Lab 2 Report

Zi Zhou Wang (zw3948)

Proof

a) GPSR suboptimal

We will show that GPSR is non-optimal by counter example. Assume that we have found a path using GPSR. Such a path must have a first node, that was closest to the sink. It is possible to construct a graph such that the first node leads the GPSR algorithm along a sub-optimal path because of the initial decision to follow the closest node.

b) Dijkstra better than GPSR

Since dijkstraPathHops() optimizes for path hops, then it must do better or equal to GPSR for path hops. Using Dijkstra's algorithm without regard for weight between nodes will provide a result similar to GPSR, but completely disregard all latency requirements. It is simple to construct a graph containing an obvious shortest path (in terms of hops), but terribly in-efficient with regard to latency. We can construct such a graph by initially setting all weights to 1, running Dijkstra's algorithm (min-hop), and then modifying the weights of all nodes in the path to have an extremely high latency. Such a graph will return the same path with a min-hop metric, but return a much more efficient path using a min-latency metric.

Efficiency

a) Memory

The data structure used for this lab consists of a map of vertices to a map of vertices to edges. In essence, each vertex holds a map with keys for each neighbor within the transmission range. The space complexity of this representation is O(V * E).

b) Runtime

Creating the data structure is rather fast as it only requires examining each edge in the graph. The runtime complexity of creation is O(E).

c) Dijkstra Runtime

The data structure used within Dijkstra's algorithm allows O(1) lookup of neighbors by allowing us to search for an edge between two vertices. Therefore, the overall runtime complexity of Dijkstra's algorithm is O(V^2). Worst-case, the algorithm will examine the metric between every pair of nodes because we do not use a priority queue in our algorithm.

Runtime Efficiency and Success Rate

Transmission Range = 5.0 meters.

The GPSR algorithm is successful 4950/4950 times.

The average time taken by the GPSR algorithm on successful runs is 565915 nanoseconds.

Dijkstra's algorithm (Min Latency) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Latency) on successful runs is 721859 nanoseconds.

Dijkstra's algorithm (Min Hops) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Hops) on successful runs is 812555 nanoseconds.

Transmission Range = 10.0 meters.

The GPSR algorithm is successful 4950/4950 times.

The average time taken by the GPSR algorithm on successful runs is 558833 nanoseconds.

Dijkstra's algorithm (Min Latency) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Latency) on successful runs is 749242 nanoseconds.

Dijkstra's algorithm (Min Hops) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Hops) on successful runs is 753913 nanoseconds.

Transmission Range = 15.0 meters.

The GPSR algorithm is successful 4950/4950 times.

The average time taken by the GPSR algorithm on successful runs is 608679 nanoseconds.

Dijkstra's algorithm (Min Latency) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Latency) on successful runs is 772838 nanoseconds.

Dijkstra's algorithm (Min Hops) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Hops) on successful runs is 741235 nanoseconds.

Transmission Range = 20.0 meters.

The GPSR algorithm is successful 4950/4950 times.

The average time taken by the GPSR algorithm on successful runs is 562438 nanoseconds.

Dijkstra's algorithm (Min Latency) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Latency) on successful runs is 763120 nanoseconds.

Dijkstra's algorithm (Min Hops) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Hops) on successful runs is 793823 nanoseconds.

Transmission Range = 25.0 meters.

The GPSR algorithm is successful 4950/4950 times.

