AiProjet

IHW

January 2023

1 Cannibales et missionnaires

Question 1 :

Pour résoudre ce problème, il est nécessaire de prendre en compte les états utiles qui vont servir trouver la solution la plus optimale. On peut décrire ces états grâce aux variables suivantes :

- m : Le nombre de missionnaires
- c : Le nombre de cannibales

Question 2 :

Au départ, tous les missionnaires et cannibales sont sur une même rive du fleuve et la barque est vide.

Sur la rive de départ (état initial) :

- Il y a 3 missionnaires sur la rive gauche
- Il y a 3 cannibales sur la rive gauche
- La barque est vide, c'est-à-dire qu'il n'y a personne dedans

Sur la rive d'arrivée (état final) :

- Il y a 3 missionnaires sur la rive droite
- Il y a 3 cannibales sur la rive droite
- La barque est vide, c'est-à-dire qu'il n'y a personne dedans

Question 3 :

Pour déterminer le nombre majorant d'État, nous devons prendre en compte tous les états possibles, qui sont déterminés par le nombre de missionnaires et de cannibales sur chaque rive du fleuve, ainsi que par le nombre de personnes dans la barque.

En utilisant 3 missionnaires et 3 cannibales, il y a 4 possibilités pour le nombre de missionnaires et 4 possibilités pour le nombre de cannibales sur chaque rive, ainsi que 3 possibilités pour le nombre de personnes dans la barque. Cela donne un total de 48 états possibles. Schématiquement cela donne :

- Missionnaire : {0,1,2,3}
- Cannibales : {0,1,2,3}
- Barque : {0,1,2}
- Nombre d'état possible : 4*4*3 = 48

Pour stocker 48 états, il faut stocker 3 variables pour chaque état (nombre de missionnaires sur chaque rive, nombre de cannibales sur chaque rive et nombre de personnes dans la barque).
Donc en total il faut stocker 48 * 3 = 144 variables. Et si on suppose que chaque variable prend 2 octets pour stocker la valeur, cela signifie que le stockage nécessaire pour stocker tous les états est de 144 * 2 = 288 octets, soit 288/1024 = 0.28125 Ko, c'est suffisamment raisonnable pour être stocké dans la mémoire d'un ordinateur moderne.

Question 4 :

Dans ce problème, une action est une transition d'un état à un autre en effectuant une opération spécifique, comme traverser le fleuve en barque. Les actions possibles dans ce problème en respectant les contraintes imposés par le problème sont les suivantes :

- Emmener 1 missionnaire et 1 cannibale en barque en même temps
- Emmener 2 missionnaires en barque en même temps
- Emmener 2 cannibales en barque en même temps
- Emmener 1 cannibale en barque seul
- Emmener 1 missionnaire en barque seul

Question 5 :

Un exemple d'état pourrait être {md3, cd3, b0, mg0, cg0} où 3 missionnaires et 3 cannibales se trouvent sur la rive gauche du fleuve, la rive droite est vide, et personne ne se trouve dans la barque. A partir de cet état initial, deux actions différentes sont possibles :

- Embarquer 1 missionnaire et 1 cannibale sur la barque.
  État atteint : {md2, cd2, b0, mg1, cg1} (2 missionnaires à gauche, 2 cannibales à gauche, barque vide, 1 missionnaire à droite, 1 cannibale à droite).
- Embarquer 2 cannibales sur la barque.
  État atteint : {md3, cd1, b0, mg0, cg2} (3 missionnaires à gauche, 1 cannibale à gauche, barquevide, 0 missionnaire à droite, 2 cannibales à droite).

Il ne faut pas oublier qu'il existe d'autre exemple d'état ou 2 actions différentes sont possibles.

Question 6 :

Un exemple d'état à partir duquel 1 seule action est possible est {md1, cd1, b0, mg2, cg2}. Dans cet état, il est impossible de déplacer plus d'une personne à la fois, car cela violerait la contrainte où le nombre de missionnaires sur une rive ne doit pas être inférieur au nombre de cannibales. L'état qu'on atteint alors est celui où l'on a déplacé une seule personne. L'état qu'on atteint alors

est {md0, cd1, b0, mg3, cg2} ou on a déplacé un missionnaire.

Question 7 :

Il existe des états à partir desquels aucune action n'est plus possible. Par exemple, si les trois cannibales et aucun missionnaire se trouvent sur la rive droite du fleuve {md3, cd0, b0, mg0, cg3}, il n'est plus possible de continuer à passer de l'autre côté car cela violerait la condition selon laquelle le nombre de missionnaires d'un côté ou de l'autre du fleuve ne peut jamais être strictement inférieur au nombre de cannibales.

Question 8 :

Il n'y a pas de fonction de coût spécifique à choisir pour les actions dans ce problème. Le but est de passer toutes les personnes de l'autre côté du fleuve en respectant les contraintes de la barque (1 ou 2 personnes) et en s'assurant que le nombre de missionnaires d'un côté ou de l'autre du fleuve ne peut jamais être strictement inférieur au nombre de cannibales. Il n'est pas nécessaire de mesurer un coût pour les actions, il suffit de vérifier si les contraintes sont respectées et si l'objectif final est atteint.

Question 9 :

Il existe plusieurs algorithmes de recherche qui peuvent être utilisés pour résoudre ce problème, tels que:

- L'algorithme de recherche en profondeur d'abord (DFS) qui explore les états en allant en profondeur dans l'arbre de recherche pour trouver une solution ou un état mort.
- L'algorithme de recherche en largeur d'abord (BFS) qui explore les états en allant en largeur dans l'arbre de recherche en explorant toutes les actions possibles à partir d'un état donné avant de passer à l'état suivant.
- L'algorithme de recherche A* : cet algorithme utilise une fonction d'évaluation (heuristique) pour évaluer la qualité des états et orienter la recherche vers les états les plus prometteurs. Il est généralement considéré comme étant l'un des algorithmes les plus efficaces pour résoudre ce type de problème.
- L'algorithme IDA* (Iterative Deepening A*) : c'est une variante de A* qui limite la profondeur de recherche pour éviter d'explorer des états inutiles. Il peut être plus efficace que A* pour les problèmes qui ont des états avec une profondeur très élevée.

En pratique, l'algorithme A* ou IDA* ont des performances très bonnes pour résoudre ce genre de problème, car ils permettent de prioriser les états les plus prometteurs et de limiter les états explorés inutilement. Cependant l'algorithme IDA* est celui qui à la meilleure performance car il permet de limiter la profondeur de recherche tout en utilisant une fonction de coût pour évaluer les états. Il est donc plus efficace que la recherche en profondeur d'abord, qui peut s'égarer dans des branches qui ne mènent pas à une solution, tout en

étant plus rapide que la recherche en largeur d'abord qui doit explorer tous les états.