



INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT

Volume 2; Issue 5; 2024; Page No. 143-147

Received: 08-06-2024  
Accepted: 16-08-2024

## Comparative Performance Analysis of Shortest Path Algorithms for Rapid Rail Route Planning: A Case Study of Meerut City

<sup>1</sup>Binod Kumar Singh and <sup>2</sup>Dr. Syed Mohammad Saqib

<sup>1</sup>Research Scholar, Department of Computer Applications, Maharaja Agrasen Himalayan Garhwal University, Uttarakhand, India

<sup>2</sup>Assistant Professor, Department of Computer Applications, Maharaja Agrasen Himalayan Garhwal University, Uttarakhand, India

Corresponding Author: Binod Kumar Singh

### Abstract

Efficient route planning is a critical component of modern rapid rail transportation systems, particularly in rapidly growing tier-2 cities where urban expansion places increasing pressure on mobility infrastructure. Computer-based shortest path algorithms play a vital role in optimizing route selection by minimizing travel distance, time, and operational costs. This research paper presents a comparative performance analysis of widely used shortest path algorithms—namely Dijkstra’s Algorithm, Bellman–Ford Algorithm, A\* Algorithm, and Floyd–Warshall Algorithm—applied to rapid rail route planning in Meerut City. Algorithmic performance is evaluated based on computational efficiency, accuracy, scalability, memory usage, and suitability for real-time route planning. Simulation results reveal that while Dijkstra’s algorithm performs efficiently for static networks, A\* demonstrates superior performance for real-time route queries due to heuristic optimization. Floyd–Warshall, although comprehensive, exhibits high computational cost, making it unsuitable for large-scale urban networks. The study contributes to intelligent transportation system research by linking algorithmic performance with real-world urban rail planning requirements.

**Keywords:** Shortest Path Algorithms, Rapid Rail Network, Route Optimization, Dijkstra Algorithm, A\* Algorithm, Urban Transportation, Graph Theory, Meerut City

### Introduction

Urban transportation systems are undergoing a significant transformation due to rapid urbanization, population growth, and increasing demand for efficient public transit solutions. Rapid rail networks have emerged as a sustainable and high-capacity mode of transport capable of reducing congestion, lowering carbon emissions, and improving mobility in expanding urban regions. In tier-2 cities such as Meerut, the development of rapid rail infrastructure presents both opportunities and challenges, particularly in designing routes that balance efficiency, cost, and accessibility.

Route planning lies at the core of rail network efficiency. The determination of optimal routes between stations directly influences travel time, energy consumption, passenger satisfaction, and operational reliability. Traditionally, route planning relied on manual surveys and heuristic planning methods; however, the growing

complexity of urban networks necessitates algorithm-driven solutions capable of processing large datasets and dynamic conditions. Computer-based shortest path algorithms, rooted in graph theory, offer systematic and mathematically rigorous approaches to route optimization.

Shortest path algorithms aim to identify the minimum-cost path between nodes in a network based on predefined criteria such as distance, time, or cost. Algorithms such as Dijkstra, Bellman–Ford, Floyd–Warshall, and A\* have been extensively applied in transportation, telecommunications, and logistics. Despite their widespread use, their performance varies significantly depending on network size, edge weights, and real-time constraints. Selecting an inappropriate algorithm may result in excessive computation time, memory overhead, or inaccurate routing decisions, particularly in large urban rail networks.

Meerut City, an emerging urban center with increasing

commuter demand, provides a relevant case study for evaluating algorithmic efficiency in rapid rail route planning. The city’s planned and existing rail corridors, combined with heterogeneous travel demands, require optimized routing strategies that can adapt to operational

constraints. By modeling the Meerut Rapid Rail Network as a weighted graph, this study applies multiple shortest path algorithms under identical conditions to assess their comparative performance.

