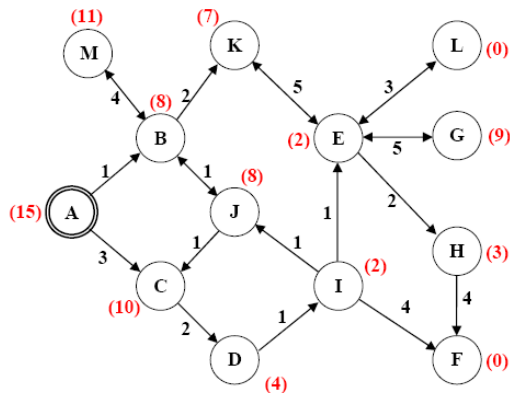


INTELLIGENCE ARTIFICIELLE

Problème de recherche

Le graphe suivant est le graphe d'état d'un problème :



Les chiffres indiqués entre parenthèses sont les valeurs de la fonction heuristique, et ceux qui ne sont pas entre parenthèses sont les coûts des actions permettant de passer d'un état à un autre. L'état initial est le noeud A.

Question 1. analyse du graphe

En observant le graphe, déduisez quel est l'état terminal, ou quels sont les états terminaux (i.e. les objectifs).

L'heuristique utilisée est-elle admissible ? Pourquoi ?

Question 2. recherche aveugle

Sans appliquer l'algorithme, quel est d'après vous le type de recherche aveugle le plus adapté à ce problème : recherche en profondeur ou recherche en largeur ? Expliquez rapidement pourquoi.

Question 3. recherche informée

Appliquez l'algorithme A*. Détaillez le déroulement de l'algorithme pas à pas (avec du texte, comme en cours et en TP, inutile de donner l'arbre de recherche). A la fin, vous indiquerez quel est le chemin que l'algorithme a parcouru pour trouver la solution, et vous indiquerez aussi le chemin qui constitue la solution.

La solution est-elle optimale ? Pourquoi ?

Remarque importante : quand deux noeuds de la fringe ont la même valeur, on les range dans l'ordre alphabétique.

SOLUTION

Question 1. analyse du graphe

Les états terminaux sont ceux qui ont une heuristique nulle, donc ici : F et L .

L'heuristique n'est pas admissible, car il existe des noeuds pour lesquels elle est supérieure à la distance réelle à la solution. Par exemple : pour G , l'heuristique est 9. Or, la distance à F est 11 et la distance à L est 8, donc pour ce noeud G l'heuristique surestime la distance minimale à la solution.

Question 2. recherche aveugle

L'algorithme de recherche en profondeur va devoir explorer un arbre de recherche de profondeur infinie. En effet, si on introduit les fils dans l'ordre alphabétique, l'algorithme va se bloquer dans le cycle C, D, I, J, C (**remarque** : on ne demandait pas d'appliquer l'algorithme dans le sujet) :

- [A]
- [B, C] ←
- [B, D]
- [B, I]
- [B, E, F, J]
- [B, E, F, B, C] ←
- etc.

Et même si on introduit les fils dans l'ordre anti-alphabétique, on a un blocage sur le cycle B, J, B :

- [A]
- [C, B] ←
- [C, M, K, J]
- [C, M, K, C, B] ←

La recherche en largeur sera donc plus efficace dans ce cas-là.

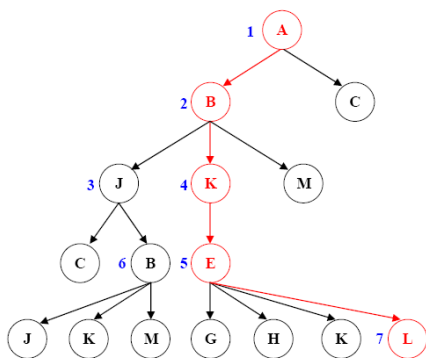
Question 3. recherche informée

Déroulement de l'algorithme A^* :

- **frange** : [A(15)]
- **dév.** : $A \rightarrow B(1+8), C(3+10)$

- **frange** : [B(9), C(13)]
- **dév.** : B \rightarrow J(2+8), K(3+7), M(5+11)
- **frange** : [J(10), K(10), C(13), **M(16)**]
- **dév.** : J \rightarrow C(3+10), B(3+8)
- **frange** : [K(10), **B(11)**, C(13), **C(13)**, M(16)]
- **dév.** : K \rightarrow E(8+2)
- **frange** : [E(10), B(11), C(13), C(13), M(16)]
- **dév.** : E \rightarrow G(13+9), H(10+3), K(13+7), L(11+0)
- **frange** : [B(11), **L(11)**, C(13), C(13), **H(13)**, M(16) , **K(20)**, **G(22)**]
- **dév.** : B \rightarrow J(4+8), K(5+7), M(7+11)
- **frange** : [L(11), **J(12)**, **K(12)**, C(13), C(13), H(13), M(16) , **M(18)**, K(20), G(22)]
- **dév.** : L \rightarrow **solution**

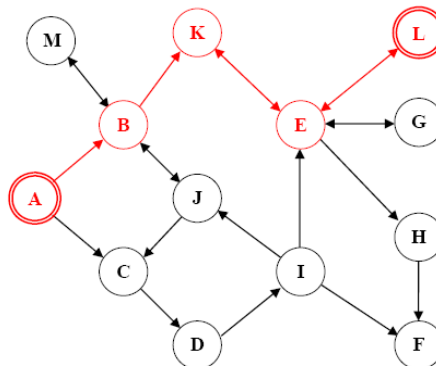
Arbre de recherche :



Chemin parcouru : A, B, J, K, E, B, L

Chemin solution : A, B, K, E, L

graphe :



La solution n'est pas optimale, car il existe des chemins moins coûteux :

- A, B, K, E, L coûte 11
- Mais A, B, J, C, D, I, F coûte 10
- Et A, C, D, I, F coûte aussi 10
- Etc.

Ce qui est normal, puisque l'heuristique n'est pas admissible : l'algorithme A* n'est alors pas optimal.