CS/CE 412/471 Algorithms: Design & Analysis
Paper Selection Proposal: *Dynamizing Dijkstra*

Adina Adnan Mansoor, Syeda Rija Hasan Abidi, Syed Zuhair Abbas Rizvi

April 3, 2025

1 Paper Details

- **Title:** Dynamizing Dijkstra: A Solution to Dynamic Shortest Path Problem through Retroactive Priority Queue
- **Authors:** Sunita, Deepak Garg
- **Conference:** Journal of King Saud University – Computer and Information Sciences
- **Year:** 2021
- **DOI/Link:** Link to paper

2 Summary

This paper presents an enhancement to Dijkstra's algorithm by introducing a *retroactive priority queue* (RPQ), allowing insertions and deletions to be invoked or revoked at any point in time. The priority queue is implemented with the help of red-black trees to maintain elements and their lifetimes in the queue. This approach efficiently addresses the dynamic shortest path problem, which is crucial in real-world applications like transportation and network routing. The key contribution is the development of a flexible, dynamic data structure that supports updates without recomputing the entire shortest path tree.

3 Justification

This paper aligns well with our course material. We have been studying and implementing graph algorithms such as the Bellman-Ford Algorithm to solve the shortest path problem, and static Dijkstra's algorithm is also planned for Week 15 of the semester. This paper considers a version of the shortest path problem closer to reality, where the edges are mutable. Using this paper for our project will further enrich our concepts of shortest paths, optimization, graph operations, and their implementation in real-world circumstances. As stated before, dynamic shortest paths are highly relevant to real-world scenarios such as GPS navigation and network optimization. Furthermore, any combinatorial optimization problem can be reduced to a shortest path problem, making this paper both practically and theoretically relevant.

4 Implementation Feasibility

The paper contains pseudocode of the dynamic Dijkstra algorithm as well as all the helper functions needed for the RPQ. All project implementation will be done using Python. For experimental analysis, we will generate randomized graphs using Python libraries such as Networkx, matplotlib, etc. and also utilize real-world dynamic graph datasets from various domains and resources such as Pakistan's Open-Street Map and Bitcoin OTC.

5 Team Responsibilities

Given the sequential nature of implementation, analysis, and report-writing phases, all team members will be contributing equally to all phases.