

Scientific Computing (M3SC) Project 1

Omar Haque

February 19, 2017

1 MAIN SOLUTION CODE

```
1 # imports
2 import Dijkstra as dijk
3 import misc
4 import numpy as np
5 import csv
6 import matplotlib.pyplot as plt
7
8 def next_node(path):
9     """ Returns the next index (after the node itself) in the path.
10         If the path contains only one node, returns the node itself.
11     """
12     if len(path) == 1:
13         return path[0]
14     else:
15         return path[1]
16
17 def update_weight_matrix(epsilon, c, original_weight_matrix, noNodes=58):
18     """
19     This function updates the weight matrix according to step 5 of the
20     Project. Note the added fix – the weight matrix is not changed if
21     the original entry was 0.
22
```

```

23
24
25     :param epsilon: given in question
26     :param c: the vector containing number of cars at each node
27     :param original_weight_matrix: the weight matrix given by RomeEdges
28     :param noNodes: number of nodes in the system
29     :return: the updated weight matrix
30     """
31     new_weight_matrix = np.zeros((noNodes, noNodes))
32     for i in range(noNodes):
33         for j in range(noNodes):
34             if original_weight_matrix[i, j] != float(0):
35                 new_weight_matrix[i, j] = original_weight_matrix[i, j] +
36                                         (epsilon * (float(c[i]) +
37                                                         float(c[j]))) / flo
38     return new_weight_matrix

```