Assignment Report – Flatlands

Motion Planning, in a 2D Grid Environment; A Comparative Analysis of Planning Algorithms
In this project I investigated the effectiveness of four planning algorithms. Depth First Search (DFS) Breadth First Search (BFS), Dijkstra's Algorithm and a Random Planner in navigating a 2D grid environment. The grid environment consisted of a 128x128 matrix with scattered obstacles of varying densities. The objective was to discover a path from the left corner to the bottom right corner while avoiding these obstacles.

Approach

I implemented these four algorithms using Python programming language utilizing libraries such as NumPy for matrix operations and Matplotlib for visualization purposes. The grid environment was initialized with obstacle densities ranging from 0% to 75% increasing in increments of 5%. For each density level I executed the four algorithms to find a path from the starting point to the destination. Additionally, I tracked the number of iterations performed by each algorithm until reaching either the goal or determining that no viable path existed.

Findings:

The performance of these algorithms exhibited variations based on obstacle densities. At densities (10%) all algorithms except, for the Random Planner successfully discovered a navigable path.
However, when the density reached 40% none of the algorithms were able to find a path suggesting that there were obstacles blocking the way, to the goal. The Random Planner consistently. Frequently failed to find a path at lower densities.

Figures:

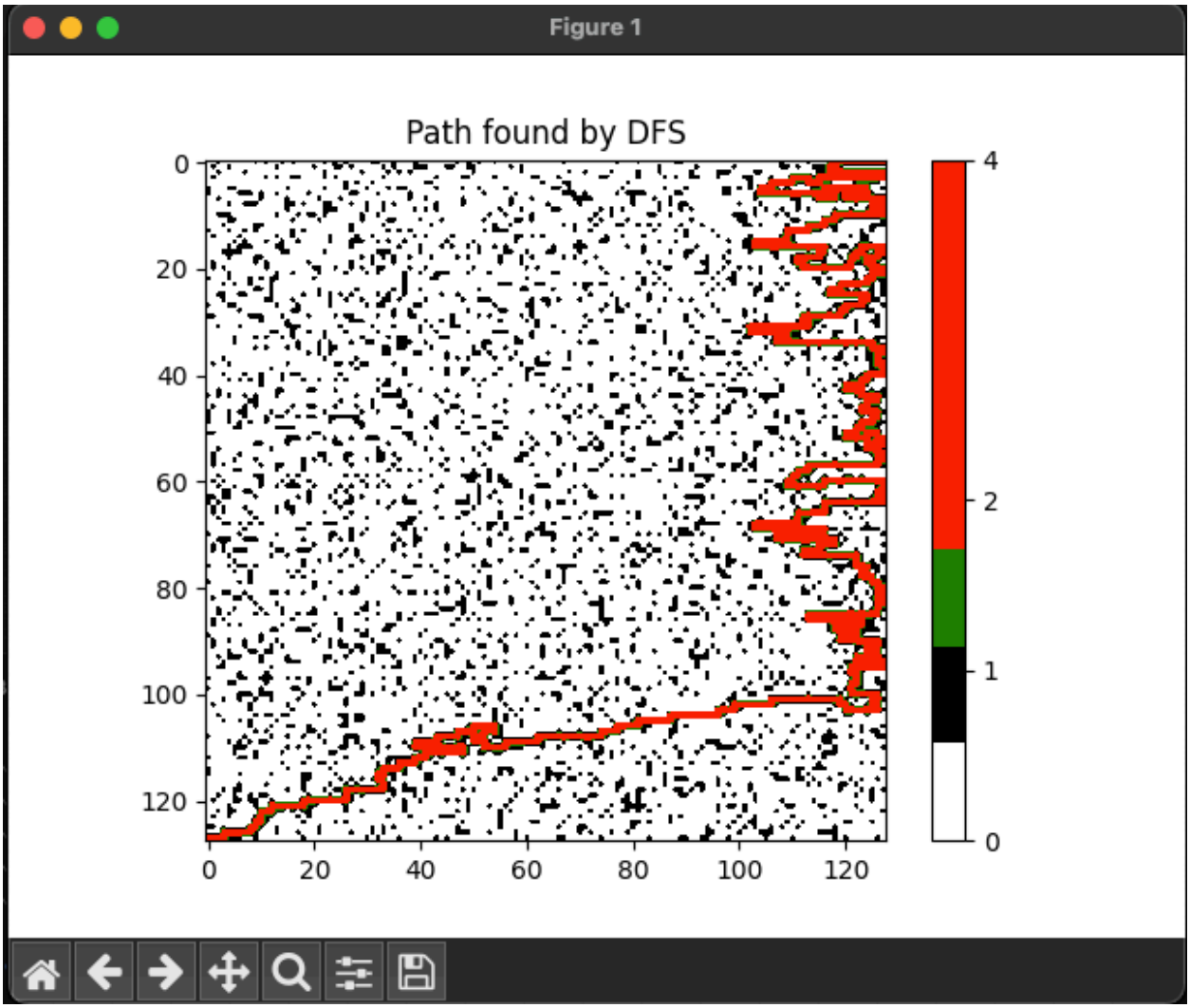


Figure 1
Path found by DFS
0
20
40
60
80
100
120
0
20
40
60
80
100
120
4
2
1
0

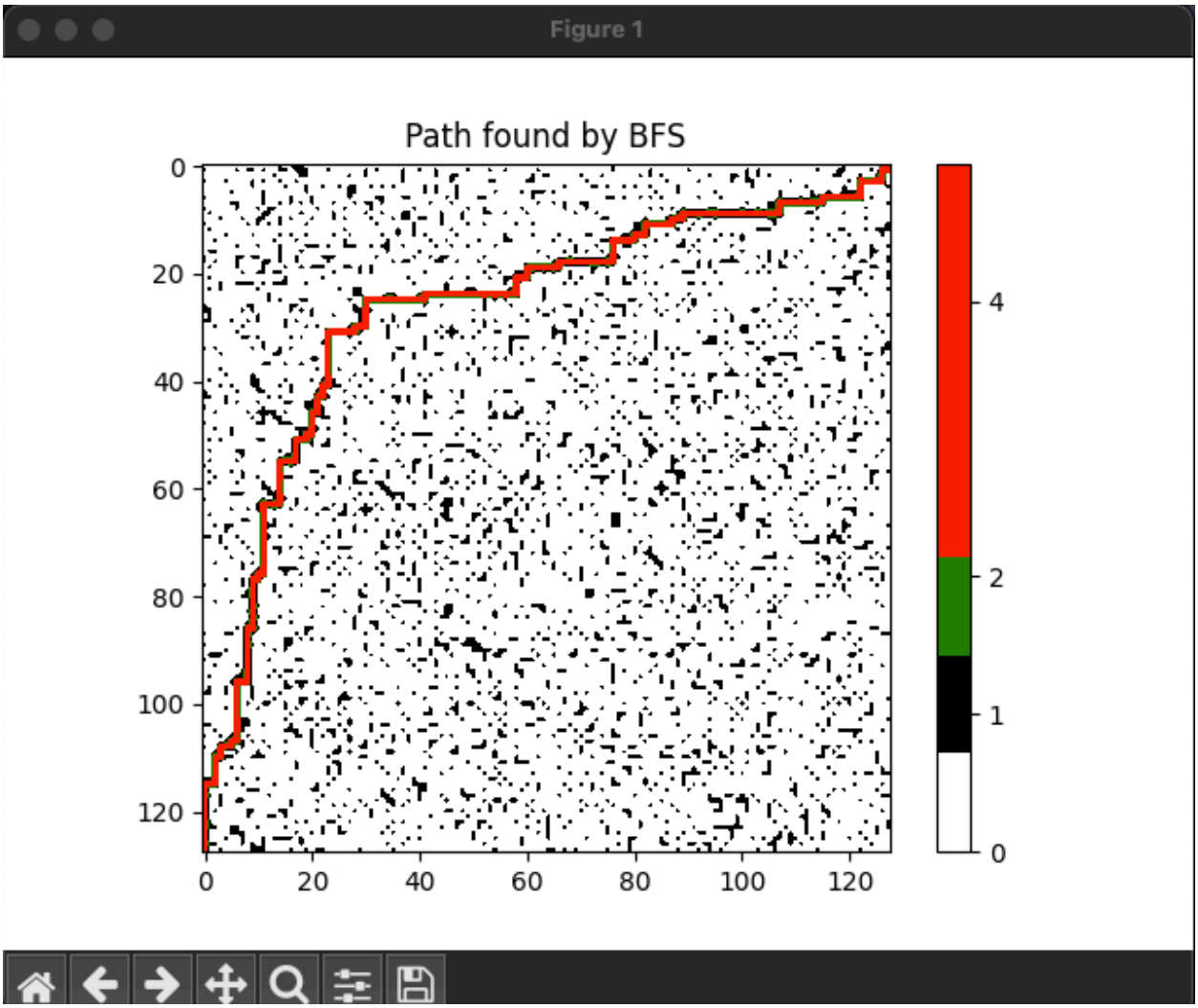


Figure 1
Path found by BFS
0
20
40
60
80
100
120
0
20
40
60
80
100
120
4
2
1
0

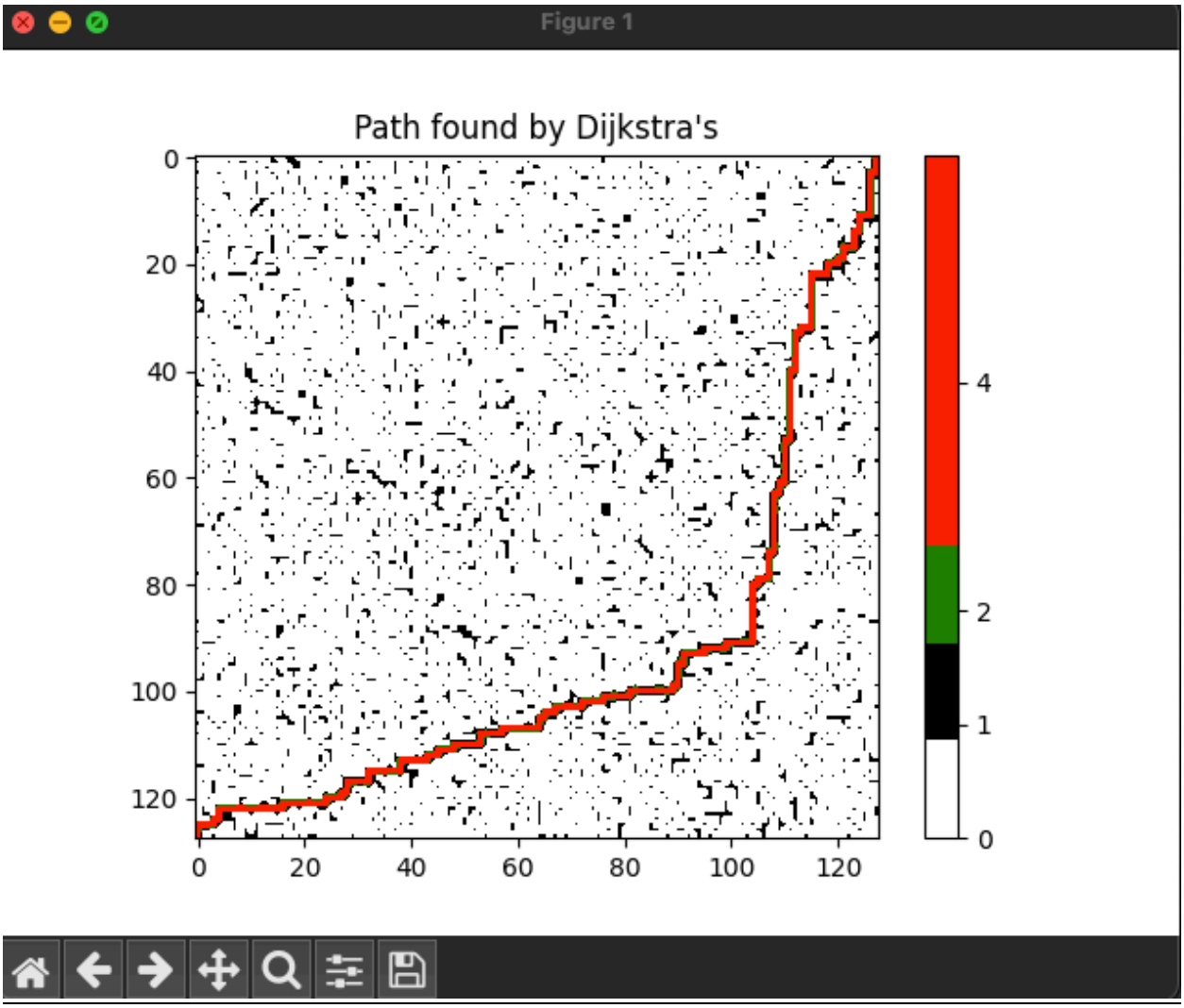
