Intelligent Systems

Lab 3: (Un)informed Search Lab

José Ramón Romero Chávez (A01700318)

Tecnológico de Monterrey, Campus Querétaro

4th March 2019

- Which heuristics did you use for the A* algorithm?
  - The sum of the distance between each element and its goal position
  - Example:
    - Current State -> (A,B); ( ); ( )
    - Goa State -> ( ); ( ); (A, B)
    - H(Current State) -> 4 = (A=abs(0-2))+ (B=abs(0-2))

- 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.

	UCS (#Nodes, Cost, Path)	Acceptable A* (#Nodes, Cost, Path)	Not- Acceptable A* (RandInt [50,100], with seed = 50) (#Nodes, Cost, Path)
Test1 3 (A); (B); (C); () (); (A); (B); (C)	40 6 (2, 3); (1, 2); (0, 1)	12 6 (2, 3); (1, 2); (0, 1)	68 6 (2, 3); (1, 2); (0, 1)
Test2 2 (A); (B); (C) (A, C); X; X	7 3 (2, 0)	2 3 (2, 0)	13 3 (2, 0)
Test3 2 (A); (B); () (A, B); X; X	3 2 (1, 0)	2 2 (1, 0)	10 2 (1, 0)
Test 5 3 (B, A); (C, D, E); () (E); (C, B, A); (D)	453 17 (1, 0); (1, 2); (0, 2); (0, 2); (0, 1); (2, 1); (2, 0)	225 17 (1, 0); (1, 2); (0, 2); (0, 2); (0, 1); (2, 1); (2, 0)	881 21 (1, 0); (1, 2); (0, 2); (0, 1); (0, 2); (1, 0); (2, 1); (0, 1); (2, 0)
Test 6 3 (C); (D); (A); (B) (D, A, B); (); (); (C)	719 15 (0, 2); (1, 0); (3, 1); (2, 3); (2, 0); (1, 0)	214 15 (0, 2); (1, 0); (3, 1); (2, 3); (2, 0); (1, 0)	681 18 (0, 2); (1, 0); (3, 1); (2, 1); (2, 0); (1, 3); (1, 0)
Test 7 3 (A); (B); (C); (D) (B, D, C); (); (A); ()	741 16 (2, 1); (0, 2); (1, 2); (1, 0); (3, 0); (2, 0)	321 16 (0, 2); (1, 0); (3, 0); (2, 1); (2, 0); (1, 2)	242 21 (1, 3); (0, 3); (2, 1); (3, 2); (3, 0); (3, 0); (1, 0)

- 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?
    - Least: A* with consistent heuristic
    - Most: A* with and inconsistent heuristic
- Why does this happen?
    - Because an incorrect heuristic may ignore the correct path as it will be added at the end of the priority queue, on the other hand a correct heuristic will put the correct path on the beginning of the queue translated to less iterations.
- Which algorithms are optimal? Why?
    - A* with consistent heuristic and UCS as they find the best solution always by looking for the shortest path in each iteration and the A* will reduce the amount of these by affecting the priority of the paths
- In your opinion, what are the benefits of simpler algorithms versus more complex ones?
    - The simpler algorithms, are easier to implement with the side effect that they may not be optimal, also I think that they are more used to go through the nodes without knowing about cost or distance.