Report: Implementing ANA* algorithm
-Amey S Kulkarni

ANA*(Anytime Non-parametric A*) is a variant of the standard A* algorithm. It is a greedy algorithm and helps to find an optimal solution in less amount of time. The name suggests that it's an anytime algorithm. This algorithm quickly produces a sub-optimal solution and then improves it over time.

The major advantage of ANA* over other anytime algorithms is that it does not require other ad-hoc parameters to make calculations. The heuristics function is scaled with a quantity which depends on the standard parameters.

$$f(s)=g(s)+\varepsilon\, h(s)$$

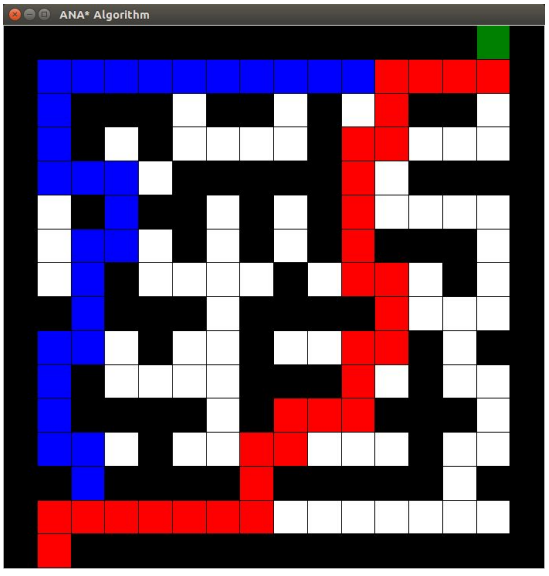
Epsilon here is the factor which is defined as :

$$\varepsilon = (G - g(s))/h(s)$$

G is the cost of the current best solution. g(s) is the cost till current node and h(s) is the heuristic function.
Initially, the cost G is kept infinity. The first solution is calculated quickly and then it is improved over time. The following figures explain the solutions found in the easy,hard and my maps. Only four neighbor connections are exploited.

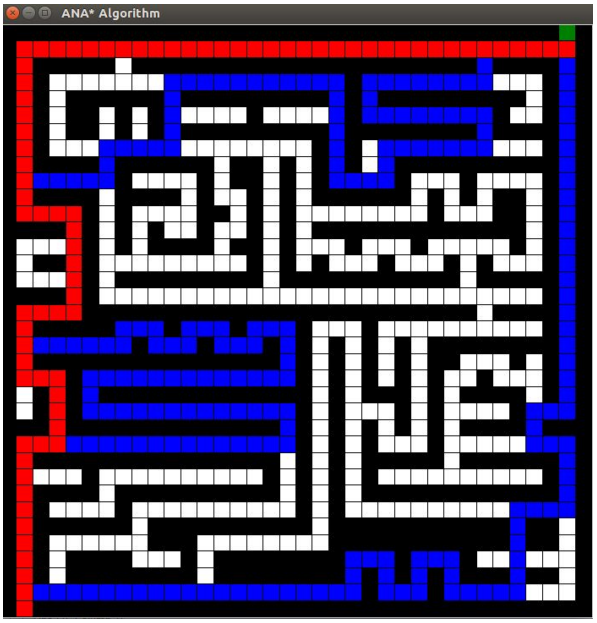
Color coding:
Green color- Destination
Red color- Final optimal solution obtained
Blue color- all feasible solutions
Black- obstacles
White- Free area

1. Easy map



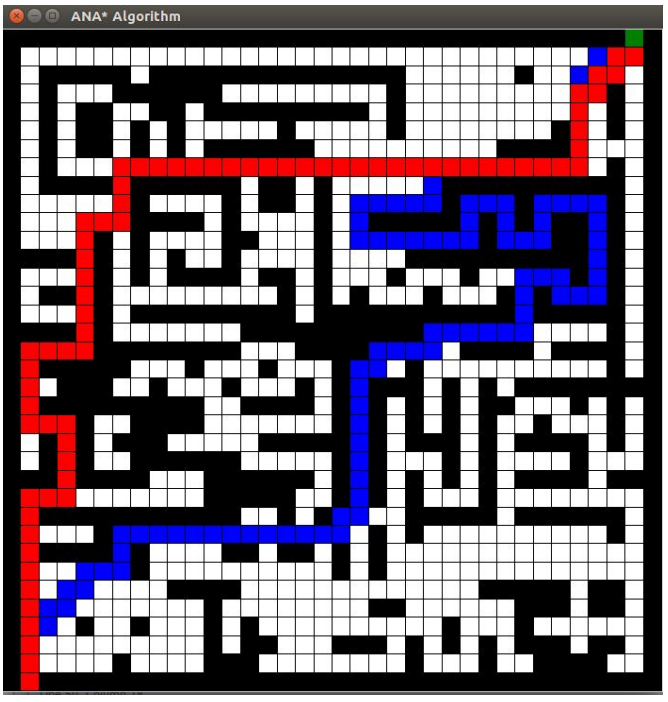
Total number of nodes in all solutions combined:64
Total time taken to find the optimal solution:0.0687s

2. Hard map



Total number of nodes in all solutions combined:424
Total time taken to find the optimal solution:1.023s

