Proceedings of the 4th International Conference on Applied Physics and Mathematical Modeling
DOI: 10.54254/2753-8818/2026.36559

Graph-Based Route Planning for Map Navigation Using Shortest-Path Algorithms

Qianrui Cheng

Shanghai Pinghe School, Shanghai, China
chengqianrui@shphschool.com

Abstract. Map navigation and graph-based path planning are the foundations of intelligent transportation systems today, supporting all kinds of applications in daily life, logistics and emergency rescue. A spatially weighted graph of the real road network is built in the system, and then the best path is selected based on the weights of these factors. This paper will introduce the main methods of the study and first present road-network models that combine static data from sources such as OpenStreetMap with dynamic Global Positioning System (GPS) trajectory data to construct and update weighted directed graphs. Next, this paper will introduce the basic and advanced shortest-path algorithms for static and dynamic urban environments, such as Dijkstra's algorithm, A* algorithm, etc., and compare their strengths and weaknesses. The following are the steps of the whole navigation workflow, as well as data preprocessing, multi-source fusion and execution of multi-objective path search. Lastly, this paper will also discuss the current deficiencies and problems in the above methods, such as missing data, algorithmic complexity of large-scale networks, trade-offs in multi-objective optimisation, GPS errors and user privacy, etc. Graph theory, spatial data science and optimisation algorithms have been combined to provide the theoretical basis for intelligent path planning in this paper and offer a reference for future research on lightweight, high-efficiency dynamic navigation.

Keywords: map navigation, route planning, weighted graphs, shortest-path algorithms, spatial data

1. Introduction

Map navigation and graph-based path planning are the basic technologies for many aspects of urban life, logistics distribution, emergency rescue, shared mobility and intelligent transportation systems. Navigation Platforms build a spatially weighted graph of the physical road network and then optimise routes in light of all sorts of constraints, such as distance, travel time, road hierarchy, turn restrictions, real-time traffic conditions, etc. [1, 2]. Open map resources, a large number of Global Positioning System (GPS) trajectories, roadside traffic information and location-based service data have been used, and to date, research has focused on the following four areas: road-network graph modelling, efficient shortest-path search, dynamic edge-weight update and multi-objective route optimisation. This paper organizes the latest research results in the field of road network modelling and path search algorithms; many of these are based on open spatial data, such as OpenStreetMap,

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

Geolife GPS trajectories and T-Drive taxi trajectories, and consist of weighted directed graphs with nodes, road segment lengths, driving speeds, time-varying traffic conditions and road attribute labels [3-6]. In the next section, this paper will compare the application scope, merits and demerits of classical shortest-path algorithms for static and dynamic urban navigation problems, such as Dijkstra's algorithm, A* algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm, bidirectional search and hierarchical network acceleration method. Based on the above, the research objectives of this paper are to systematically organize the theoretical basis of graph-driven navigation planning, explain how spatial road data is utilized to support route retrieval and multi-option route comparison, and analyze the fundamental ideas of graph theory, linear-algebraic graph representation and data science in the construction of an entire navigation system. It has provided some theoretical support for the study of light-weight, high-efficiency dynamic navigation in cities [2].

2. Basic concepts of road networks, spatial data and weighted graphs

In order to realise automatic map navigation and route optimisation, it is necessary to abstract the physical transportation infrastructure of cities into computable weighted graph structures that can be used as the mathematical foundation for graph-based path-finding algorithms in navigation systems [7]. A directed graph is strongly connected if every node can be reached from all other nodes by a directed path. It is weakly connected if the corresponding undirected graph is connected. Nodes in traffic modelling are the intersections of roads, interchanges, traffic light positions, etc., and each node has corresponding longitude and latitude coordinates as basic spatial attributes [6]. Edges connect two nodes to represent actual road sections; according to traffic regulations, the graph can be divided into undirected and directed graphs. Undirected edges are undirected roads that allow two-way travel, and directed edges are one-way streets that permit travel only from the start node to the end node. Fundamental graph properties include connectivity, paths and cycles. A path is a continuous sequence of adjacent edges from the start node to the end node, and a cycle is a closed path that begins and ends at the same node, which may result in unnecessary detours in the route calculation.

