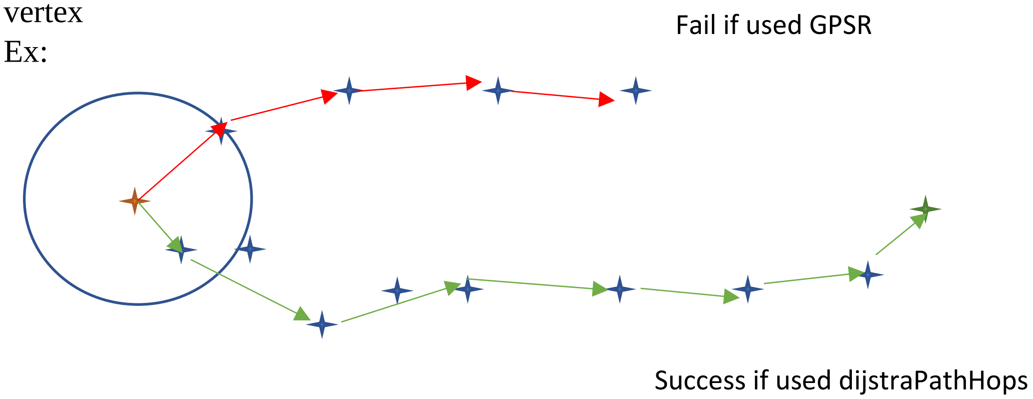
Lab 2 Report

Proofs:

A) The GPSR is not optimal if the transmission range is not sufficient enough to reach the sink through all the closest path. This will lead to a fail routing option if greedy algorithm is utilized. In the example below, this is a counter example to the GPSR since GPSR will not be able find a successful path to the sink vertex

Ex:



B) dijkstraPathHops will be more optimal than GPSR since it will give a path to the sink as long as the transmission range permit it to. However, it is not the most optimal since the it does not account for high latency so you could have a path that have the least amount of hop but have extremely high latency.

Efficiency Analysis:

A) The Big-O for the memory for the graph representation for is O(V+2E) or O(V +E) since I represented as an Adjacency List.

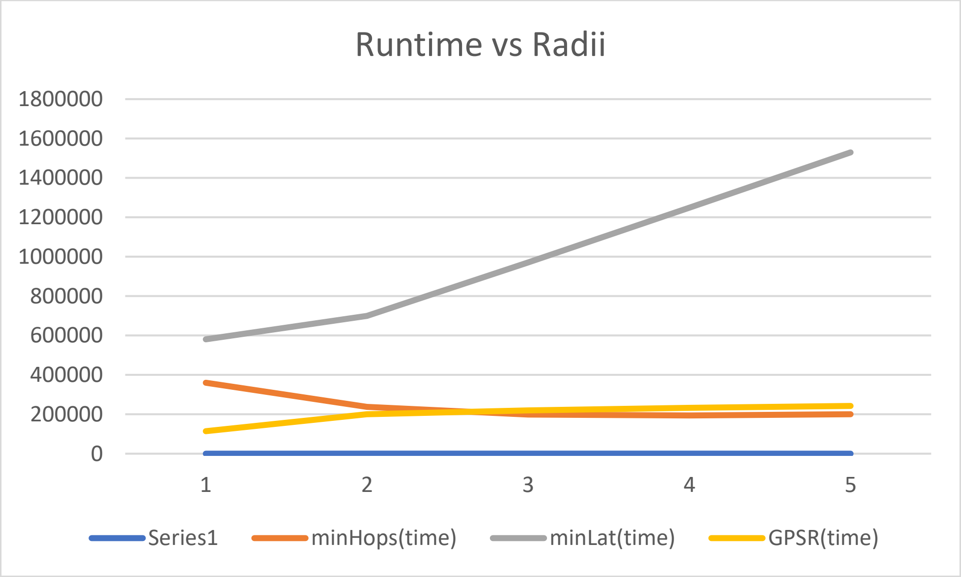
B) The runtime for the graph representation is $O(n^2)$. This is because I originally created the graph as an Adjacency Matrix since I had to compare every vertex to see if it with in the range of another vertex. Afterward, I reduced the Adjacency Matrix to an Adjacency List since it allowed me to optimize the runtime for GPSR and Dijkstra's algorithms. Since instead of comparing every node to see which node has the lowest latency or closest to the sink while making sure it fit into the graph, I only have to compared the edge that is in the transmission range. So even though my graph representation is long but it allowed me to reduce the runtime on other the GPSR and Dijkstra's algorithms.

C) The overall runtime for the Dijkstra's algorithm is O(n^2), however it might be reduce since I don't necessary compare every single edges but just the edges that is within the transmission range.

Runtime Efficiency and Success Rate:

A) As the vertex radii increase, Dijkstra for min latency increase quadratically while Dijkstra for min Hop and GPSR have a runtime decrease as vertex radii increase. I will used GPSR if I know the transmission range and that all the node is connected with that transmission range since GPSR run much faster than Dijkstra. I would use Dijkstra if I don't know if all the node is connected or not since Dijkstra has a higher success rate than GPSR.

Radii	minHops(time)	minLat(time)	GPSR(time)	minHops(success)	minLat(success)	GPSR(success)
5	360158	580669	114212	100	100	100
10	237724	699471	200009	100	100	100
15	198137	970807	219078	100	100	100
20	194006	1248328	233036	100	100	100
25	200210	1529701	242427	100	100	100


Runtime vs Radii
Series1
minHops(time)
minLat(time)
GPSR(time)