INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Artificial Intelligence

Report Lab 3

Team:

Shara Teresa González Mena A01205254

Bernardo Gómez Romero A01209704

Due Date:

September 21, 2018

Which heuristics did you use for the A* algorithm?

For A* Algorithm we used 3 different heuristics:

1. The first heuristic consisted in counting the number of misplaced blocks so that the closer to the goal, the smaller the value returned by the heuristic function.
2. This heuristic consisted in removing the constraint of just moving the topmost block, so that for each displaced block, we got the cost of moving it from its stack to its corresponding stack goal (no matter in which position of the stack) and calculated the cumulative cost of ordering all the blocks by the initial rule: 0.5 up, 1 for each lateral movement and 0.5 down.
3. Our third heuristic was an inconsistent one, we used the same logic as the first one and add an arbitrary value to it: the number of stacks in the problem times 3.

Test your program with a couple of different problems. Increase the size of the problem to test the limits of your program. Make a table comparing **how many nodes are searched** to find the answer for each problem. For this table, you should compare a number of different problems (at least 3) to avoid a statistical bias. Which of the three algorithms (UCS, A *with consistent and and A* with an inconsistent heuristic) searches the least nodes and which one take the most?

Test input 1:

3
(A, D); (B); (C); (E)
(X); (D, A); (C, B); ()

Test input 2:

4
(A, B, C); (); (D, E, F);
(A, C, F, D); (E, B); ();

Test input 3:

2
(A, B); (F, C); (D, E);
(F, A); (X); (E, C);

	Test 1			
Algorithm	UCS	A* H1	A* H2	A* H3
Cost	10	10	10	10
Nodes Visited	354	106	47	106
Time (Sec)	0.14902	0.02418	0.00850	0.02398
	Test 2			

Algorithm	UCS	A* H1	A* H2	A* H3
Cost	19	19	19	19
Nodes Visited	14703	4506	1772	4506
Time (Sec)	324.46170	68.34113	11.15173	54.89843
	Test 3			
Algorithm	UCS	A* H1	A* H2	A* H3
Cost	22	22	24	22
Nodes Visited	2483	1507	1434	1507
Time (Sec)	3.02459	1.22230	1.18096	1.22382

Why does this happen?

Because A* search algorithm uses extra information to take decisions about which node to expand. This extra information, heuristic, is an approximation of how far (steps) is a given state from the goal, so by adding this heuristic value, the algorithm will avoid expanding those nodes that are farthest from the goal and which.

In the other hand, the uniform-cost search algorithm just takes into account the cost of the path and expands the node with less cost which is not a guarantee that the chosen node is closer to the goal. This algorithm find the optimal path but after expanding all the nodes (ordered by cost) until it finds the goal.

Which algorithms are optimal? Why?

Both are optimal because A* Search (with a consistent heuristic) and UCS find the optimal path, the one with less cost. The difference is that the execution time increases with UCS and also the memory needed to store all the expanded nodes.

Therefore A*search is better.

A* Search using a inconsistent heuristic didn’t find the optimal path, so that is not optimal.

In your opinion, what are the benefits of simpler algorithms versus more complex ones?

They are easier to implement and can get to the goal. If you are not concerned about memory and time of execution, is a good option.