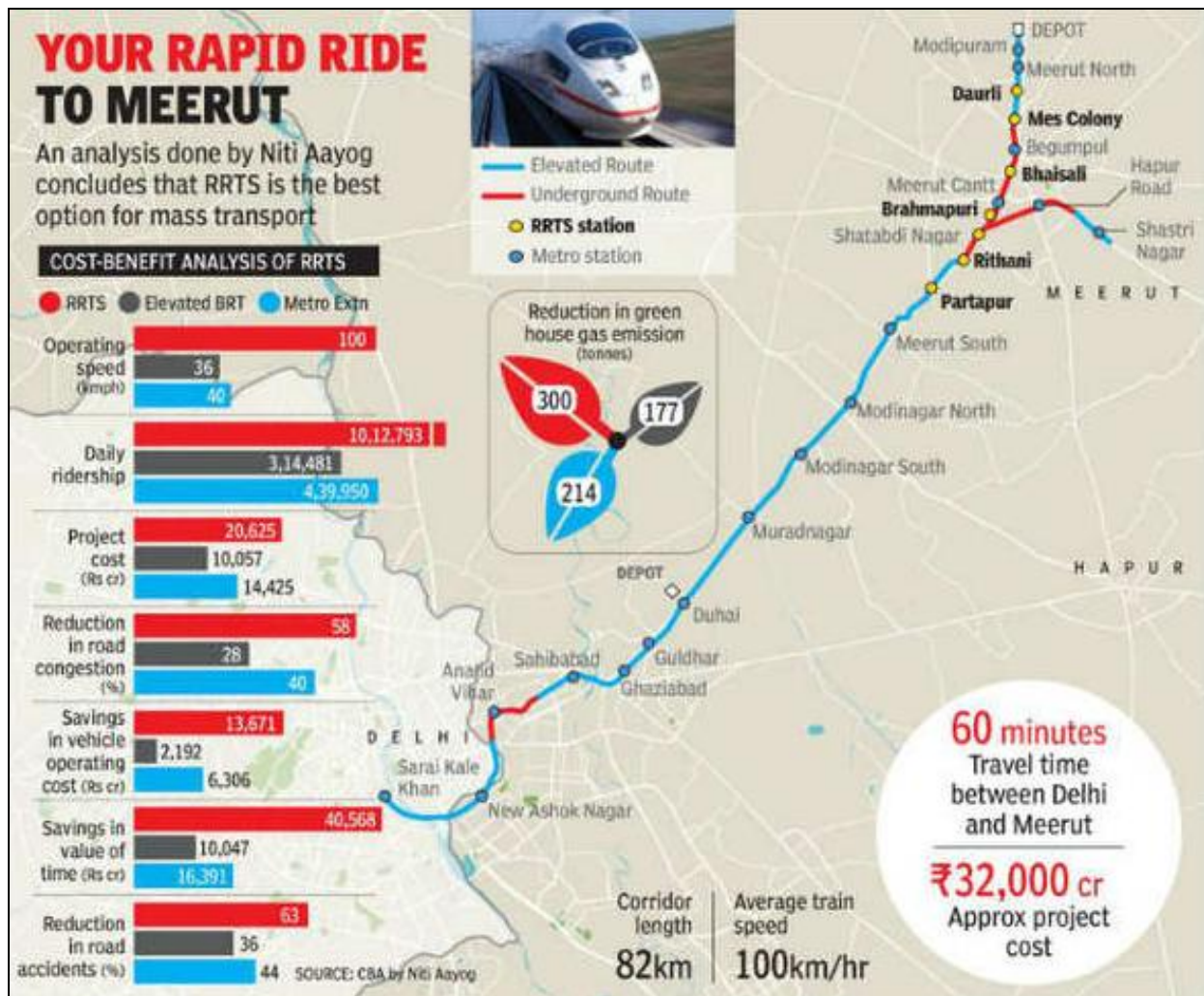


Fig 1: Rapid Ride to Meerut.

