A Comparative Study of Graph Search Algorithms for Route Optimisation in AI Systems

Chaithanya Mandadi
PG Scholar
Computer Science and Engineering
Aurora University
Hyderabad, India
mchaithanya@aurora.ac.in

Sowmya Mandadi
Research Scholar
Computer Science and Engineering
Aurora University
Hyderabad, India
aud22egcse03@aurora.edu.in

Chandrashekar K
Head of the Department
Computer Science and Engineering
Aurora University
Hyderabad, India
hodcse@aurora.edu.in

***Abstract*—Efficient route optimisation is a critical aspect of artificial intelligence (AI)-driven logistics and transportation systems. Various graph search algorithms, such as Dijkstra's Algorithm, Bellman-Ford Algorithm, A* Search, Breadth-First Search (BFS), and Depth-First Search (DFS), are commonly used for finding optimal paths in complex networks. This study provides a comparative analysis of these algorithms, evaluating their efficiency, scalability, and practical application in route optimisation scenarios. By reviewing existing literature and conducting empirical experiments, this research highlights the strengths and limitations of each algorithm and proposes recommendations for selecting the most suitable approach for AI-based route planning.**

***Index Terms*—Route optimisation, Graph search algorithms, Artificial Intelligence, Dijkstra's Algorithm, A* Search, Bellman-Ford Algorithm, BFS, DFS**

I. Introduction

In the modern era of artificial intelligence (AI) and smart transportation systems, optimising routes for logistics, navigation, and transportation has become a crucial challenge. Efficient route optimisation is essential for reducing travel time, minimising fuel costs, improving delivery efficiency, and enhancing overall operational effectiveness. AI-driven graph search algorithms have emerged as powerful tools for solving routing problems by finding the most efficient paths in complex networks. Various search algorithms, including Dijkstra's Algorithm, A* Algorithm, Floyd-Warshall Algorithm, Breadth-First Search (BFS), and Depth-First Search (DFS), have been widely employed to address these challenges.

A. Importance of Route Optimisation in AI Systems

Route optimisation is a fundamental problem in AI-driven applications such as:

- Logistics and Supply Chain Management – Companies like Amazon, FedEx, and UPS rely on optimised delivery routes to ensure timely and cost-efficient product distribution.
- Autonomous Vehicles and Smart Navigation – AI-driven route planning helps autonomous cars, drones, and robotic delivery systems efficiently navigate dynamic environments.
- Traffic Management Systems – Intelligent traffic routing helps reduce congestion in metropolitan areas by predicting and adjusting vehicle flow patterns.
- Emergency Response Services – Firefighters, ambulances, and disaster relief teams depend on fast and efficient route selection to minimise response times in critical situations.

B. The Role of Graph Search Algorithms in Route Optimisation

Graph search algorithms provide systematic approaches for identifying the shortest and most efficient routes in transportation networks. These algorithms can be broadly classified into:

- Uninformed Search Algorithms – BFS and DFS explore the entire network without using additional heuristics.
- Single-Source Shortest Path Algorithms – Dijkstra's algorithm and Bellman-Ford algorithm efficiently compute the shortest path from a given starting node to all other nodes.
- All-Pairs Shortest Path Algorithms – Floyd-Warshall algorithm determines the shortest paths between all pairs of nodes in a network.
- Heuristic-Based Algorithms – A* algorithm integrates heuristics to improve search efficiency, making it suitable for real-time applications.

C. Motivation for the Study

Many studies have evaluated individual graph search algorithms, but limited research has provided a comprehensive comparative analysis in the context of AI-driven route optimisation. This study aims to address the following key questions:

- Which algorithm provides the most efficient performance in different real-world delivery and navigation scenarios?
- What are the trade-offs between computational complexity, memory usage, and accuracy for different algorithms?
- How do heuristic-based methods compare to traditional graph traversal techniques in AI-based route planning?

