FIT1045 Assignment: Algorithmic Problem Solving

Greedy Discussion

Ray Krishardi Layadi - 26445549

For the greedy cheapest connection, my approach to the problem is to first update all the weights in intermediate graph (K) with smaller weight for each intersection between the intermediate graph (K) and the two graphs to be connected by the intermediate graph (G and I). Next, we need to find the starting and stopping /ending vertices to connect the two graphs that do not share any vertices (G and I). This can be achieved by using the intermediate graph (K). In order to determine the starting vertices, we can simply find the intersect vertices between the intermediate graph (K) and one of the graph that we intend to connect (i.e. G). The intersection of vertices between graph “K” and “G” will result in the starting vertices “w” and “Q”. Subsequently, we could also determine the stopping/ending vertices by finding the intersect vertices between the intermediate graph (K) and the other graph that we intend to connect (i.e. I). The intersection of vertices between graph “K” and “I” will result in the stopping/ending vertices “front” and “tank1”.

The next step would be to traverse through the intermediate graph (K) starting from each of the vertex in the starting vertices and determine the next adjacent vertex with the least amount of weight (in this case, we also need to make sure that the selected start vertex does not visit the other start vertices (i.e. “Q” must not visit “w” and vice versa)). After we have determined the next adjacent vertex with the least amount of weight then we visit that vertex and repeat the same process until the vertex that we are trying to visit is one of the vertex in the stopping/ending vertices.

I chose this approach because I realized that the starting and stopping/ending vertices to connect the two graphs (G and I) could be obtained by finding the intersect vertices between the intermediate graph (K) and the two graphs to be connected (G and I). After realizing that, all we need to do for the greedy approach is to traverse through the intermediate graph (K) starting from each vertex in the starting vertices (we need to iterate through each vertex in the starting vertices as they might visit different vertices in the intermediate graph) then we need to select the next adjacent vertex with the least amount of weight and stop when we are trying to visit one of the vertex in the stopping/ending vertices.

Finally, in the case of finding the cheapest connection for graphs G and I via K, my solution is able to get the optimal result as there is no multiple adjacent vertices that share the same smallest weight at a particular vertex (i.e. all adjacent vertices from “Q” have unique weights) and I do not allow the selected start vertex to visit the other start vertices (i.e. I do not allow “Q” to go back to “w” even though at starting vertex “Q”, the cost to visit “w” is 2 which is the least among other adjacent vertices). However, my application of greed would probably fail to produce the optimal solution if there is more than one (multiple) adjacent vertices which share the same smallest weight when making a greedy decision to traverse to a particular vertex at a given vertex because the selected vertex might not produce the optimal solution while the other not selected vertex might.