nœud 'S'. Montrez l'évolution des listes OUVERT et FERMÉ. Dites quel est le chemin retourné.

2. Optimalité de l'algorithme A* (4 points)

Prouvez que l'algorithme A* retourne un chemin de coût optimal si la fonction heuristique $h(n)$ est admissible.

3. Transformation de A* pour trouver un chemin le plus long (6 points)

L'algorithme A* permet de trouver un plus court chemin dans un graphe. On va chercher ici comment le transformer pour qu'il permette de trouver un plus long chemin.

1. Que se passe-t-il si on n'utilise pas de fonction heuristique pour trouver un chemin le plus long?

2. Quelle doit être la propriété d'une fonction heuristique pour que le chemin retourné soit un plus long chemin ?
Justifier théoriquement que cela garantit que le chemin retourné sera un chemin le plus long.
3. Donner l'algorithme retournant un plus long chemin.
4. Le tester sur le petit graphe de la figure 1 et montrer l'ordre d'exploration des nœuds.

4. Jeu à trois joueurs (7 points)

Comment pourrait-on modifier l'algorithme minimax pour qu'il soit utilisable dans le cas d'un jeu à information parfaite à trois joueurs ?

Par exemple, considérons un jeu à trois joueurs nommés 0, 1 et 2 (voir figure 2). On va supposer que la fonction d'évaluation retourne un triplet de valeurs indiquant l'évaluation de la position pour les joueurs 0, 1 et 2. Par exemple, le triplet (6, 2, 4) indique que cette position est très désirable pour le joueur 0, nettement moins pour le joueur 1 et moyennement pour le joueur 2.

1. Compléter l'arbre de jeu de la figure 2 en remplissant les triplets jusqu'à la racine.
 - (a) Dans le cas où chaque joueur considère les deux autres joueurs comme coalisés contre lui.
 - (b) Dans le cas où chaque joueur tente de maximiser son propre gain (pas de coalition).
2. Ré-écrire le programme MinMax pour qu'il joue correctement sur ce type d'arbre pour l'un des deux cas précédents que vous choisirez.

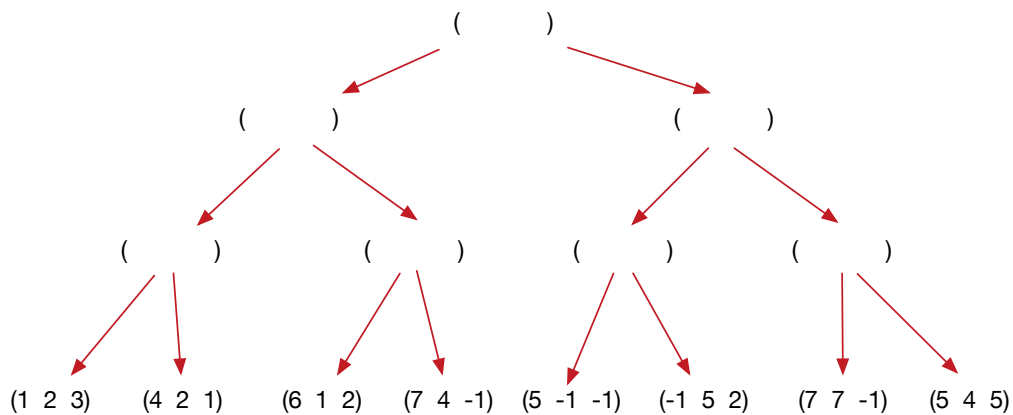


FIGURE 2 – Les trois premiers coups d'un jeu à trois joueurs.