Artificial Intelligence

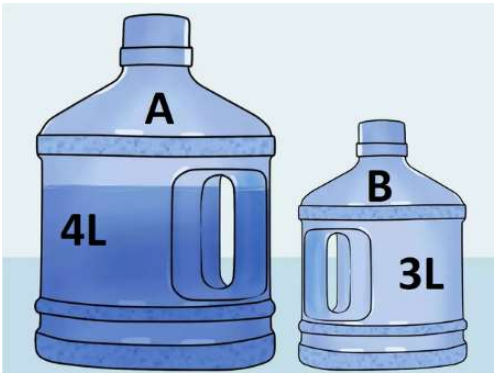
Exercise Sheet 1

Formulation of Search Problems and Problem-Solving using Search

1.1 Two Buckets Problem

Two buckets of capacities c1 (e.g. 4 liters) and c2 (e.g. 3 liters), respectively, are initially empty. Buckets do not have any intermediate markings. The only operations you can perform are:

- Fill (completely) a bucket
- Empty a bucket.
- Pour one bucket into the other (until the second one is full or until the first one is empty).



The aim is to determine which operations to carry out so that the first bucket contains *n liters* (e.g. 2 litres).

a) Formulate this problem as a search problem by defining the state representation, initial state, operators (their name, preconditions, effects, and cost), and objective test.
b) What is the space state size for this problem? Represent it showing the possible objective states and display some of the possible transitions in the space state from the initial state.
c) Solve the problem, by hand, using tree search.
d) Using a programming language of your choice, solve the problem by applying:
   (c1) Breadth-first search strategy.
   (c2) Depth-first search strategy (limited depth).
   (c3) Iterative deepening strategy.

1.2 Missionaries and Cannibals Problem

Three missionaries and three cannibals are on one of the banks of the river with a boat that only takes one or two people. The boat cannot travel the river alone. The goal is to find a way to get the six to the other bank of the river without ever leaving more cannibals than missionaries on one of the banks (even at the instant they leave/join the boat) during the process.

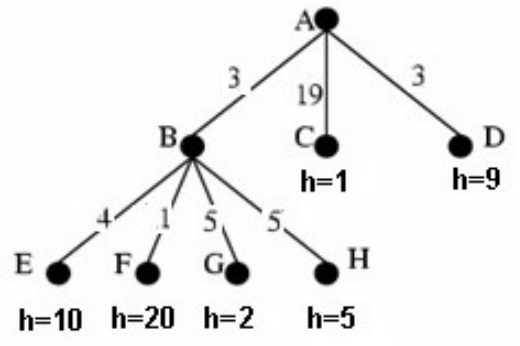
a) Formulate this problem as a search problem by defining the state representation, initial state, operators (their name, preconditions, effects, and cost), and objective test.

b) What is the space state size for this problem? Represent it showing the possible objective states and display some of the possible transitions in the space state from the initial state.
c) Solve the problem, by hand, using tree search.
d) Using a programming language of your choice, solve the problem by applying:
(d1) Breadth-first search strategy; (d2) Depth-first search strategy (limited depth);
(d3) Iterative deepening strategy.

1.3 Strategies for Uninformed/Informed Search

Assuming the following search tree in which each arc displays the cost of the corresponding operator, and the nodes contain the value of the heuristic function, indicate justifying, which node is expanded next using each of the following methods:

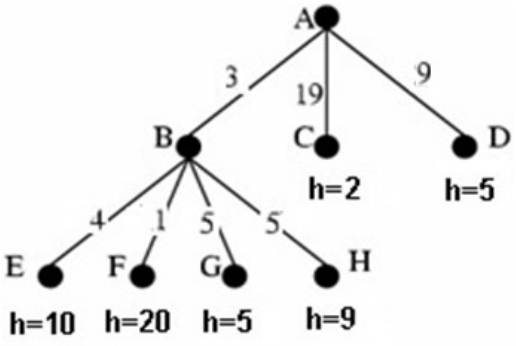
a) Breadth-First Search ("Pesquisa Primeiro em Largura"); C
b) Depth-First Search ("Pesquisa Primeiro em Profundidade"); E
c) Uniform Cost Search ("Pesquisa de Custo Uniforme"); D
d) Greedy Search ("Pesquisa Gulosa");
e) A* Algorithm Search ("Pesquisa com oAlgoritmo A*")



1.4 Strategies for Uninformed/Informed Search

Assuming the following search tree which node is expanded next using each of the following methods:

a) Breadth-First Search ("Pesquisa Primeiro em Largura"); C
b) Depth-First Search ("Pesquisa Primeiro em Profundidade"); E
c) Uniform Cost Search ("Pesquisa de Custo Uniforme"); F
d) Greedy Search ("Pesquisa Gulosa");
e) A* Algorithm Search ("Pesquisa com o Algoritmo A*")



1.5 Solving the Two Buckets Problem with Informed Search

a) Implement code to solve this problem using Greedy Search and using the A* Algorithm.
b) Compare the results achieved with the results from exercise 1.1 (uninformed search).

1.6 Solving the Missionaries and Cannibals Problem with Informed Search

a) Implement code to solve this problem using Greedy Search and using the A* Algorithm.
b) Compare the results achieved with the results from exercise 1.2 (uninformed search).

1.7 Solving the N-Puzzle Problem

The objective of this exercise is the application of search methods, with emphasis on informed search methods and the A* algorithm, to solve the well-known N-Puzzle problem. The desired objective state for the puzzle is as follows (0 represents the empty space):

```
9Puzzle          16Puzzle
 1 2 3           1  2  3  4
 4 5 6           5  6  7  8
 7 8 0           9 10 11 12
                13 14 15  0
```

Starting from a given initial state, the goal is to determine which operations to perform to solve the puzzle, reaching the desired objective state.

- **a)** Formulate the problem as a search problem indicating the state representation, operators (their names, preconditions, effects, and cost), initial state, and objective test.
- **b)** Implement code to solve this problem using the “breath-first” strategy (in this case identical to "Uniform Cost").
- **c)** Implement code to solve this problem using Greedy Search and using the A* Algorithm. Suppose the following heuristics for these methods: i) H1 - Number of incorrected placed pieces; ii) H2 - Sum of Manhattan distances from incorrected placed pieces to their correct places.
- **d)** Compare the results obtained concerning execution time and memory space occupied in solving the following problems using the previous methods:

```
Probl1     Probl2     Prob3        Prob4
1 2 3      1 3 6      1 6 2       5  1  3  4
5 0 6      5 2 0      5 7 3       2  0  7  8
4 7 8      4 7 8      0 4 8      10  6 11 12
                                  9 13 14 15
```