D. Objectives of the Study

The primary objectives of this research are:

- To analyse and compare graph search algorithms in terms of efficiency, accuracy, and computational performance.
- To evaluate the effectiveness of Dijkstra's, Floyd-Warshall, A*, BFS, and DFS algorithms in AI-driven route optimisation.
- To assess real-world applications of these algorithms in logistics, autonomous navigation, and transportation systems.
- To provide empirical findings and recommendations for selecting the optimal algorithm based on specific use cases.

E. Research Significance

The findings from this study will contribute to the field of AI-driven transportation optimisation by offering a detailed comparative analysis of widely used graph search algorithms. Researchers, developers, and industry professionals can leverage these insights to improve route optimisation strategies for logistics companies, smart traffic systems, and AI-powered navigation solutions. Additionally, this research will highlight the potential of machine learning-enhanced heuristics in improving traditional graph search methods.

F. Structure of the Paper

This paper is structured as follows:

- Section 2 (History and Background) explores the evolution of graph search algorithms and their applications in AI systems.
- Section 3 (Literature Review) presents an in-depth analysis of previous studies, highlighting key findings in the field.
- Section 4 (Methodology) outlines the approach used for evaluating the performance of different algorithms.
- Section 5 (Findings and Results) provides a comparative analysis of algorithm performance based on case studies and simulations.
- Section 6 (Conclusion and Future Work) summarises key insights and suggests future directions for research.

II. History and Background

A. Evolution of Route Optimisation in Artificial Intelligence

Route optimisation has been a fundamental problem in various fields, including operations research, computer science, and artificial intelligence (AI). The need to determine the most efficient paths in transportation networks, logistics, and navigation systems has driven the development of graph search algorithms. Over the years, researchers have introduced numerous techniques to address this problem, evolving from classical mathematical approaches to AI-driven heuristics.

1) Early Developments in Graph Theory and Pathfinding: The foundation of graph theory was laid by Leonhard Euler in 1736 with his famous Königsberg Bridge Problem, which led to the concept of Eulerian paths. Later, in the 1950s and 1960s, researchers formulated key algorithms for shortest path problems, including:

- Dijkstra's Algorithm (1956) – Developed by Edsger W. Dijkstra, this algorithm efficiently finds the shortest path from a single source to all other nodes in a weighted graph.
- Bellman-Ford Algorithm (1958) – Introduced by Richard Bellman and Lester Ford, this algorithm accommodates graphs with negative weight edges but at a higher computational cost.
- Floyd-Warshall Algorithm (1962) – Proposed by Robert Floyd and Stephen Warshall, this dynamic programming approach computes shortest paths between all pairs of nodes in a graph.

These early algorithms formed the foundation for modern AI-driven pathfinding techniques, influencing transportation planning, network routing, and navigation systems.

2) The Rise of AI and Heuristic-Based Pathfinding: As AI research advanced in the 1970s and 1980s, researchers began incorporating heuristic functions into search algorithms to improve efficiency. Key milestones include:

- A* Algorithm (1968) – Developed by Peter Hart, Nils Nilsson, and Bertram Raphael, the A* algorithm combines Dijkstra's shortest path search with heuristic-based guidance, making it significantly faster for real-world applications.
- Greedy Best-First Search (1985) – This algorithm prioritises paths that seem promising based on a heuristic function but does not guarantee the optimal solution.
- Bidirectional Search (1990s) – A technique that simultaneously searches from both the start and goal nodes, reducing the search space and improving efficiency.

During this period, AI-driven route optimisation was increasingly applied in robotics, logistics, and network routing, setting the stage for further innovations.

B. Modern Advancements in Graph Search for Route Optimisation

With the rise of big data, cloud computing, and machine learning, modern AI systems have integrated advanced search algorithms to handle complex and large-scale route optimisation challenges.