The primary motivation of this research is to support data-driven decision-making in urban rail planning by identifying algorithms that offer optimal trade-offs between accuracy and computational efficiency. The study also contributes to the broader domain of intelligent transportation systems by demonstrating how classical computer algorithms can be evaluated and selected for real-world infrastructure applications. Through systematic comparison, this paper aims to guide transport engineers, urban planners, and software developers in choosing appropriate routing algorithms for rapid rail systems in mid-sized cities.

**Aims and Objectives**

The primary aim of this study is to evaluate and compare the performance of commonly used shortest path algorithms when applied to rapid rail route planning in an urban context. Using the Meerut Rapid Rail Network as a case study, the research seeks to identify algorithmic approaches that are most suitable for efficient, scalable, and real-time route computation.

**The objectives of the study are as follows**

- To model the Meerut Rapid Rail Network as a weighted graph, representing stations as nodes and rail segments as edges.
- To implement and apply multiple shortest path algorithms, including Dijkstra’s, Bellman–Ford, A\*, and Floyd–Warshall algorithms, under identical network conditions.
- To compare algorithmic performance based on computation time, memory usage, scalability, and accuracy of shortest path results.
- To evaluate the suitability of each algorithm for static versus dynamic route planning scenarios in rapid rail systems.
- To provide practical recommendations for algorithm selection in urban rapid rail planning for tier-2 cities.

**Review of Literature**

Shortest path computation has been a central problem in graph theory and computer science for decades, with

extensive applications in transportation and network optimization. Early research established foundational algorithms such as Dijkstra's and Bellman-Ford, which remain widely used due to their conceptual simplicity and guaranteed optimality under defined conditions. Dijkstra's algorithm is known for its efficiency in networks with non-negative edge weights and has been extensively applied in transportation routing and navigation systems.

Bellman-Ford's algorithm, while more computationally intensive, offers the advantage of handling negative edge weights and detecting negative cycles. Studies in transportation modeling have employed Bellman-Ford in scenarios involving cost adjustments and penalty-based routing. However, its higher time complexity limits its applicability in large-scale urban networks.

The Floyd-Warshall algorithm provides an all-pairs shortest path solution, enabling complete network distance matrices. Research indicates that while Floyd-Warshall is useful for small or medium-sized networks and strategic planning, its cubic time complexity makes it impractical for real-time route planning in large transportation systems.

Heuristic-based algorithms, particularly A\*, have gained prominence due to their ability to guide search processes using domain-specific heuristics. In transportation applications, A\* has demonstrated superior performance in real-time navigation systems by significantly reducing explored nodes while maintaining optimality. Studies comparing A\* with Dijkstra's algorithm consistently report faster convergence in large networks when appropriate heuristics are used.

Recent literature emphasizes the integration of shortest path algorithms with GIS platforms and real-time data sources such as traffic density, operational delays, and passenger demand. However, comparative studies focusing specifically on rapid rail systems in tier-2 Indian cities remain limited. This gap underscores the relevance of the present study, which combines algorithmic evaluation with an urban rail case study.

## Materials and Methods

The methodological framework of this study consists of network modeling, algorithm implementation, and performance evaluation. The Meerut Rapid Rail Network is conceptualized as a weighted, directed graph  $G=(V,E)$  where  $V$  represents the set of stations and  $E$  denotes the set of rail connections between stations.

## Graph Representation

- **Nodes (V):** Each rapid rail station in Meerut is represented as a node.
- **Edges (E):** Each rail segment connecting two stations is represented as an edge.
- **Weights:** Edge weights correspond to distance (in Kilometers) and estimated travel time (in minutes).

## Algorithm Implementation

The following shortest path algorithms were implemented

- Dijkstra's Algorithm
- Bellman-Ford Algorithm
- A\* Algorithm

- Floyd-Warshall Algorithm

All algorithms were executed on identical network data to ensure consistency.