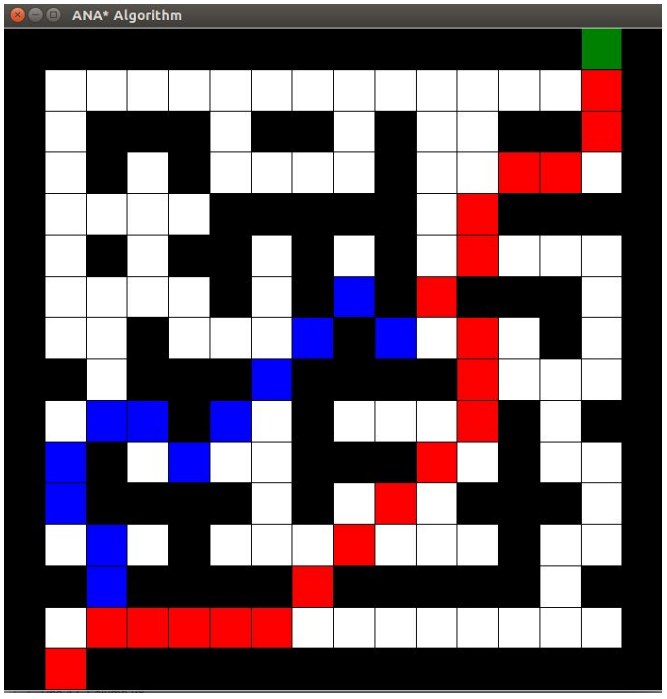
3. My_map



Total number of nodes in all solutions combined:176
Total time taken to find the optimal solution:0.911s

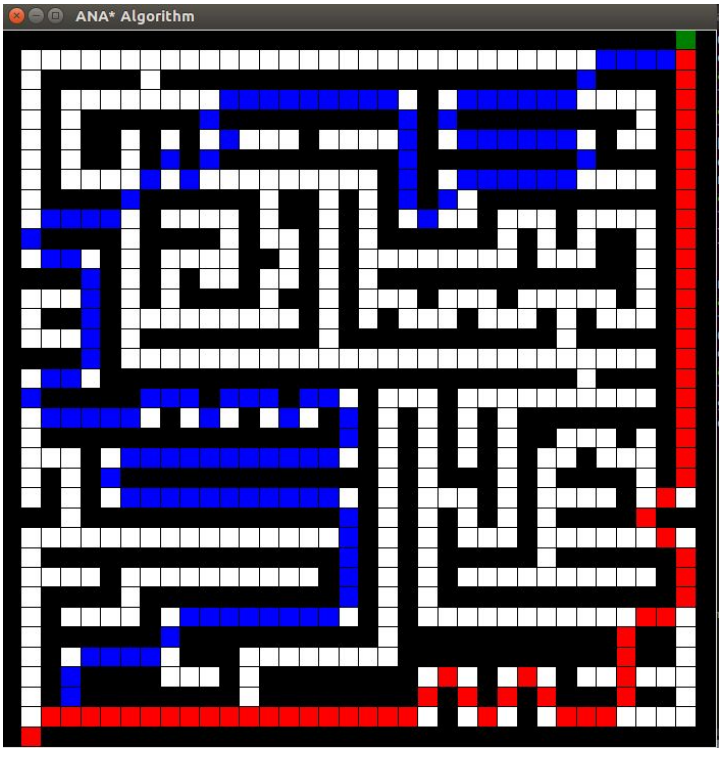
Another experiment completed is with 8 connected neighbors. The cost of moving into diagonally opposite neighbors will obviously be more than moving between horizontally or vertically connected edges but the number of nodes explored is much lesser which reduces the overall cost.

1. Easy map



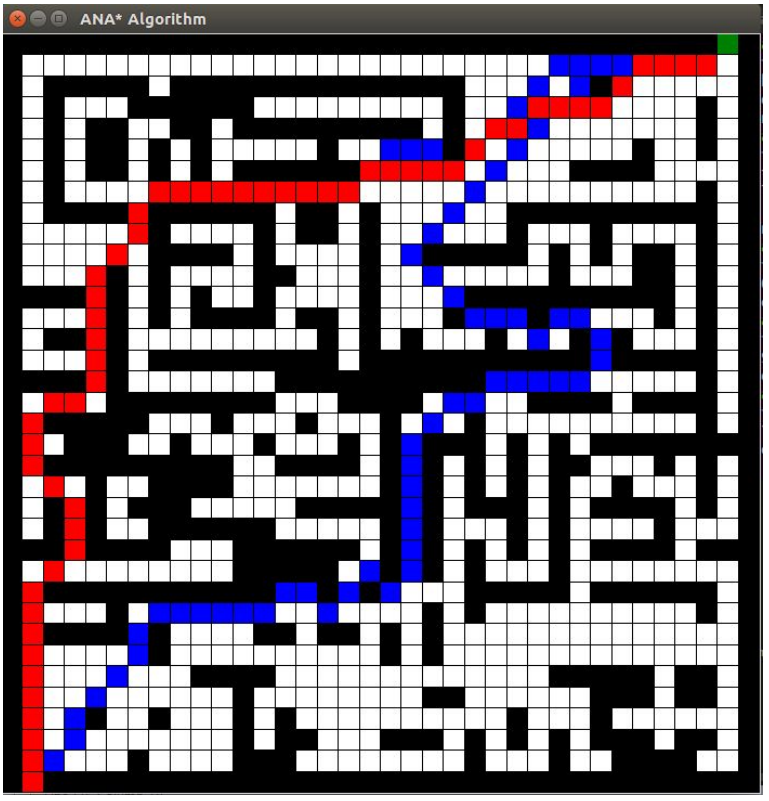
Total number of nodes in all solutions combined:41
Total time taken to find the optimal solution:0.087s

2. Hard map



Total number of nodes in all solutions combined:253
Total time taken to find the optimal solution:1.96s

3. My_map



Total number of nodes in all solutions combined:172
Total time taken to find the optimal solution:2.73s

It is observed that even though the number of nodes used for solution decreases, time for computation increases much more which is undesirable. Hence the use of 8-connected neighbors need to be thought about for specific scenarios.