All edges have a numerical weight that shows the cost of crossing them, and the multi-dimensional traffic indicators have been converted into weights according to the standard quantitative mapping rules. According to the latitude and longitude coordinates of the nodes, the Euclidean or spherical distance to the road segments can be computed, and these can then be set as fixed-distance weights. Based on the actual travel time, these are used to adjust the weights of speed, speed limit and time-varying congestion. Road hierarchy information can be assigned a higher weight coefficient to give more weight to high-class roads in the route selection by differentiating them from residential side streets. The weight of the edge or node can be increased in accordance with the actual traffic restrictions; the weight can be set to infinity for prohibited turns, and extra time penalties can be added for left turns, sharp curves and signal-controlled intersections to reflect the cost of turning. The two common ways to store the road network graph are an adjacency matrix and an adjacency list. Among them, the adjacency matrix is a two-dimensional array, and both the number of rows and columns are equal to the total number of vertices in the graph. Each element in this array stores the weight of the directed edge from the start node to the end node; if there is no direct edge between two nodes, then the corresponding position will be set to infinity. There is a structure with O(1) edge-weight queries, but the memory required for a large-scale urban road network (hundreds of thousands of nodes) is too large. To reduce the amount of memory, the neighbours and edge weights of all vertices in a sparse road network can be stored using an adjacency list [7].

All the navigation systems based on graphs can use the six kinds of spatial data to construct and dynamically update weighted roadmaps. First, roadmap data from OpenStreetMap (OSM) can be loaded to obtain static topological information, such as road geometry, one-way markings, speed limits, etc., which form the foundation of the graph model [6]. Crowdsourced vehicle trajectories can also add the missing turning constraints and partial intersection topology to the raw OSM data [3].

Second, there are many trajectory samples of T-Drive taxi records and GPS trajectory data, as well as Geolife human movement trajectories [4, 5]. Map matching is employed to map the discrete trajectory points onto the edges of the road network, and thus, time-varying driving weights are given according to different peak and off-peak periods; low-quality GPS sampling data can be improved by a special trajectory enhancement algorithm before matching [8], and map matching can also correct noisy GPS trajectory points based on the road network [9]. Third, based on the traffic speed data from roadside sensors and floating cars at any time, the weights will change every minute, and sudden traffic jams or temporary road closures will also be indicated. Fourth, data on points of interest (POIs) such as shopping centres, parking lots and service stations are added as auxiliary map attributes to support all kinds of route functions, such as refueling and resting. Fifth, Origin-Destination (OD) data contain the origin and destination of trips and can be used to estimate travel demand and calibrate route-choice models. Sixth, with a long history of traffic data, a regular traffic pattern and future time-dependent edge weights can be constructed. According to the graph theory mentioned above, many sources of spatial data have been integrated to build a complete route-planning technology framework; static OSM data provide a fixed topology structure and weighted graphs, trajectory and real-time traffic data continuously update the edge weights, and personalised optimisation constraints are added by incorporating point-of-interest (POI) and historical Origin-Destination (OD) data. The standard model is a general way to represent abstract graph theory and geographic spatial features, so Dijkstra's algorithm and the A* algorithm can be applied to solve problems in urban navigation.

3. Shortest-path algorithms and route optimisation methods

The navigation map algorithm and acceleration method in modern map navigation are used to find a good route in various travel circumstances. Dijkstra's algorithm is the shortest-path algorithm for graphs that do not have negative weights. It is efficient at searching sparse road networks represented by adjacency lists to find the shortest distance or travel time [1]. A* has a heuristic function to estimate the cost from the current node to the end, thereby narrowing the search; it should be admissible and consistent for this reason [10]. To improve the efficiency of large-scale navigation, simultaneous bidirectional searches are carried out from both the start and end points, and contraction hierarchies can be used to reduce query overhead significantly by contracting nodes and adding shortcuts in the road network beforehand [11, 12]. The algorithms above can be used to solve multi-objective path optimisation problems and balance all kinds of factors according to different travel needs, such as short distance, short travel time, low fuel cost and few turns [1, 5]. Preprocessing of the basic data can improve the accuracy of the algorithm's results; therefore, it is necessary to clean the road network of invalid road sections, correct GPS trajectory errors by map matching, and add multiple sources of traffic data to update dynamic edge weights. When learning-based travel-time or edge-weight models are employed, the data is split into training, validation and test sets, and only the hyperparameters need to be optimised with the validation set. General

DOI: 10.54254/2753-8818/2026.36559

evaluation indices are path length, predicted travel time error, algorithm computation time, path stability in the presence of traffic and actual arrival deviation. The above indicators show the practicality and stability of all the navigation algorithms.

4. State-of-the-art research on graph-based map navigation and route planning

At present, there are considerable differences in the data resources, experimental environments and technical logic of research on road network analysis and navigation path planning; thus, different route effects are produced in various travel scenarios [1]. Most of the main studies use the static road topology of the OpenStreetMap built by OSMnx as the base network skeleton [6] and add crowdsourced trajectory data to correct missing turning and traffic attribute information in the original maps [3].

