**MDL Assignment: Search**

* Eshika Khandelwal | 2020114018
  1. States:
     1. On(A, B) - Block A is on Block B
     2. OnTable(A) - Block A is on table
     3. Clear(A) - Nothing is on Block A
     4. Holding(A) - the robot arm holds A
     5. ArmEmpty() - the robots arm is empty
  2. Actions:
     1. Unstack(A, B) - remove block A from block B
     2. stack(A,B) - stack block A on block B
     3. Pickup(A) - pick A up using the robot arm
     4. PutDown(A) - put A on any free space on the table
  3. Initial state: The initial state would be the initial configuration given
  4. Goal test: The goal state would be the goal configuration given
  5. Path cost:
     1. 1 for every action i.e moving a block from its current position to another position
     2. 0 otherwise

The optimal solution would be the one where we reach the final/goal configuration using the minimal cost

1. Initial State:

[C ]

A B

Final State:

[A]

B

C

1. Breadth first
   1. First iteration

[B ]

C C C C

A, A B, A B C, A B, A B

* 1. Second iteration

[ B ]

C C C C C

A B, A B C, A B, A B, A, A B

* 1. Third iteration

Because the configuration (node) [ C ] is already visited in our BFS in the

A B

Initial iteration, we skip this configuration and move on to the next node.

[ B ]

C C C C B, B C C A A

A B, A B, A, A B, A C, A C B, A C, A B, A B, A B C, B C, B C, A B C

1. Depth-first
   1. First iteration

[B ]

C C C C

A, A B, A B C, A B, A B

* 1. Second Iteration

[ B ]

C C C C C

A, A B, A B, A B C, A B, A B

* 1. Third iteration

Because the configuration (node) [ B ] is already visited in our DFS in the

C

A

second iteration, we skip this configuration and move on to the next node.

Now,

Because the configuration (node) [ C ] is already visited in our DFS in the

A B

Initial iteration, we skip this configuration and move on to the next node. This happens twice, so the next node becomes [A B C]

[ B, B C C A A C C]

A C, A C B, A C, A B, A B, A B C, B C, B C, A B C, A B, A B

1. Uniform search: cost is the number of steps required to reach the configuration from the initial state.

i. First Iteration

[ A ]

C C C C

A B, A B, A B, B, C A B

Cost: 1 0 0 1 1

Because the configuration (node) [ C ] is already visited in

A B

first iteration, we skip this configuration and move on to the next node.

ii. Second Iteration

[ A A ]

C C C C C C C

A B C, A B, A B, A B, B, A B, A B, B, C A B

Cost: 1 0 1 1 1 0 0 1 1

Because the configuration (node) [ C ] is already visited in

A B

first iteration, we skip this configuration and move on to the next node.

Let us remove all the visited configurations (node) from the list:

[A ]

C

B, A B C

Cost: 1 1

iii. Third iteration

[A ]

C C

B, B A, A B C

Cost: 1 1 1

3. Let us consider these configurations to understand both the examples:

Current state:

[ C ]

A B

Final state:

[ C ]

A

B

Heuristic 1: Count all the blocks which are not in their correct position ( position as required in the final/goal configuration). The block which is in the arm of the robot will not be counted.

For the above example:

Neither A nor C is not in their correct position, so two is added to the heuristics.

Heuristic 2: Calculate the distance between the current state of a block and the final/goal state of the block. Details of each block will be looked at.

For the above example:

In the goal configuration A is supposed to be above B and below C, but in the current configuration it is neither above B nor below C and so 2 will be added to the heuristics. We do the same for the other two blocks.

4. We will use the first heuristics for this question.

The initial and final states used are the same as Q2.

Initial State:

[C ]

A B

Final State:

[A]

B

C

1. First iteration

[B ]

C C C C

A, A B, A B C, A B, A B

Cost: 3, 3, 2, 3, 3

We will expand the node with cost 2.

1. Second iteration

[ B ]

B, B C C A A C C C C

A C, A C B, A C, A B, A B, A B C, B C, B C, A B C, A, A B, A B, A B

Cost: 1, 2 , 2, 3, 3, 2, 2, 2, 2, 3, 3, 3, 3

Now, we will expand the node with cost 1 and remove all the visited nodes.

1. Third iteration

[ A B ]

B B B B C C A A C

A C B, A C, C, A C, A C, A B, A B, B C, B C, A

Cost: 2 1 0 1 2 3 3 2 2 3