Artificial Intelligence Assignment-2 Report

DESCRIPTION ABOUT THE BLOCKS WORLD DOMAIN

Blocks World Domain Game starts with an initial state consisting of a fixed number of blocks arranged in 3 stacks and we can move only top blocks of the stacks and we have to achieve a goal state that is a particular arrangement of blocks by moving these blocks.

1. **State Space:**   State is defined by 3 stacks which contains blocks.

2. **Start node:** It's a state to initialize the algorithm. e.g.,

   E,B,F

   D,A

   C

   where , "," represents push operation in stack and new line means new stack.

3. **Goal node**: It's final state to end the algorithm. e.g.,

   A,D,B
   E,F,C

4. **MOVEGEN algorithm:**   Iterate over all the top elements of 3 stacks then pop an element from one of the stack and push it in one of rest of the stacks which gives a state that represents the successors of the given state.

5. **GOALTEST algorithm:** Iterate over all the elements of stack in current state and in final state then compare each . If all of the 3 stacks found equal then return true otherwise return false.

HEURISTIC FUNCTIONS

Function that gives an approxiamate value for the given problem. Let's consider the states as follows:

Initial state: [E,B,F],[D,A],[C]   Goal state: [A,D,B] , [E,F,C]

Heuristic1: Add 1 for every block which is at the correct position as goal state otherwise subtract 1. E.g.,

Heuristic of initial state= -1-1-1-1-1-1 = -6

Heuristic of goal state = +6

Heuristic2: Add height upto the block if it is at the same position as in goal state otherwise subtract .

Heuristic of initial state = -1-2-3-1-2-1 = -10

Heuristic of goal state = 1 +2+3+1+2+3 = 12

Heuristic3: Get the coordinates of every block in current state and final state the find the Manhattan distance (d = |x2-x1| + |y2-y1|)

We can minimize the Manhattan distance or maximize (some large value – d). e.g.,   Let's take 100-d, then

Heuristic of initial state = 100 -11 = 89

Heuristic of final state = 100-0 = 100

HILL CLIMBING

Hill Climbing is a heuristic search used for mathematical optimization problems in the field of A.I. Given a large set of inputs and a good heuristic function , it tries to find a optimal solution to the problem.

- ▪ Start from initial state if it is not the goal state mark it as visited then get its neighbors and do the same process for the next state which is the optimized neighbor based on the heuristic function's value.

Test Cases (input.txt)	Heuristic Funtion	No. of states explored	Time Taken (ms)	Optimal Solution
1.	1	5	7.9	No
2.	1	2	1.4	Yes
1.	2	4	7.8	No
2.	2	2	4.0	Yes
1.	3	3	3.8	No
2.	3	2	2.6	Yes

The above table gives the no. of states explored, time taken and test of optimal solution for input.txt which contains two test cases.