University of Girona

Spain

Autonomous Robotics
Lab 5 - Graph Search - A* Algorithm

Submitted by :
Nayee Muddin Khan DOUSAI

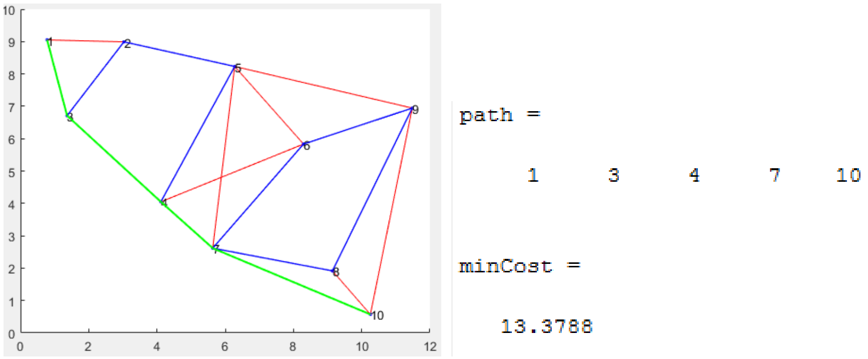
1 Introduction

In this lab, we are going to implement and discuss the implementation details of graph search A- Star algorithm. This algorithm is used for path planning and graph traversal and its been executed in MATLAB. The required input for A- Star algorithm is obtained from rotational plane sweep (RPS) algorithm from the Lab 4 or directly pass the edges. In our case we are testing with the given edges from the practical Lab 5 guide.

2 A- Star Algorithm

A- Star Algorithm is used to find the optimal path from the present position to the goal position by using visibility graph, with all constrained parameters. By using A-star we can implement the heuristic guide to perform the best first search around the visibility graph. Visibility graph is built in our Lab 4, this approach is used in path planning. A-star is considered as one of the best optimal path invented by Nils Nilson while working with shakey robot. A-star algorithm is based on the shortest path finding with respect to Dijkstra's Algorithm which prefers vertices which are close to the starting point.

At first, A-Star considers all the nearest vertices by using weighted function and choose the path which has the lowest cost. It counts the vertex with the lowest cost and continues the tree till goal position till it finds the lowest path. Our approach to find this is explained as, we have calculated the heuristic distance for all the vertices and stored in a matrix. We will use two different lists in our algorithm named as *Olist*, this express set of nodes to explore and *Clist*, this gives the explored nodes. We will calculate all the costs from the starting point to the connected vertices and are inserted into Open list including indexes and their parent point while the lowest cost is stored in the other parent point. The index of this are stored into closed list, which is used to find the shortest path. This process of updating goes on till we reach the optimal goal path.

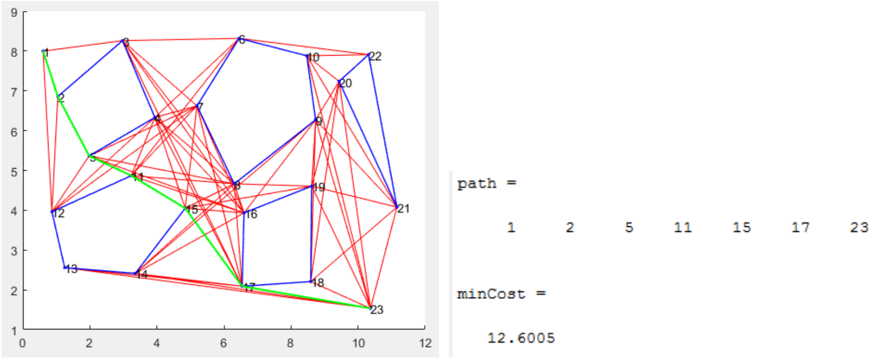


```
path =

     1     3     4     7    10


minCost =

   13.3788
```

Figure 1: A- star algorithm for simple convex obstacle including the optimal path distance

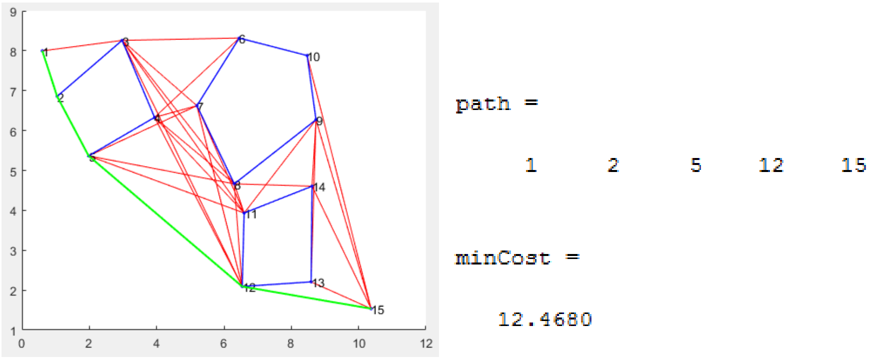
3 Results

For the lab 5, we have implemented A-Star algorithm from the pre-defined vertices as *main.m* file whereas the A-Star Algorithm is written as*Astar.m*file. The *main.m* file calls the A-star file, as the vertices and edges are not described in the algorithm.



```
path =

     1     2     5    11    15    17    23


minCost =

   12.6005
```

Figure 2: A- star algorithm for complex convex obstacles including the optimal path distance



```
path =

     1     2     5    12    15


minCost =

   12.4680
```

Figure 3: A- star algorithm for different convex obstacles including the optimal path distance

4 Conclusion

In this lab, we have implemented A-Star Algorithm for optimal path planning from the given vertices and edges. The results are evaluated and labelled in graph and the attached matlab files. It has implemented on small and large environments. A-Star algorithm limitation is, it can be applied only when we have the given edges.