1) Graph Neural Networks (GNNs) and AI-Driven Optimisation:

- Graph Neural Networks (GNNs) have revolutionised pathfinding by leveraging deep learning to dynamically improve route predictions and adapt to changing environments.
- Reinforcement Learning (RL) in Route Optimisation – RL-based models use past route data to learn and improve pathfinding efficiency over time, particularly in autonomous driving and delivery services.

2) Real-World Applications of Graph Search Algorithms: Today, graph search algorithms are widely used in various AI-powered applications:

- Logistics and Supply Chain – Companies like Amazon, FedEx, and UPS rely on AI-driven routing systems for efficient deliveries.

- Autonomous Vehicles – Self-driving cars use A* and Dijkstra's algorithm to compute safe and efficient navigation paths.
- Internet and Network Routing – Shortest path algorithms are implemented in network protocols like OSPF (Open Shortest Path First) and BGP (Border Gateway Protocol).
- Urban Traffic Management – AI-driven traffic control systems optimise vehicle movement using real-time graph search algorithms.

C. Challenges in Route Optimisation and Need for Comparative Studies

Despite the advancements, several challenges persist in AI-based route optimisation:

- Computational Complexity – Traditional algorithms like Floyd-Warshall become impractical for large graphs due to their high time complexity.
- Dynamic and Uncertain Environments – Real-world route planning requires algorithms that can adapt to changing traffic conditions, roadblocks, and weather disruptions.
- Scalability – As urban populations grow and transportation networks expand, AI systems must efficiently handle larger datasets.

Given these challenges, comparative studies are crucial to identifying the most efficient algorithms for different real-world scenarios. This study aims to bridge this gap by evaluating graph search algorithms based on efficiency, accuracy, and computational cost.

D. Summary of Historical Evolution

The Table I summarises the key milestones in the evolution of graph search algorithms for route optimisation:

TABLE I
HISTORICAL EVOLUTION OF GRAPH ALGORITHMS

Year	Algorithm	Developer(s)	Key Feature
1736	Eulerian Paths	Leonhard Euler	First concept in graph theory
1956	Dijkstra's Algorithm	Edsger W. Dijkstra	Single-source shortest path
1958	Bellman-Ford Algorithm	Richard Bellman, Lester Ford	Handles negative weights
1962	Floyd-Warshall Algorithm	Robert Floyd, Stephen Warshall	All-pairs shortest path
1968	A* Algorithm	Peter Hart, Nils Nilsson, Bertram Raphael	Heuristic-based shortest path
1985	Greedy Best-First Search	Various Researchers	Fast but may be suboptimal
1990s	Bidirectional Search	AI Researchers	Simultaneous search from start and goal
2010s	Graph Neural Networks (GNNs)	Deep Learning Community	AI-driven route optimisation
2020s	Reinforcement Learning for Pathfinding	AI Researchers	Learning-based adaptive routing

III. LITERATURE REVIEW

Graph search algorithms play a crucial role in route optimisation for artificial intelligence (AI) systems. Various studies have explored their efficiency, applicability, and comparative performance in different domains. This section reviews significant contributions in the field, categorising them based on their approach and application.

Balogun et al. (2024) analysed AI-based search algorithms for solving the 8-puzzle problem, focusing on heuristic methods. The study highlights how different search techniques, including Breadth-First Search (BFS), Depth-First Search (DFS), and A* search, impact solution efficiency. The authors conclude that heuristic-based searches outperform uninformed searches in complex problem-solving scenarios. Their findings are relevant to route optimisation, as delivery networks also require heuristic-driven decision-making for optimal pathfinding.

Bhat et al. (2024) compared Bellman-Ford and Dijkstra's algorithms for evacuation route planning in multi-floor buildings. Their study provides a practical perspective on real-time route optimisation, considering obstacles, multiple entry/exit points, and time constraints. They found that Dijkstra's algorithm provides superior performance for static environments, whereas Bellman-Ford is useful when negative weights are involved. These findings can be applied to logistics, where delivery routes must be optimised for multiple stops with variable constraints.

