Description:

When new nodes are added by the producer process, the shortest paths between such new nodes and other nodes need to be computed, and the shortest paths between old nodes might need to be updated. We need to find an optimized way to do this. Below are a few notations and the corresponding values we will be using:

Let, V (No. of vertices) = 4000, E (No. of Edges) = $8 * 10^4$, M (No. of new nodes) = 30.

Ordinary Approach:

Apply Dijkstra's algorithm on all nodes after the new nodes have been added.

Time complexity = (V+M) * E * logV = $1.16 * 10^9$ operations

Our Proposed Approach:

1) We first apply Dijkstra's algorithm on the new nodes to compute the shortest paths between :

    - pairs of new nodes,

    - pairs of old and new nodes.

   which gives a total time complexity of = 10 * M * E log V = $8.64 * 10^7$ operations. We mutiply by 10 because each consumer will have to do dijkstra from each new node to get the path distances from new nodes to old nodes.

2) Now, for updating the shortest paths between pairs of old nodes (if needed), we apply the following optimization :

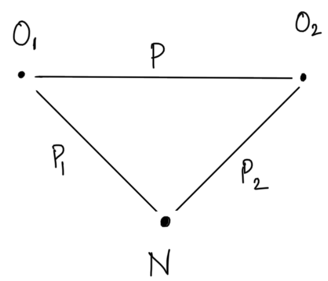
    - Let O_1 and O_2 be a pair of old nodes and P be the shortest path between them

      before adding the new nodes.

    - Let N be one of the new nodes being added in the producer process.

    - For every pair O_1, O_2 and N we repeat the following process based

      on the two possible cases:



P_1 : Shortest Path length b/w O_1 and N

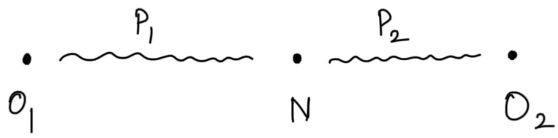
P_2 : Shortest Path length b/w O_2 and N

Case 1: $P <= P_1 + P_2$

Solution: No changes made

Case 2: $P > P_1 + P_2$

Solution: Update the shortest path between O_1 and O_2 as:



Correctness of the Approach:

The following statements about the optimization step can validate the approach:

- To perform the optimization step as described above, the shortest paths between new nodes and other nodes need to be precomputed, which is done in Step 1 of our approach.
- If the introduction of new nodes still doesn't create a shorter path between two old nodes, then the shortest paths between these nodes remain the same. Otherwise, the previously shortest path between two old nodes is replaced by a new shortest path, involving one or more new nodes. Hence, this approach is guaranteed to give the shortest possible route between all pairs of nodes.
- Though this approach seemingly involves only one new node during the shortest path finding, multiple new nodes can get involved in the shortest path implicitly because the shortest paths between (O_1 and N) and (O_2 and N) might involve new nodes.

Total Time Complexity:

For the above optimization step,

- Total no. of pairs involving old nodes = $V^2 / 2$
- Total no. of new nodes = M

Thus, the total time complexity of the above pairwise optimization step + applying Dijkstra on all new nodes gives :

$= M * V^2 / 2 + 10*M * E \log V$

$= 0.326 * 10^9$ operations

Improvement = 1.16 / 0.326 = 3.558