Search-Based Path Finding

Description: research on A* searching algorithm, in which different heuristic functions were implemented, including Manhattan distance, Euclidean distance and Diagonal distance. The effect of Tie Breaker was also examinated.

When heuristic is 0, the algorithm is Dijkstra, which can always get the optimal path but has more visited nodes.

Manhattan distance = dx + dy, which is fast and has very lest visited nodes in test cases, but sometimes the output path is not optimal.

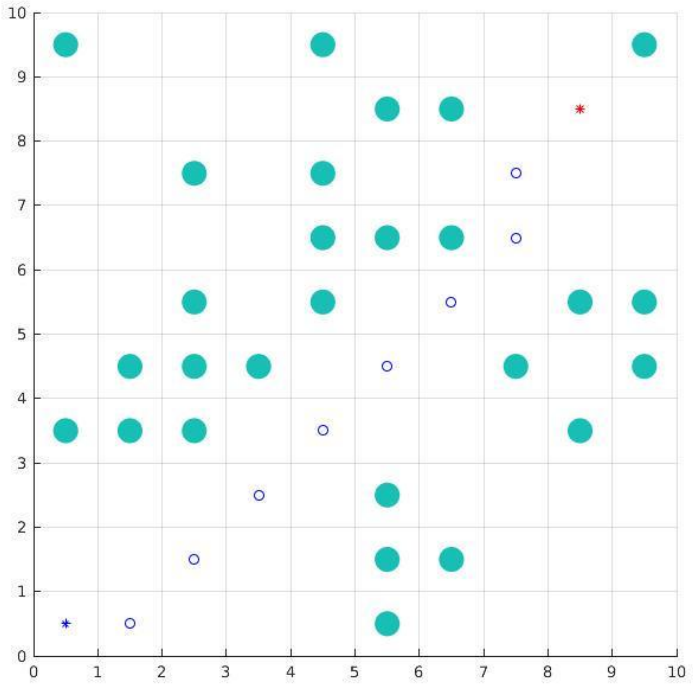
Euclidean distance = sqrt(dx ^ 2 + dy ^ 2), which is less fast than by Manhattan distance, but admissible.

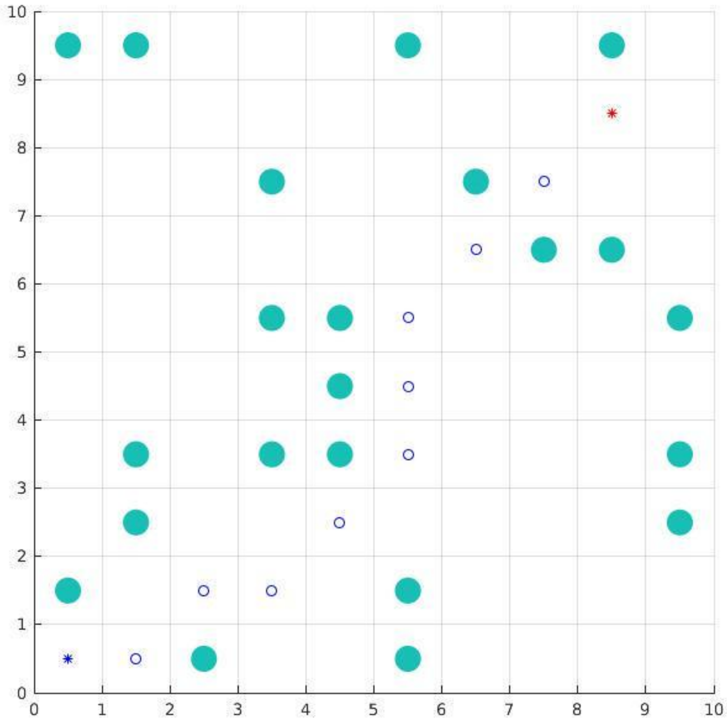
Diagonal distance = dx + dy + (sqrt(2) - 2) * min(dx, dy), which has better performance than by Euclidean distance in the test cases and tight to the real path.

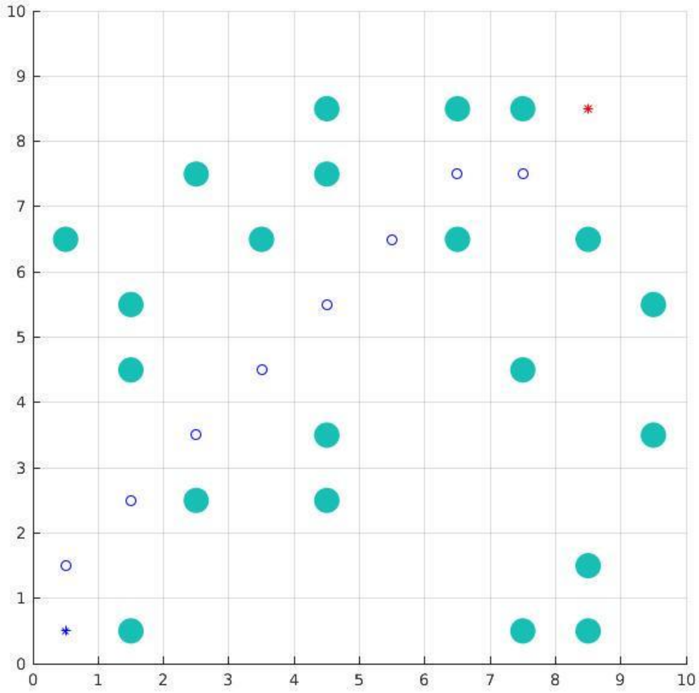
To add a tie breaker, just multiply the heuristic cost by a factor, which is

expected min step cost / expected maximum cost. Here choose 1 / dx * dy. The output time consummed is shorter and got less visited nodes in test cases.

Outputs of 3 random obstacle map cases:




0
1
2
3
4
5
6
7
8
9
10


0
1
2
3
4
5
6
7
8
9
10