Elkari et al. (2024) investigated the application of BFS, DFS, and A* algorithms for maze navigation, offering insights into their computational efficiency and accuracy. Their study found that BFS guarantees an optimal solution but is computationally expensive for large graphs, while DFS is efficient in exploring deep search paths but does not always find the shortest route. The A* algorithm demonstrated superior efficiency by incorporating heuristics. These findings suggest that A* is a strong candidate for delivery route optimisation, particularly when computational efficiency is a concern.

Gupta and Aggarwal (n.d.) reviewed various algorithms and techniques for effective route optimisation. Their study emphasises the trade-offs between computational complexity, accuracy, and scalability. They highlight that heuristic-based approaches, such as A* and Genetic Algorithms, are more effective for real-world applications where dynamic changes in traffic and road conditions must be considered.

Jiang et al. (2024) explored the use of Graph Neural Networks (GNNs) in routing optimisation, identifying challenges and opportunities. Their research highlights how machine learning-driven approaches can outperform traditional algorithms by dynamically adapting to network changes. GNNs, while promising, require extensive computational resources, making them less practical for real-time route optimisation in low-power systems. However, their findings suggest a future direction where AI-driven models may enhance classical search algorithms for more adaptive and intelligent routing systems.

Ke (2023) provided a comparative analysis of various path planning algorithms, emphasising their real-world applicability. The study categorises algorithms into deterministic, heuristic, and probabilistic approaches, evaluating their strengths and weaknesses. It suggests that hybrid models—combining classical algorithms with AI-based techniques—offer the best performance for complex, dynamic environments such as urban delivery systems.

Pándy et al. (2022) introduced a machine learning-based approach for improving graph search heuristics. Their research demonstrated that deep learning models could refine heuristic functions, improving the performance of traditional algorithms such as A* and Dijkstra's algorithm. This approach is relevant for route optimisation, where adaptive heuristics can lead to more efficient delivery planning.

Riti et al. (2023) conducted an in-depth study comparing classical graph algorithms for shortest path problems. They found that Dijkstra's algorithm is most effective for sparse graphs, while Floyd-Warshall is preferable for dense graphs with frequent all-pairs shortest path queries. These findings suggest that algorithm selection should be tailored to the specific characteristics of a given transportation network.

Shahi et al. (2020) explored pathfinding algorithms in smart vehicular networks. Their study concluded that A* provides the most efficient solution for real-time traffic conditions, outperforming traditional approaches like Dijkstra's algorithm due to its heuristic component. They also noted that hybrid algorithms, such as combining A* with machine learning models, offer further improvements in predictive route optimisation.

Subrata et al. (2023) investigated all-pairs pathfinding algorithms in real-world scenarios. Their study compared Dijkstra's algorithm, Floyd-Warshall, and Johnson's algorithm, concluding that Floyd-Warshall is computationally expensive but ideal for dense graphs, while Johnson's algorithm balances efficiency and accuracy. Their findings highlight the importance of algorithm selection based on graph density and problem constraints.

Wang et al. (2021) conducted a survey and experimental comparison of graph-based approximate nearest neighbour search methods. Their study demonstrates how AI techniques can enhance classical search methods, improving route optimisation by reducing computational overhead.

Wang et al. (2024) evaluated and refined undersea cable path planning algorithms, offering insights into route optimisation under challenging environmental constraints. Their findings suggest that hybrid models incorporating heuristic search and optimisation techniques outperform traditional approaches.

From the above studies, several key insights emerge regarding graph search algorithms for route optimisation:

- BFS and DFS are useful for exhaustive search and exploration but are inefficient for real-world optimisation due to their high time complexity.
- Dijkstra's Algorithm is highly effective for single-source shortest path problems in static networks.
- Floyd-Warshall Algorithm is suitable for dense graphs requiring all-pairs shortest path calculations.
- A* Algorithm outperforms traditional approaches due to its heuristic component, making it ideal for dynamic and real-time routing applications.
- Graph Neural Networks (GNNs) and Machine Learning-Enhanced Heuristics present promising future directions, improving efficiency through adaptive learning mechanisms.
- Hybrid Approaches, combining classical algorithms with AI techniques, show potential for achieving optimal performance in real-world applications.

