https://acadpub.com/ics
Insights in
Computer Science

A Comparative Analysis of Dijkstra, Bellman-Ford and AODV Algorithms with Applications in Modern Navigation Systems

Huzaima Farooq

Department of Computer Science, University of Salford, 43 Crescent, Salford, Greater Manchester, M5 4WT, England

About the Article

Open Access

Systematic Review

How to Cite: H. Farooq, "A comparative analysis of dijkstra, bellman-ford and AODV algorithms with applications in modern navigation systems," Insights Comput. Sci., Vol. 1, pp. 15-18. 2025.

Keywords:
AODV routing protocol, bellman–ford algorithm, Dijkstra's algorithm, routing algorithms, satellite navigation systems

Corresponding author:
Huzaima Farooq,
Department of Computer Science, University of Salford, 43 Crescent, Salford, Salford, Greater Manchester, M5 4WT, England

ABSTRACT

Background and Objective: This study aimed to examine and compare the theoretical foundations, computational characteristics and practical applications of three fundamental routing algorithms-Dijkstra's Algorithm, the Bellman–Ford Algorithm and the Ad hoc On-Demand Distance Vector (AODV) Routing Protocol. Particular emphasis was placed on their relevance to contemporary satellite navigation (SatNav) and digital mapping systems.
Materials and Methods: A systematic review of scholarly literature and technical documentation was conducted to analyze the operational principles, strengths and limitations of each algorithm. Comparative assessment focused on deterministic shortest-path computation, distributed routing behavior and reactive route discovery. Algorithmic evolution toward hybrid models incorporating heuristics and real-time traffic data was also examined.
Results: The analysis showed that deterministic algorithms such as Dijkstra's have undergone significant optimization, enabling their integration into large-scale navigation platforms including Google Maps, Waze and Apple Maps. Bellman-Ford demonstrated advantages in distributed environments, while AODV exhibited strong adaptability in highly dynamic, infrastructure-less networks. Each algorithm displayed distinct performance characteristics with respect to scalability, routing overhead and responsiveness to network changes.
Conclusion: The findings indicate that Dijkstra's algorithm and its variants (particularly heuristic-enhanced forms such as A*) remain the most efficient and scalable solutions for SatNav applications requiring real-time routing updates. In contrast, AODV continues to be better suited for ad hoc and vehicular network scenarios where dynamic topology and decentralized route discovery are critical.

INTRODUCTION

Routing algorithms constitute the fundamental basis of both computer communication networks and real-world navigation systems. At their core, these algorithms determine the most efficient path between two or more nodes within a network graph. Whether the purpose is to transmit data packets across digital infrastructures or to guide vehicles through complex urban environments, routing fundamentally relies on graph theory, optimization techniques and dynamic decision-making[1].

In this study, three prominent routing algorithms representing distinct methodological paradigms were examined: Dijkstra's algorithm (a deterministic, link-state method), the Bellman-Ford algorithm (a distributed, distance-vector approach) and the Ad hoc On-Demand Distance Vector (AODV) protocol (a reactive, on-demand routing method)[2-5]. Each of these algorithms has significantly shaped routing mechanisms in both wired and wireless networks, as well as the real-time navigation platforms used by millions of users worldwide.

Understanding the mechanics of these algorithms revealed that routing efficiency affects not only computational performance but also overall user experience, particularly in applications such as Google Maps where milliseconds influence route responsiveness. The purpose of this investigation was therefore to connect theoretical algorithmic principles with their practical implementations in global positioning and satellite navigation (SatNav) systems, while also evaluating their comparative strengths and limitations.

Dijkstra's algorithm: Dijkstra's algorithm, introduced by Dijkstra[6], is a deterministic link-state shortest-path algorithm designed to compute the minimum-cost route from a source node to all other nodes in a graph with non-negative edge weights. The algorithm iteratively selects the vertex with the smallest tentative distance and relaxes all adjacent edges until the shortest paths are fully determined.

Its time complexity ranges from $O(V^2)$ when implemented with adjacency matrices to $O(E+V\log V)$ when min-priority queues are used[7]. Due to its predictability and efficiency, Dijkstra's algorithm forms the foundation of widely deployed routing protocols such as OSPF (Open Shortest Path First) and IS-IS (Intermediate System to Intermediate System). In contemporary navigation systems-including Google Maps, Apple Maps and Waze-Dijkstra's principles are applied through enhanced variants such as A* and Contraction Hierarchies (CH), which reduce computational latency[8]. These optimizations integrate heuristic estimates based on geographic constraints, road network topology and live traffic conditions.

Bellman-ford algorithm: Proposed by Richard Bellman[9], the Bellman-Ford algorithm adopts a distance-vector strategy in which distance estimates are iteratively refined by relaxing all edges up to (V-1) times. Unlike Dijkstra's algorithm, Bellman-Ford supports negative edge weights, making it suitable for analyzing networks with variable or dynamic cost structures.