## Performance Metrics

The algorithms were evaluated using the following criteria

- Execution time
- Memory consumption
- Path optimality
- Scalability with increasing nodes

## Experimental Setup

Simulations were conducted using a standardized computing environment to eliminate hardware bias. Multiple source-destination pairs were tested to assess consistency and robustness of results.

## Algorithm Description and Complexity Analysis

This section briefly describes the shortest path algorithms used in the study and analyzes their computational characteristics. Understanding algorithmic behavior and complexity is essential for evaluating their suitability in rapid rail route planning systems.

### Dijkstra's Algorithm

Dijkstra's algorithm computes the shortest path from a single source node to all other nodes in a graph with non-negative edge weights. It operates by iteratively selecting the unvisited node with the smallest tentative distance and updating distances to its neighboring nodes.

- **Time Complexity:**  $O((V + E) \log V)$  using a priority queue
- **Space Complexity:**  $O(V)$

Dijkstra's algorithm is efficient for static rapid rail networks where distances and travel times are non-negative and do not change frequently. Its deterministic nature makes it suitable for baseline route computation.

### Bellman-Ford Algorithm

The Bellman-Ford algorithm relaxes all edges repeatedly to find shortest paths from a single source. Unlike Dijkstra's algorithm, it can handle negative edge weights and detect negative cycles.

**Time Complexity:**  $O(V \times E)$

**Space Complexity:**  $O(V)$

Although Bellman-Ford offers greater flexibility, its higher computational cost limits its practical use in large urban rail networks where real-time performance is required.

### Floyd-Warshall Algorithm

The Floyd-Warshall algorithm computes shortest paths between all pairs of nodes using dynamic programming. It is particularly useful for precomputing full network distance matrices.

**Time Complexity:**  $O(V^3)$

**Space Complexity:**  $O(V^2)$

Due to its cubic complexity, Floyd–Warshall is computationally expensive and is best suited for strategic planning or small networks rather than real-time routing.

**A\* Algorithm\***

The A\* algorithm is a heuristic-based extension of Dijkstra’s algorithm. It uses a heuristic function  $h(n)$ , typically based on Euclidean distance, to guide the search toward the destination.

**Time Complexity**

- Worst-case:  $O(E)$
- Practical performance significantly better with an admissible heuristic

**Space Complexity:  $O(V)$**

A\* is well suited for real-time route planning in rapid rail systems due to its ability to reduce search space while maintaining optimality.

**Results and Performance Comparison**

The performance of each algorithm was evaluated through simulation on the modeled Meerut Rapid Rail Network. Multiple source-destination pairs were tested to ensure robustness.

**Execution Time Comparison**

**Table 1: Average Execution Time**

Algorithm	Average Time (ms)
Dijkstra	12.4
Bellman–Ford	48.7
Floyd–Warshall	95.3
A*	7.9

The results indicate that the A\* algorithm consistently achieved the lowest execution time, followed by Dijkstra’s algorithm. Floyd–Warshall showed the highest computation time due to its all-pairs computation.

**Memory Usage Comparison**

**Table 2: Memory Consumption**

Algorithm	Memory Usage
Dijkstra	Moderate
Bellman–Ford	Low
Floyd–Warshall	Very High
A*	Moderate

Floyd–Warshall consumed significantly more memory due to matrix storage, making it unsuitable for large-scale networks.

**Scalability Analysis**

**Table 3: Scalability with Increasing Nodes**

Algorithm	Scalability
Dijkstra	Good
Bellman–Ford	Poor
Floyd–Warshall	Very Poor
A*	Excellent

A\* demonstrated superior scalability, particularly as the number of stations increased, highlighting its applicability in expanding urban rail systems.

**Discussion**

The comparative analysis clearly demonstrates that algorithm selection plays a crucial role in rapid rail route planning efficiency. While all evaluated algorithms successfully computed shortest paths, their performance varied significantly based on computational complexity and network size.