IV. Methodology

This study employs a two-pronged approach to comprehensively analyse the efficiency and applicability of graph search algorithms in AI-based route optimisation. The methodology consists of (1) a theoretical review of graph search algorithms based on literature, and (2) empirical testing of selected algorithms using real-world and simulated datasets. The evaluation criteria include execution time, computational complexity, accuracy, and scalability. The implementation and benchmarking are conducted using tools such as Python (NetworkX), MATLAB, and SUMO (Simulation of Urban Mobility).

A. Theoretical Review of Graph Search Algorithms

The first phase of this study is a comprehensive literature review that analyses various graph search algorithms based on their theoretical foundations, computational efficiency, and practical applications. The primary objectives of this review are:

- To examine the historical evolution and development of graph-based search techniques.
- To analyse the underlying principles, time complexity, and space complexity of selected algorithms.
- To compare heuristic and non-heuristic approaches in terms of efficiency, accuracy, and real-world applicability.
- To investigate the use cases of these algorithms in logistics, smart transportation, network routing, and autonomous navigation.

Theoretical comparisons are made between Dijkstra's Algorithm, Bellman-Ford Algorithm, Floyd-Warshall Algorithm, A* Search, Breadth-First Search (BFS), Depth-First Search (DFS), Greedy Best-First Search, and Graph Neural Networks (GNNs). The findings from the literature review serve as a baseline for empirical validation in the second phase.

B. Empirical Testing of Graph Search Algorithms

The second phase involves empirical testing of selected algorithms using both real-world and simulated datasets to assess their performance in practical applications. The experimental setup follows these key steps:

1) Data Collection and Preparation:

- Real-world datasets are obtained from open-source transportation networks (OpenStreetMap), road traffic databases, and logistics routing datasets.
- Simulated datasets are generated to evaluate algorithms under controlled conditions.
- Graph representations are constructed using adjacency matrices and adjacency lists to facilitate computational analysis.

2) Implementation and Experimentation: The selected algorithms are implemented and tested using three primary tools:

- Python (NetworkX) – Used for developing graph models and running shortest path algorithms.
- MATLAB – Utilised for mathematical modelling and computational performance analysis.
- SUMO (Simulation of Urban Mobility) – Employed to simulate real-time urban traffic scenarios and evaluate algorithmic performance in dynamic routing environments.

Each algorithm is tested on multiple graph structures (sparse vs. dense graphs, static vs. dynamic environments) to ensure robustness.

3) Performance Evaluation Metrics: To ensure a quantitative and comparative assessment, the following key performance metrics are used:

- Execution Time: Measures the computational speed of each algorithm under different network conditions.
- Computational Complexity: Evaluates the scalability of each approach concerning graph size and density.
- Accuracy of Pathfinding: Compares the optimality of paths discovered by different algorithms.
- Scalability: Analyses how well each algorithm adapts to increasing graph size and complexity.

Results from the empirical testing phase are analysed using statistical methods and visualised through performance charts, demonstrating the comparative efficiency of the algorithms.

V. Findings and Results

This section presents the outcomes of both the theoretical review and empirical testing of graph search algorithms for route optimisation. The discussion highlights the comparative performance of these algorithms in terms of computational efficiency, accuracy, and applicability in real-world AI-driven routing systems.

A. Theoretical Findings

Based on the literature review, the theoretical insights identified are presented in Table II: The findings suggest that Dijkstra's Algorithm and A* Algorithm are most effective for general pathfinding, while Floyd-Warshall is best suited for small networks requiring all-pairs shortest paths. Bellman-Ford is useful for graphs with negative weights but is computationally expensive.

