Jack O'Neill
AE 5222
Homework 7

Method

For this problem I used the Dijkstra's algorithm function written by Joseph Kirk which can be found at https://bit.ly/2Zzu1uG. I modified it slightly in order to be able to access the number of iterations for which the algorithm executed.

To complete this task I first needed to create an adjacency matrix of the graph, defining adjacency as the nodes to the right, left, above, or below any given node. Next I created n edge matrix which contains a pair of nodes that form an edge, along with that edge's respective cost which was calculated using the cost function defined in `hw2_2017_main.m`.

I could then input this edge cost matrix to Dijkstra's algorithm along with the desired start node of 1 and end node of N^2_G. This would output the optimal path, its total cost, and the number of iterations taken to find this optimal path. The results given different values of N_G are shown below.

Results

For N_G = 3, the following results are shown:

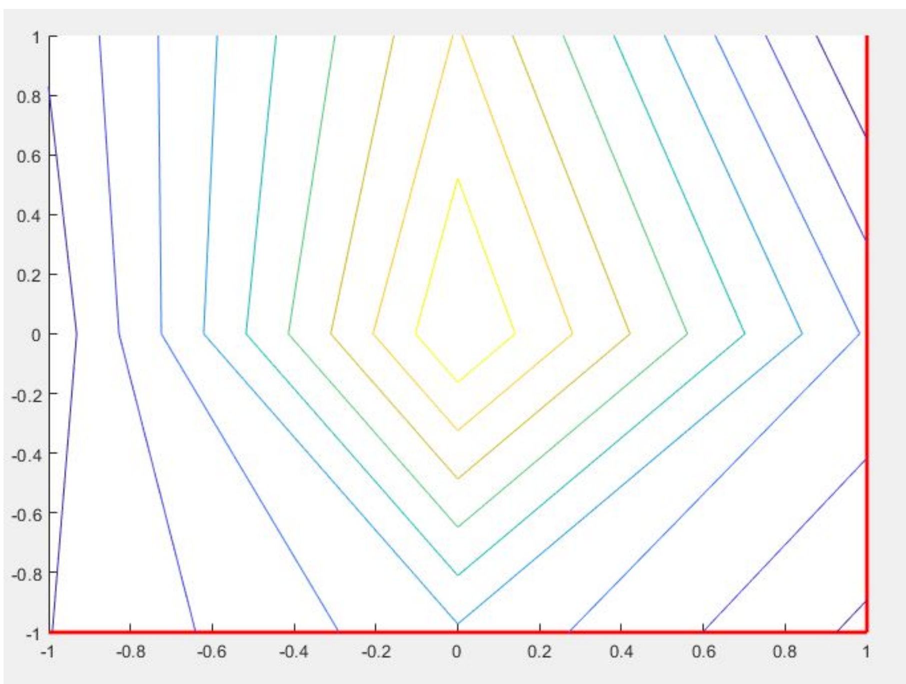

Figure 1: Optimal path for N_G = 3

For N_G = 3, the path cost = 12.609, and number of iterations = 9.

For N_G = 15, the following results are shown:

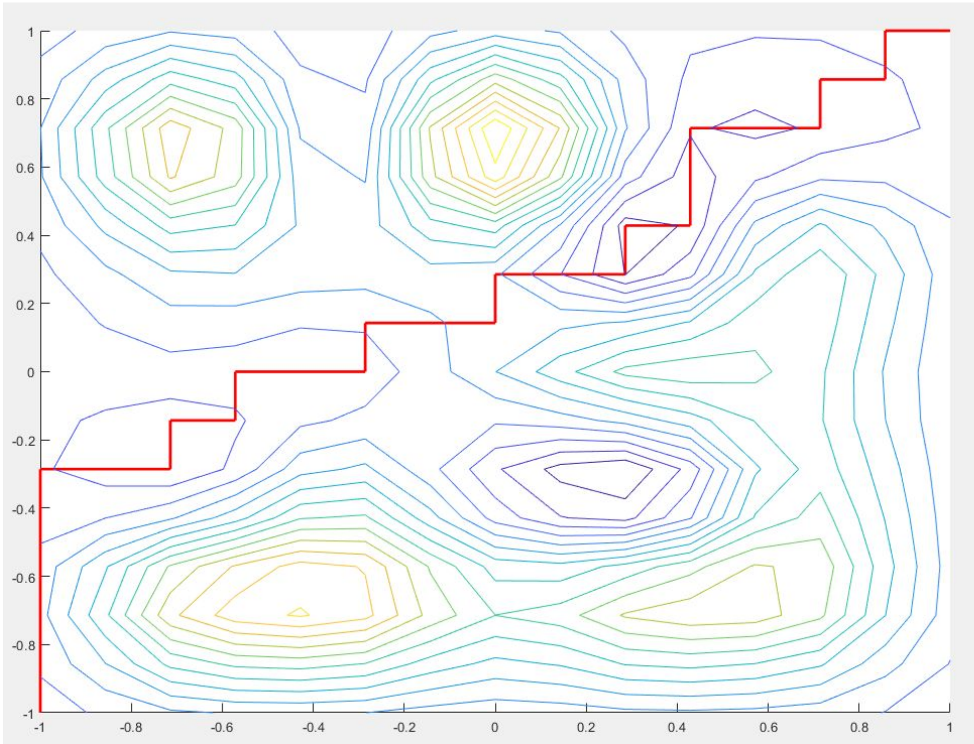

Figure 2: Optimal path for N_G = 15

For N_G = 15, the path cost = 80.559, and number of iterations = 225.

