EE 360C

Xiangxing Liu

xl5587

Professor Touba

TA: Mona Mishra

Lab 2 Report

1. Proof

   a) Correctness of GPSR Algorithm

   According to the definition of Greedy Algorithm, the GPSR Algorithm introduced in this lab is a kind of Greedy Algorithms. However, GPSR Algorithm is not an optimal solution. Suppose that there exist two paths from an arbitrary source vertex and sink vertex in a graph. One path is the straight line from source to sink on the Cartesian coordinate, which should be the optimal path from the source to sink. The first connecting vertex of this path starting from the source is within the transmission radius.

   In contrast, the other path is not a straight line and the distance is from source to sink is longer. But the first connecting vertex of this path starting from the source is right on the edge of the transmission radius.

   Thus, the first vertex on the second path (non-optimal) will be chosen by GPSR since it is closer to the sink vertex. Hence, the path chosen by GPSR algorithm is not optimal.

   b) Correctness of Dijkstra Algorithm

   Dijkstra path hops solution is better than or equal to GPSR since it optimizes for path hops. But it is still not optimal because this method ignores the weight between vertices. The Dijkstra path hops method is able to find a path of shortest distance in terms of hopes. However, it is not efficient regarding latency among vertices. Therefore, if the weights among all vertices in the graph are all the same, then the Dijkstra path hops would find the optimal solution. But if there is different weight/latency, then the Dijkstra path hopes would not find the optimal solution.

2. Efficiency Analysis

   a) Memory Efficiency of Graph Representation

      The data structure used in this assignment is a map of vertices to a map of vertices to corresponded edges. Thus, for each vertex in the map, there are a set of edges to its neighbors within the range.

      The memory of the graph representation is O(V * E).

   b) Runtime Analysis

      The runtime complexity of my function that generates the graph by putting each edge into the map, and it takes E edges to generate.

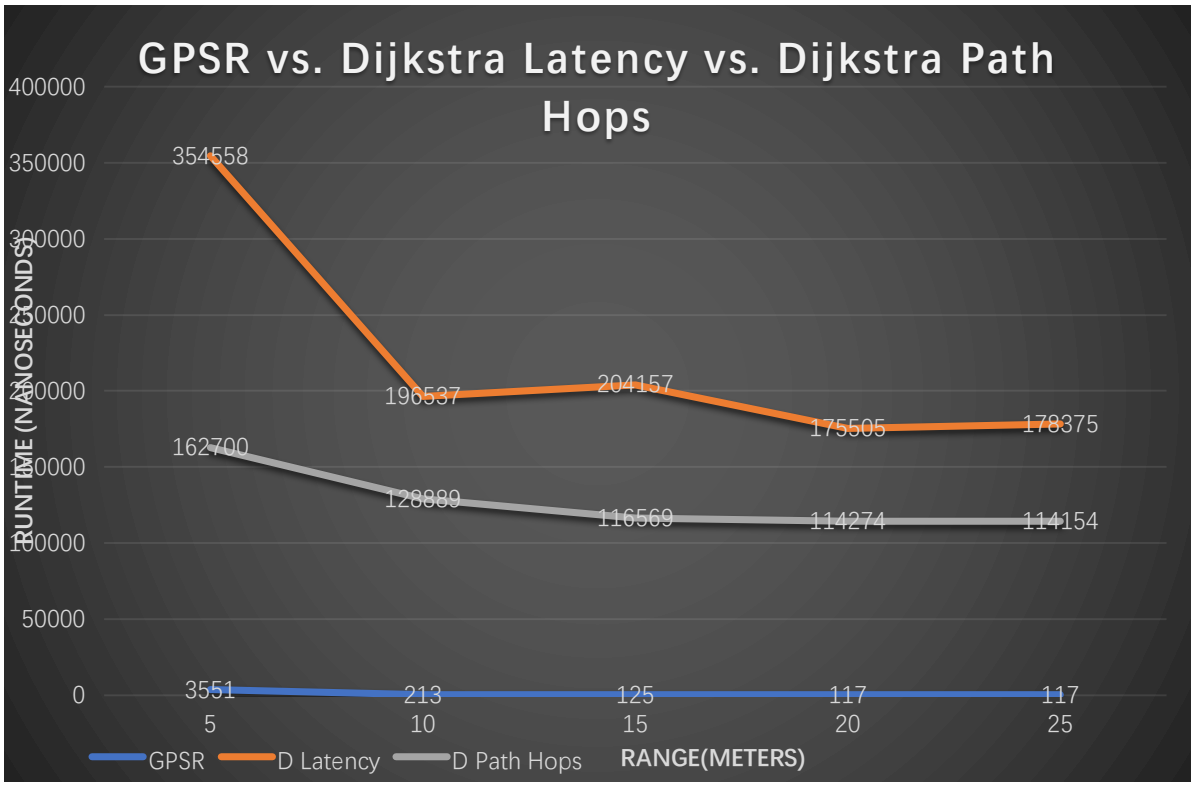
      Therefore, the runtime of this is O(E).

   c) Dijkstra Runtime Analysis

      The Dijkstra method iterates thru all vertices, and for each vertex, it goes thru all edges of this vertex. In case of the worst case – a fully connected graph, there are total of (V - 1) edges.

      The overall complexity of my Dijkstra algorithm is O(V^2).

3. Runtime Efficiency and Success Rate



The success rate is 100% for all test inputs. In concept from the algorithm analysis, the increase of range should increase the runtime of the program. But my results have a decrease in runtime for larger range. I think the change in runtime is caused by my computer since there is a significant decrease for the first run, and all three methods have decreased together.

Base on the runtime plot, the GPSR would be the one method to choose because it is much slower than the other two. The two Dijkstra methods does have similar runtime(in the same scale) so I would choose the Latency one because there is not a lot slower than the Hops one.