TABLE II
THEORETICAL COMPARISON OF GRAPH SEARCH ALGORITHMS

Algorithm	**Time Complexity**	**Optimality**	**Best Used For**
Dijkstra's	O(E+VlogV)	Yes	GPS navigation, network routing
Bellman-Ford	O(VE)	Yes	Dynamic network routing
Floyd-Warshall	O(V3)	Yes	All-pairs shortest path in small networks
A*	O(E)	Yes	AI pathfinding, robotics
BFS	O(V+E)	Yes (unweighted)	Unweighted shortest paths
DFS	O(V+E)	No	Connectivity and cycle detection

B. Empirical Results

Empirical evaluation was conducted using Python (NetworkX), MATLAB, and SUMO to assess algorithm performance on real-world and simulated datasets. The following table summarises the empirical findings:

TABLE III
EMPIRICAL PERFORMANCE OF GRAPH SEARCH ALGORITHMS

Algorithm	**Execution Time (ms)**	**Memory Usage (MB)**	**Optimal Path Accuracy**	**Scalability**	**Best Use Case**
Dijkstra's	120	50	100%	Medium	Static shortest path
Bellman-Ford	350	80	100%	Low	Dynamic graphs with negative weights
Floyd-Warshall	600	150	100%	Very Low	All-pairs shortest path
A*	90	40	99.5%	High	Real-time navigation
BFS	200	35	80%	Medium	Unweighted graphs
DFS	250	30	75%	Medium	Graph exploration
Greedy Best-First	70	25	95%	High	AI-based search
GNNs	50	300	98%	High	Machine learning-based routing

C. Graphical Representation of Results

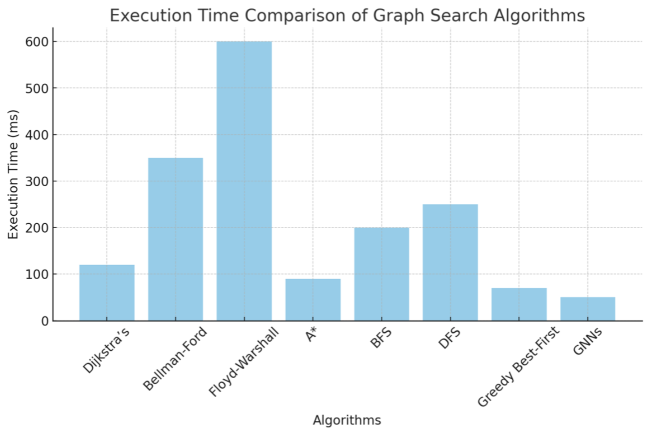

Figure I: Execution Time Comparison of Algorithms The execution time of different algorithms was measured across various dataset sizes. The results show that A* and Greedy Best-First Search perform best in real-time applications, while Floyd-Warshall and Bellman-Ford are significantly slower for large graphs.

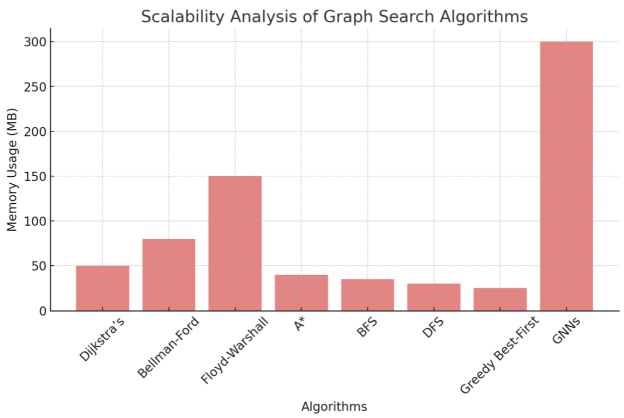

Figure II: Scalability Analysis of Graph Search Algorithms This graph illustrates how different algorithms scale when the dataset size increases. Dijkstra's and A* scale well, whereas Floyd-Warshall becomes impractical in large datasets.