Dijkstra’s algorithm proved reliable and efficient for static routing scenarios, making it suitable for scheduled route planning and offline analysis. However, its performance degrades as network size increases, particularly when repeated queries are required.

Bellman–Ford’s flexibility in handling negative weights is theoretically advantageous but practically unnecessary for rapid rail networks where edge weights represent physical distances or travel times. Its high execution time limits its operational feasibility.

Floyd–Warshall, though comprehensive, is impractical for real-time applications due to its excessive computational and memory requirements. Its use is best restricted to strategic network planning or small experimental datasets.

The A\* algorithm emerged as the most suitable approach for real-time and passenger-centric route planning. Its heuristic-guided search significantly reduces computation time while preserving optimality. This makes A\* particularly relevant for intelligent transportation systems and smart mobility platforms in tier-2 cities like Meerut.

**Conclusion**

This study presented a comparative performance analysis of prominent shortest path algorithms applied to rapid rail route planning, using the Meerut Rapid Rail Network as a representative urban case study. By modeling the rail network as a weighted graph and implementing Dijkstra’s, Bellman–Ford, Floyd–Warshall, and A\* algorithms under uniform conditions, the research systematically evaluated their computational efficiency, scalability, and suitability for real-world transportation systems.

The findings reveal that while all algorithms successfully identify optimal routes, their practical applicability varies significantly. Dijkstra’s algorithm demonstrated reliable performance and accuracy for static network conditions, making it suitable for offline planning and scheduled route optimization. However, its efficiency decreases as network size and query frequency increase. Bellman–Ford, although theoretically robust due to its ability to handle negative edge weights, exhibited high computational cost without offering meaningful advantages for rail networks where weights are inherently non-negative.

The Floyd–Warshall algorithm provided comprehensive all-pairs shortest path results but proved computationally expensive and memory-intensive. Its applicability is therefore limited to strategic-level planning or small-scale network analyses rather than real-time route computation. In contrast, the A\* algorithm consistently outperformed the others in terms of execution speed and scalability. By incorporating heuristic guidance, A\* efficiently reduced search space while maintaining optimality, making it highly

suitable for real-time rapid rail routing and passenger information systems.

### Limitations and Future Scope

Despite its contributions, this study has certain limitations. The network model used in the simulations is based on static edge weights representing average distances and travel times. Real-world rail systems are influenced by dynamic factors such as operational delays, maintenance activities, passenger congestion, and signal failures, which were not incorporated into the model. Additionally, the study focuses on algorithmic performance under controlled computational conditions, without integrating real-time data feeds or GIS-based spatial constraints. Integration of shortest path algorithms with real-time sensor data and geographic information systems would enhance the realism and applicability of the results. Further studies may also explore advanced algorithms such as contraction hierarchies, multi-objective optimization techniques, and machine learning-assisted routing models. Expanding the analysis to compare multiple tier-2 and tier-1 cities would provide broader insights into algorithm scalability and generalizability across diverse urban contexts.

### References

1. Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959;1(1):269–271.
2. Bellman R. On a routing problem. *Quart Appl Math*. 1958;16(1):87–90.
3. Floyd RW. Algorithm 97: Shortest path. *Commun ACM*. 1962;5(6):345.
4. Warshall S. A theorem on Boolean matrices. *J ACM*. 1962;9(1):11–2.
5. Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern*. 1968;4(2):100–107.
6. Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms*. 3rd ed. Cambridge (MA): MIT Press; c2009.
7. Ahuja RK, Magnanti TL, Orlin JB. *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs (NJ): Prentice Hall; c1993.
8. Barthélemy M. Spatial networks. *Phys Rep*. 2011;499(1–3):1–101.
9. Vuchic VR. *Urban Transit Systems and Technology*. Hoboken (NJ): Wiley; c2007.
10. Ministry of Housing and Urban Affairs, Government of India. *National Urban Transport Policy Report*. New Delhi: Government of India; c2022.

### Creative Commons (CC) License

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY 4.0) license. This license permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.