Design Strategy for Optimisation

Problem statement

For the dynamic situation in which each time a Producer process wakes, new nodes and edges are added to the existing graph, calculate the shortest path between all newly added K number of nodes by the producer and the already existing graph for each consumer process.

Approach without optimisation

Apply the Dijkstra algorithm simply for each of the nodes in the graph currently and get the shortest path between all nodes. Analysis of the following:
Let N and K be the original and newly added nodes respectively. Applying single source Dijkstra Algorithm to each of the (N+K) nodes would result in a time complexity of O(V*E*LogV) where V and E are the total number of vertices and edges in the newly created graph.

Approach with optimisation

We have already found the shortest path from all nodes in the original graph containing N nodes, in the previous iteration. Now, after the nodes updation, we will first apply the single source shortest path Dijkstra algorithm for each of the newly created K nodes. After that, we will one by one check if in the original set of N nodes there is a possibility of path updation. For that, we will iterate through each of the K nodes (p) and for each pair of nodes (i and j) among N, check if the path length between i and j is larger than the path between i and j via p:

To check if: length(i, p) + length(p, j) < length(i, j)

If this is true, we will update the shortest path between i and j via p.

Time Complexity: O(E*LogN) + O(M*e*log*K) + O(K*N^2)
Where E and e are the number of edges in the graph of N and K nodes respectively.

The proposed method provides a massive improvement in the time complexity as the maximum value possible of K for an iteration is only 30 (as provided in the assignment).