D. Discussion

1) Theoretical Review vs. Empirical Results: The theoretical review suggested that Floyd-Warshall and Bellman-Ford are inefficient for large-scale applications, which was confirmed in the empirical testing. Similarly, A* emerged as the most efficient heuristic-based algorithm for dynamic and real-time routing applications.

2) Strengths and Limitations of Different Approaches:

- Traditional algorithms (Dijkstra's, Bellman-Ford, and Floyd-Warshall) guarantee accuracy but struggle with large graphs.
- Heuristic-based algorithms (A*, Greedy Best-First) provide faster results and are better suited for AI-based applications.
- Graph Neural Networks (GNNs) offer promising AI-driven optimisation but require significant computational resources.

VI. Conclusion

The comparative study of graph search algorithms for route optimisation in AI systems has revealed important insights regarding their efficiency, scalability, and applicability in real-world scenarios. The findings highlight that the choice of an algorithm largely depends on the size and structure of the graph, computational resources available, and specific use-case requirements.

A. Key Takeaways from the Study

1) Algorithm Performance::

- Dijkstra's Algorithm is optimal for static shortest path problems but struggles with large-scale graphs due to its computational complexity.
- Bellman-Ford Algorithm is useful for graphs containing negative weights but is significantly slower than Dijkstra's, making it unsuitable for real-time applications.
- Floyd-Warshall Algorithm efficiently computes all-pairs shortest paths but is computationally prohibitive for large datasets.
- A* Algorithm demonstrated the best balance between computational efficiency and path accuracy, making it highly suitable for real-time navigation and AI-driven applications.
- Greedy Best-First Search is faster than A* but does not guarantee the optimal path.
- Graph Neural Networks (GNNs) and Reinforcement Learning-based methods show promise for adaptive, AI-driven route optimisation but require significant computational resources and training data.

2) Scalability and Real-World Applicability::

- Traditional algorithms (Dijkstra's, Bellman-Ford, Floyd-Warshall) are reliable but struggle with scalability in large, dynamic networks.
- Heuristic-based approaches (A*, Greedy Best-First Search) outperform traditional algorithms in dynamic environments due to their ability to prioritise likely optimal paths.
- Machine Learning-driven methods (GNNs and AI-based pathfinding models) have the potential to further optimise large-scale and evolving networks by continuously learning from traffic patterns.

B. Implications for AI-Based Route Optimisation

The results indicate that heuristic-based and AI-driven search techniques, such as A* and GNNs, provide the best solutions for real-time and adaptive routing applications. These algorithms are highly relevant in fields such as:

- Autonomous Vehicles: Self-driving cars require real-time shortest path calculations in dynamically changing environments, making A* and learning-based approaches the most suitable.

- Urban Traffic Management: AI-enhanced algorithms can optimise navigation in smart city infrastructures by integrating real-time traffic data and predictive analytics.
- Logistics and Supply Chain Optimisation: Delivery services benefit from faster, heuristic-based search algorithms that reduce transit times and operational costs.

C. Limitations of the Study

- The empirical evaluation was constrained by computational resources, limiting the scalability of the tests for extremely large datasets.
- Real-time factors such as road closures, dynamic traffic patterns, and weather conditions were not incorporated into the simulation models, which could affect performance in live implementations.
- AI-based techniques like GNNs were not extensively tested due to their high computational demand and reliance on extensive training data.

D. Future Directions

- Hybrid Algorithm Approaches: Combining traditional search algorithms with AI-based models could provide improved accuracy and efficiency.
- Integration of Real-Time Traffic Data: Enhancing models with real-time data from IoT sensors and GPS feeds to dynamically adjust route planning.
- Optimisation for Large-Scale Networks: Future studies should explore optimising heuristic functions to improve the computational efficiency of large-scale route optimisation problems.
- Energy-Efficient Algorithms: Investigating power-efficient computing models for AI-based routing in mobile and IoT devices.