Although, its time complexity of $O(V \times E)$ renders it computationally slower, Bellman-Ford is more robust in detecting routing loops and negative cycles. Its distributed design influenced early routing protocols, most notably the Routing Information Protocol (RIP)[10]. Despite its strengths, the slow convergence and high communication overhead associated with Bellman-Ford limit its practicality for large-scale or real-time applications such as SatNav. Nevertheless, the algorithm offers valuable insights into distributed decision-making and the trade-offs among accuracy, convergence time and computational cost.

Ad hoc on-demand distance vector (AODV) algorithm: Developed by Perkins and Royer[11], the Ad hoc On-Demand Distance Vector (AODV) algorithm is a reactive routing protocol designed for mobile ad hoc networks (MANETs). AODV establishes routes only when required, using Route Request (RREQ) and Route Reply (RREP) messages to minimize routing overhead in highly dynamic environments. This design contrasts with proactive protocols that maintain continuous routing tables regardless of network activity.

AODV's performance depends on node mobility and route request frequency. It demonstrates strong effectiveness in vehicular ad hoc networks (VANETs) and emergency communication systems, where network topologies change rapidly[12]. Although, AODV is not directly implemented in modern SatNav applications, its reactive principles influence dynamic re-routing strategies in systems such as Waze, where user-generated traffic data triggers real-time route adaptations.

Objectives of the study: This research was guided by the following objectives:

- To understand the algorithmic design and operational principles of Dijkstra's, Bellman-Ford and AODV algorithms
- To compare their complexity, scalability, convergence characteristics and adaptability in dynamic environments
- To explore their applicability to modern navigation platforms, including Google Maps, Waze, Apple Maps and traditional GPS-based SatNav systems
- To examine how algorithmic efficiency influences real-world user experience and system performance in contemporary routing applications

MATERIALS AND METHODS

This study employed a combined theoretical and simulation-based methodological approach. Graph topologies were modeled using the NetworkX library in Python to evaluate the performance of Dijkstra's and Bellman-Ford algorithms under static network conditions. For the analysis of AODV, simulations conducted in the ns-3 network simulator were consulted, providing detailed observations of dynamic packet routing behavior in mobile and fluctuating network environments.

The algorithms were assessed using the following evaluation parameters:

- Computational complexity and scalability
- Convergence time and routing stability
- Adaptability to dynamic and rapidly changing topologies
- Energy and resource efficiency in constrained environments
- Suitability for navigation-oriented and real-time routing applications

In addition, industry-level routing systems were examined through a review of white papers, technical reports and documentation from Google, Apple and OpenStreetMap. This analysis facilitated an understanding of how theoretical routing principles are operationalized within large-scale navigation and mapping platforms.

Comparative analysis: Algorithmic comparison is presented in Table 1.

RESULTS AND DISCUSSION

The analysis revealed that Dijkstra's algorithm continues to serve as the fundamental basis of contemporary route-planning systems. Its deterministic structure provides high consistency and predictability, while modern algorithmic enhancements enable the incorporation of real-time variables such as traffic congestion, speed limits and temporary road closures. Google Maps, for example, applies Dijkstra's framework in combination with the A* informed search algorithm to prioritize nodes based on both geographic distance and estimated travel time[13]. The heuristic function in A* reduces unnecessary node expansions, enabling sub-second route computation across large, complex networks[8].

In contrast, the iterative nature of the Bellman–Ford algorithm results in significantly slower execution times, rendering it impractical for real-time navigation. Nevertheless, its ability to operate in fully distributed environments makes it valuable for systems that rely on continuous local updates, such as autonomous sensor networks. Similarly, while AODV demonstrates strong performance in mobile ad hoc and peer-to-peer wireless networks, it is not efficient for large-scale graph structures. Its reactive, on-demand routing principles-where paths are constructed only when required-are conceptually reflected in Waze's user-driven re-routing system, in which crowd-sourced incident reports dynamically adjust local routing decisions[12].

Applications in modern navigation systems

Google maps: Google Maps integrates Dijkstra's algorithm with A* heuristics and machine learning models designed to predict short-term traffic behavior[13]. The underlying road network is represented as a weighted directed graph, with edge weights corresponding to travel time and continuously updated using real-time data from millions of Android devices. This hybrid approach ensures that selected routes optimize not only spatial distance but also temporal efficiency.

Furthermore, Google Maps employs Contraction Hierarchies to pre-compute multi-level abstractions of the routing graph, reducing the computational load of online queries to the millisecond scale[8]. Thus, while Dijkstra's principles remain foundational, the system's performance relies on predictive analytics, hierarchical pre-processing and large-scale distributed infrastructure.

Waze and apple maps: Waze utilizes a similar graph-based routing model but places greater emphasis on real-time, crowd-sourced traffic and incident reports. This allows the platform to adjust edge weights dynamically and initiate re-routing only when changes are detected, functioning as a semi-reactive system analogous to the principles of AODV. Apple Maps incorporates both historical and real-time traffic patterns, employing predictive modeling techniques that extend Dijkstra's deterministic framework to forecast road conditions several minutes ahead[14]. This enables the system to anticipate potential congestion and propose time-optimized routes before delays occur.