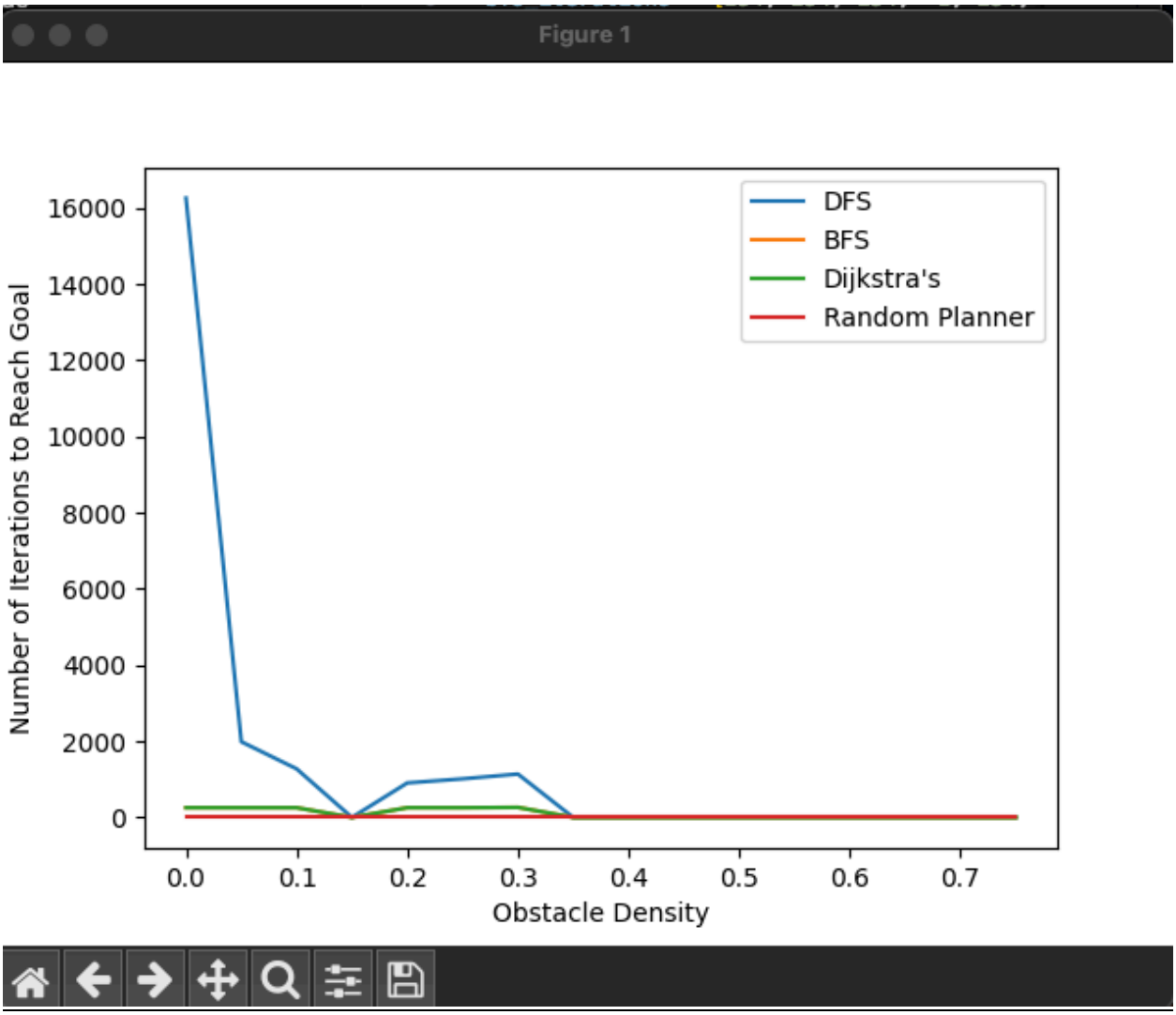


Figure 1
Path found by Dijkstra's
0
20
40
60
80
100
120
0
20
40
60
80
100
120
4
2
1
0



Discussion

The inability to find a path at densities was expected since its highly probable that obstacles obstructed all routes leading to the goal. The Random Planners subpar performance can be attributed to its lack of an organized search strategy, which diminishes its reliability. Future endeavors could involve implementing algorithms such as A* and RRT or conducting multiple simulations, for each density in order to obtain average performance metrics.