# Title: A Comparative Analysis of Dijkstra's and Bellman-Ford Algorithms for Emergency Route Planning in Urban Areas

Abstract:

## This paper presents a comparative analysis of Dijkstra's and Bellman-Ford algorithms for emergency route planning in urban areas. The study focuses on the efficiency and effectiveness of these algorithms in finding the shortest paths between different locations within a city's road network. We implemented both algorithms in Python and evaluated their performance using a simulated road network comprising various cities and their connections. Our findings provide insights into the trade-offs between computational complexity and accuracy in emergency route planning scenarios.

## 1. Introduction:

## Emergency route planning plays a crucial role in urban areas to ensure timely response and efficient allocation of resources during emergencies such as medical emergencies, accidents, or natural disasters. The shortest path between an incident location and a destination can significantly impact the response time of emergency services. Dijkstra's and Bellman-Ford algorithms are widely used for finding the shortest paths in road networks. This paper aims to compare these algorithms in the context of emergency route planning within a city.

## 2. Background:

**Dijkstra's Algorithm:** Dijkstra's algorithm is a greedy algorithm that finds the shortest path from a single source vertex to all other vertices in a weighted graph with non-negative edge weights.

**Bellman-Ford's Algorithm:** Bellman-Ford's algorithm handles graphs with negative edge weights and can detect negative weight cycles. It iteratively relaxes all edges multiple times to compute the shortest paths.

## 3. Methodology:

We implemented Dijkstra's and Bellman-Ford algorithms in Python and utilized the NetworkX library to model the road network. The road network consisted of various cities interconnected by roads with associated weights representing the distances between cities. We simulated emergency calls between random pairs of cities and measured the time taken by each algorithm to compute the shortest path for each call.

## 4. Benchmarking:

## To evaluate the performance of Dijkstra's and Bellman-Ford algorithms, we conducted benchmarking experiments using a simulated road network with varying sizes and edge weights. We measured the execution time of both algorithms for different scenarios, including graphs with positive weights. The benchmarking results provide insights into the scalability and efficiency of the algorithms under different conditions.

Created a road network with +ive weights

A diagram of a network

Description automatically generated

Simulated 100 emergency calls

A close-up of a computer code

Description automatically generated

A close-up of a computer screen

Description automatically generated

## 5. Analysis

Comparison of Time Complexities: Dijkstra's algorithm has a time complexity of O((V + E)logV) using a priority queue, while Bellman-Ford's algorithm has a time complexity of O(VE).

The choice between Dijkstra's and Bellman-Ford algorithms depends on the specific characteristics of the road network and the requirements of the emergency route planning system. Dijkstra's algorithm offers faster computation for most cases but may not be suitable for networks with negative weight edges. Bellman-Ford algorithm, on the other hand, provides a more robust solution at the cost of higher computational complexity.

The program finds out the shortest route and spits out the time taken to find that out

A screenshot of a computer

Description automatically generated

## 6. Results:

## Our experimental results indicate that Dijkstra's algorithm generally outperforms the Bellman-Ford algorithm in terms of computational efficiency for urban road networks without negative weight edges. However, as the complexity of the road network increases or in the presence of negative weight edges, the Bellman-Ford algorithm demonstrates more robust performance due to its ability to handle such scenarios.

A screenshot of a computer

Description automatically generated

## 7. Conclusion:

## In conclusion, both Dijkstra's and Bellman-Ford algorithms offer viable solutions for emergency route planning in urban areas. Dijkstra's algorithm provides fast and efficient route planning in most scenarios, while Bellman-Ford algorithm offers robustness in handling complex road networks with negative weight edges. The choice between the two algorithms should be made based on the specific requirements and constraints of the application.

## 8. References:

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische mathematik, 1(1), 269-271.

Bellman, R. (1958). On a routing problem. Quarterly of Applied Mathematics, 16(1), 87-90.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT press.

Git hub - [SchoolPython/CSC506\_DAA/Module8\_PT/Module8\_PT.ipynb at main · ArunSaxena200/SchoolPython (github.com)](https://github.com/ArunSaxena200/SchoolPython/blob/main/CSC506_DAA/Module8_PT/Module8_PT.ipynb)