GPS-based SatNav systems: Traditional GPS-based SatNav systems (e.g., TomTom, Garmin) primarily rely on pre-computed routing derived from Dijkstra's algorithm. Due to limited real-time data connectivity, these systems do not exhibit the adaptive behavior characteristic of AODV-like protocols. Instead, they prioritize stability and reproducibility, depending on periodic map updates rather than continuous live recalculation. Results and observations on the overall all examined systems are presented in Table 2.

Overall findings: Across all examined systems, Dijkstra's algorithm-particularly it's a* and Contraction Hierarchies variants-emerges as the most effective approach for large-scale SatNav and digital mapping applications. These

Table 1: Algorithmic comparison

Parameter	Dijkstra's algorithm	Bellman-ford algorithm	AODV algorithm
Routing type	Link-state	Distance-vector	Reactive/on-demand
Complexity	O(E+V logV)	O(V×E)	Variable
Negative weights	Not supported	Supported	N/A
Convergence speed	Fast	Slow	Dynamic
Adaptability	Moderate	Low	High
Scalability	High	Moderate	High
Best use case	Wired, static networks	Distributed systems	Mobile Ad Hoc networks
Example systems	Google maps, apple maps	RIP, ARPANET	VANETs, UAV routing

Table 2: Results and observations

Metric	Dijkstra	Bellman-ford	AODV	Real-world situation
Path optimality	High	Medium	Variable	Dijkstra/A*
Execution speed	Fast	Slow	Variable	Dijkstra
Adaptability	Moderate	Low	High	AODV (for dynamic)
Scalability	Excellent	Limited	good	Dijkstra
traffic adaptation	High (with heuristics)	Low	Moderate	Dijkstra/A*
Energy efficiency	Moderate	Low	High (on demand)	AODV
Best for navigation apps	✓	×	☼	Dijkstra/A*

methods integrate deterministic shortest-path computation with heuristic optimization and real-time environmental awareness, enabling efficient, scalable and low-latency route planning.

CONCLUSION

This study highlights the fundamental role of classical graph algorithms in shaping contemporary intelligent navigation systems. The analysis demonstrated that although Dijkstra's algorithm remains highly effective and computationally efficient for static network environments, its practical utility is significantly enhanced when integrated with heuristic techniques and real-time data processing. The Bellman-Ford algorithm provided broader insight into distributed routing and iterative path optimization, whereas the AODV protocol illustrated the advantages of reactive routing strategies in dynamic and decentralized mobile networks. Overall, the findings indicate that the selection of an appropriate routing algorithm extends beyond computational considerations and reflects a balance among predictability, adaptability and scalability. The operational frameworks used in real-world navigation platforms, such as Google Maps and Waze, exemplify how theoretical principles in computer science translate into functional navigation intelligence. These observations emphasize the need for continued research in adaptive routing, algorithmic optimization and the development of efficient models capable of supporting increasingly complex and dynamic network environments.

REFERENCES

[1] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 6th ed. Harlow, UK: Pearson, 2021, pp. 922.

[2] Z. Grujic and B. Grujic, "Optimal routing in urban road networks: A graph-based approach using Dijkstra's algorithm," *Appl. Sci.*, vol. 15, no. 8, 2025. [Online]. Available: 10.3390/app15084162

[3] O. Timofeeva, A. Sannikov, M. Stepanenko and T. Balashova, "Modification of the Bellman–Ford algorithm for finding the optimal route in multilayer network structures," *Computation*, vol. 11, no. 4, 2023. [Online]. Available: 10.3390/computation11040074

[4] L. Chiu-Kuo and W. Hsi-Shu, "An ad hoc on-demand routing protocol with alternate routes," in *New Horizons of Parallel and Distributed Computing*, M. Guo and L. T. Yang Eds., Boston, MA: Springer US, 2005, pp. 145-156.

[5] E. M. Belding-Royer and C. E. Perkins, "Evolution and future directions of the ad hoc on-demand distance-vector routing protocol," *Ad Hoc Networks*, vol. 1, no. 1, pp. 125-150. 2003.

[6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269-271. 1959.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, Massachusetts, USA: The MIT Press, 2022, pp.1312.

[8] H. Bast *et al.*, "Route planning in transportation networks," in *Algorithm Engineering*, L. Kliemann and P. Sanders Eds., Cham: Springer International Publishing, 2016, pp. 19-80.

[9] R. Bellman, "On a routing problem," *Q. Appl. Math.*, vol. 16, no. 1, pp. 87-90. 1958.

[10] C. Huitema, *Routing in the Internet*, 2nd ed. Paramus, New Jersey, USA: Prentice Hall, 1999, pp.384.

[11] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999. New Orleans, LA, USA: IEEE, 1999, pp. 90-100.

[12] M. H. Hassan *et al.*, "Mobile ad-hoc network routing protocols of time-critical events for search and rescue missions," *Bull. Electr. Eng. Inf.*, vol. 10, no. 1, pp. 192-199. 2021.

[13] Google AI Blog, "How Google Maps calculates the fastest routes," Dec. 2020. [Online]. Available: https://blog.google/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/.

[14] Apple Inc., "Maps," Apple, 2023. [Online]. Available: https://www.apple.com/maps/. [Accessed: 29-Nov-2025].