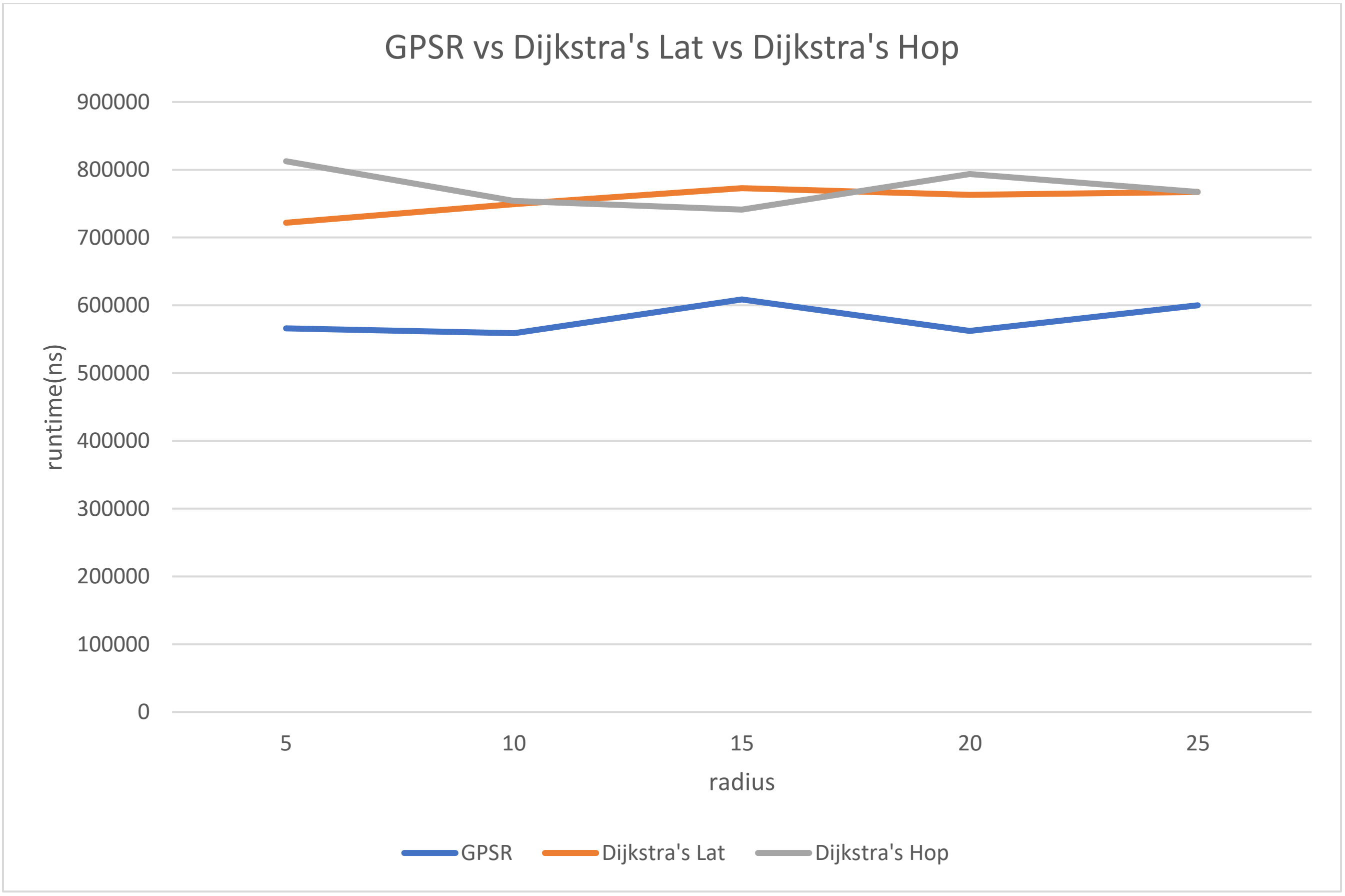
The average time taken by the GPSR algorithm on successful runs is 599874 nanoseconds.

Dijkstra's algorithm (Min Latency) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Latency) on successful runs is 767308 nanoseconds.

Dijkstra's algorithm (Min Hops) is successful 4950/4950 times.

The average time taken by Dijkstra's algorithm (Min Hops) on successful runs is 767341 nanoseconds.



The success rate of all algorithms was 100% for all tested transmission ranges in the large input testcase.

Greater vertex radius meant more searching, a bit slower for performance. Depending on requirements of application, GPSR may be suitable because it is quite fast relative to Dijkstra’s algorithm. The two versions of Dijkstra’s algorithm are similar, but it appears that the dijkstra algorithm optimizing for hops has a relatively small advantage time-wise against the latency-optimal solution.