CS31202 - Operating Systems Laboratory

Indian Institute of Technology Kharagpur

Department of Computer Science and Engineering

OS Assignment-3 Report

Authors:
Burra Nithish (20CS10018)
Kartik Pontula (20CS10031)
Astitva (20CS30007)
Rahul Mandal (20CS30039)

Date: February 27, 2023

Optimization Strategy

Consider an unweighted undirected graph G. On implementing the Dijkstra algorithm considering a source node S, we get the shortest paths to all nodes. Running this repeatedly for every source node yields a shortest-path matrix.

One optimization strategy that has been implemented is that the shortest paths are not recomputed when there are no new nodes added. This significantly reduces the time taken for Dijkstra for newer iterations owing to a smaller number of nodes that need to be relabeled (because the nodes are already labeled).
For each new node U added to the graph, we first need to run Dijkstra with the new node as the source. Following this, using the old sources S and the new graph, we only recompute the labels that need to be changed, as follows:

- Check for the smallest label among all the neighbors V of the new node U and assign U's label as this value incremented by one.
- Node U's neighbors are scanned to check if their shortest paths get affected by U (i.e. the path from S to U to V is shorter than the old path from S to V). If not, proceed to iterate using another source S with U in focus.
- If yes, relaxation is performed for those neighbors, and we store them in a min-heap, popping the top node and repeating this process until the heap is empty (this resembles Dijkstra in a way but is not the same).
- This process is repeated for each new node added.

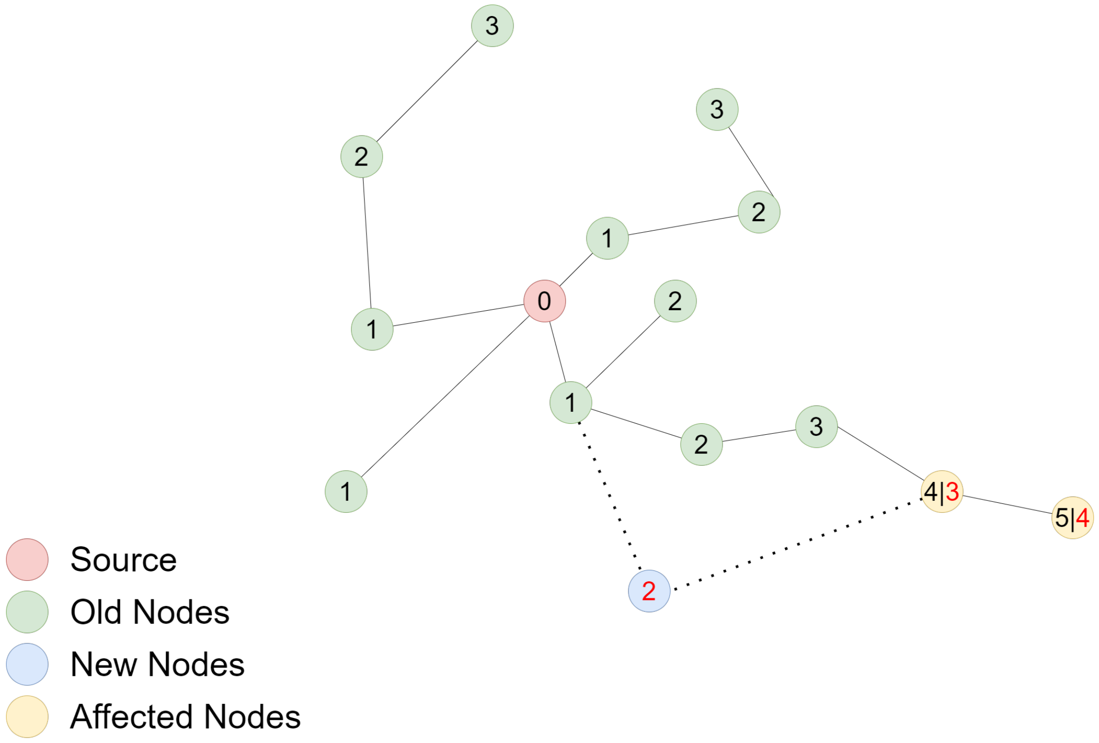

Figure 1: The Graph

In the above diagram, the minimum spanning tree of the graph is represented after running the initial Dijkstra. For the single new node, its label was computed first. The nodes which get affected by this node are shown in yellow, hence we begin reevaluating those shortest paths using a variation of Dijkstra (labels are still assigned relative to the red node as source).

This strategy uses the computed values of the first round of the Dijkstra algorithm and only recomputes the nodes connected to the new nodes. This reduces the re-computation time of the algorithm whenever a new node is encountered. This strategy is used for every source node that is considered in the graph.