For N_G = 101, the following results are shown:

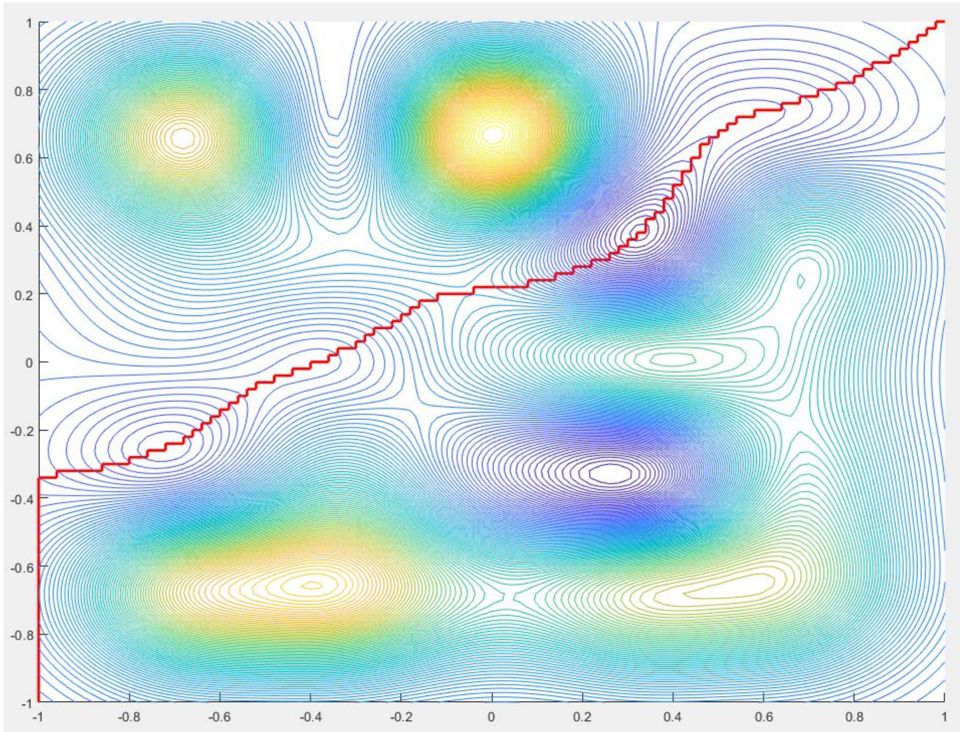

Figure 3: Optimal path for N_G = 101

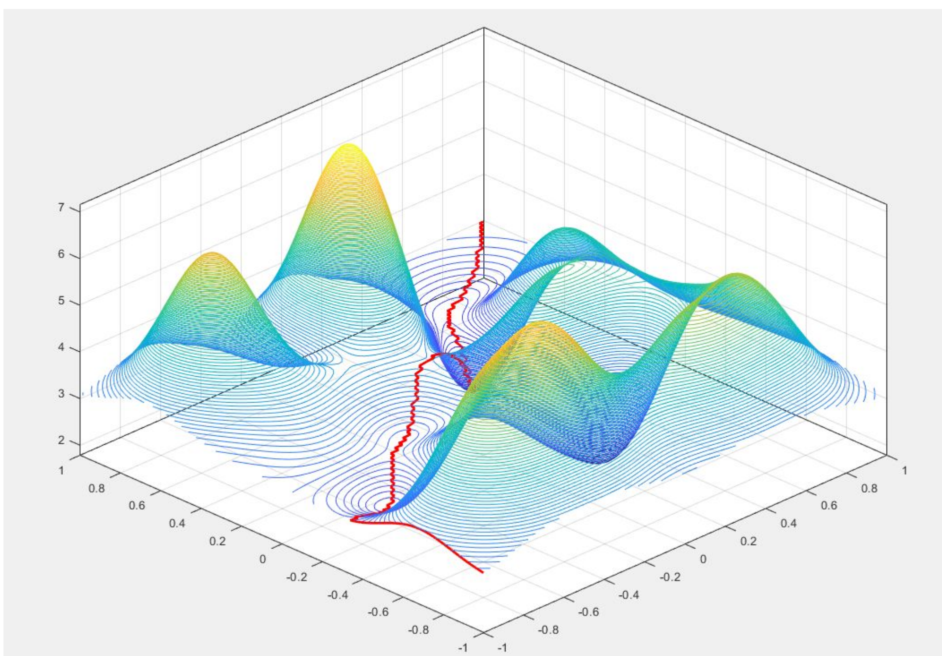

Figure 4: Another visualization of the optimal path for N_G = 101

For N_G = 101, the path cost = 566.013, and number of iterations = 10201.

Discussion

The optimal paths calculated in Matlab are visually apparent that their total costs are minimized. The results therefore confirm the correct determination of the adjacency matrix as well as the correct operation of Dijkstra’s algorithm.