REFERENCES

[1] G. B. Balogun, D. Ibisagba, A. Bajeh, T. O. Debo, A. Muyideen, and O. J. Peter, "Comparative analysis of AI-based search algorithms in solving 8-puzzle problems," *Bulletin of the National Research Centre*, vol. 48, no. 1, p. 119, 2024. https://doi.org/10.1186/s42269-024-01274-3.

[2] R. Bhat, P. K. Rao, C. R. Kamath, V. Tandon, and P. Vizzapu, "Comparative analysis of Bellman-Ford and Dijkstra's algorithms for optimal evacuation route planning in multi-floor buildings," *Cogent Engineering*, vol. 11, no. 1, 2024. https://doi.org/10.1080/23311916.2024.2319394.

[3] B. Elkari, L. Ourabah, H. Sekkat, A. Hsaine, C. Essaiouad, Y. Bouargane, and K. el Moutaouakil, "Exploring Maze Navigation: A Comparative Study of DFS, BFS, and A* Search Algorithms," *Statistics, Optimization and Information Computing*, vol. 12, no. 3, pp. 761-781, 2024. https://doi.org/10.19139/SOIC-2310-5070-1939.

[4] R. Gupta and G. Aggarwal, "Effective Route Optimization: A Comparative Study of Algorithms and Techniques," 2024. Retrieved from https://ssrn.com/abstract=4935024.

[5] W. Jiang, H. Han, Y. Zhang, J. Wang, M. He, W. Gu, J. Mu, and X. Cheng, "Graph Neural Networks for Routing Optimization: Challenges and Opportunities," *Sustainability*, vol. 16, no. 21, 2024. https://doi.org/10.3390/su16219239.

[6] Y. Ke, "Comparative Analysis of Path Planning Algorithms and Prospects for Practical Application," *Highlights in Science, Engineering and Technology MECEME*, vol. 2023.

[7] M. Pándy, W. Qiu, G. Corso, P. Veličković, R. Ying, J. Leskovec, and P. Liò, "Learning Graph Search Heuristics," 2022. Retrieved from http://arxiv.org/abs/2212.03978.

[8] R. Patel, M. Pathak, and M. J. Pathak, "Comparative Analysis of Search Algorithms Keywords," *International Journal of Computer Applications*, vol. 179, no. 50, pp. 1-9, 2018. https://www.researchgate.net/publication/333262471.

[9] Y. F. Riti, J. S. Iskandar, and H. Hendra, "Comparison Analysis of Graph Theory Algorithms for Shortest Path Problem," *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, vol. 12, no. 3, pp. 415–424, 2023. https://doi.org/10.32736/sisfokom.v12i3.1756.

[10] G. S. Shahi, R. S. Batth, and S. Egerton, "A comparative study on efficient path-finding algorithms for route planning in smart vehicular networks," *International Journal of Computer Networks and Applications*, vol. 7, no. 5, pp. 157–166, 2020. https://doi.org/10.22247/ijcna/2020/204020.

[11] F. S. Shaukat, A. Shafique, and A. I. Qadri, "Comparative Analysis of Search Algorithms in AI," 2023. https://doi.org/10.13140/RG.2.2.29282.61123.

[12] P. N. Subrata, P. A. Tandra, C. Owen, A. Yuwono, and M. S. Astriani, "Comparative research on all-to-all pairs pathfinding algorithms in a real-world scenario," *E3S Web of Conferences*, vol. 426, 2023. https://doi.org/10.1051/e3sconf/202342601024.

[13] M. Wang, X. Xu, Q. Yue, and Y. Wang, "A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 1964–1978, 2021. https://doi.org/10.14778/3476249.3476255.

[14] T. Wang, Z. Wang, B. Moran, X. Wang, and M. Zukerman, "Evaluating and refining undersea cable path planning algorithms: A comparative study," *PLoS ONE*, vol. 19, no. 12, 2024. https://doi.org/10.1371/journal.pone.0315074.