The size of the network can be a small community-level road network with a few hundred nodes or a large metropolitan road network, and at the same time, both an adjacency matrix and an adjacency list can be chosen; this choice will affect the algorithm's efficiency. The definition of edge weight has also been mentioned in the above studies, and some studies have set the geometric road length as a fixed, static weight; others, however, have included many other factors such as real-time vehicle speed, traffic congestion delay, turning penalty points and road tolls to form multi-dimensional dynamic weights that are updated at different frequencies according to performance needs in real time, sometimes every few hours or even every minute.

Given all the kinds of demands for the path-search and optimisation module, scholars have chosen Dijkstra's algorithm, A* algorithm, bidirectional search or hierarchical acceleration algorithm. Driving-Navigation studies generally focus on travel time and convenience. Public-Transport Planning will consider transfer time and walking distance. Logistics Routing Optimises Travel Time, Fuel Costs and Delivery Order. Emergency evacuation planning considers route capacity and pedestrian-flow constraints [13].

Static road network models have a small update and computation cost, but they cannot be changed according to traffic fluctuations over time. Dynamic Traffic Models can be modified according to fluctuations in traffic, but they are too complicated and need frequent updates.

Several factors affect the final path. Incomplete or noisy spatial data will result in path selection bias, large-scale urban networks can be difficult to process, and the edge weights may not be updated promptly according to changes in traffic. A single goal may also fail to reflect users' preferences for fewer stops or lower travel fees.

A typical research process is to abstract the intersections and road segments of a simple city center road network into a weighted graph, set the coordinates of the start and end points, perform graph search algorithms, and then analyze the difference between the distance-optimal path and the time-optimal path; this can be done without complicated programming. In addition to the practical navigation function, this paper will also present the theoretical basis for the whole planning process: Graph theory is used to model the topology of road networks; Spatial geographic data are employed to assign computable weights to different kinds of traffic attributes; and data science methods help to clean, integrate and dynamically update the weights of multi-source traffic data. The theoretical foundation for many modern intelligent route planning systems is the application of mathematics and data science.

DOI: 10.54254/2753-8818/2026.36559

5. Limitations and outlook

Although the current map navigation and path planning technology has brought people some convenience, it also has many deficiencies, such as problems with data, algorithms, multi-objective trade-offs and external factors.

Missing data and slow updates are also the reasons why there are problems with missing road information or incorrect attribute labels in the basic map data (such as one-way roads and turning restrictions); these can be supplemented and corrected by crowdsourced trajectory data [3]. Although the urban road network changes often, the map updates have not kept up with this change; therefore, the navigation system has not been informed promptly about newly opened or closed roads and may plan an invalid route.

Although some real-time traffic information can be obtained from the GPS data of floating vehicles (such as taxis) and roadside sensors, there are still problems of incomplete coverage and sampling bias in this data. For example, it is not easy to determine the driving speed accurately on roads with low traffic. There is also a certain delay between the occurrence of sudden traffic accidents or temporary traffic control and the system's awareness and update. GPS positioning may experience drift, while GPS signals may be attenuated or lost in areas such as urban canyons, high-rise buildings, tunnels and underground parking lots, so the location information may be inaccurate. This will not only affect the user's awareness of their current position but also cause problems for the following map matching algorithm; it will be difficult to determine accurately which road a vehicle is on at present [9]. In terms of algorithms and computing resources, the road network graph of a modern city will have a large number of nodes and edges. Given that such a graph is quite large, using Dijkstra's algorithm will require substantial computational resources and memory, and real-time operation cannot be guaranteed. Acceleration technology can address this problem; for instance, there are the A* algorithm and contraction hierarchies [11, 12], but saving and processing a large volume of road network data is still very expensive. Due to the problem of trade-off in multi-objective optimisation, users have many different and personalised requirements for routes, and often need to balance several objectives, such as short travel time, short travel distance, low cost, fewer turns, good scenery, etc. These goals sometimes need to be balanced, as they may not all be achievable at the same time. The development of algorithms to generate a small set of preference-aware Pareto-optimal routes in a short time is still a difficult multi-objective optimisation problem. In order to give timely direction and personalised services, navigation applications regularly collect a large amount of users' personal data, such as location, speed, past travel records, etc. How to use the data to improve services while protecting user privacy and preventing data leakage and misuse is a very serious problem for the industry. Navigation systems can generally forecast and plan based on past data and the current traffic situation, but they may be slow to react to emergencies in bad weather, sudden traffic restrictions or traffic jams caused by major events, etc. A sudden change will immediately affect the traffic capacity of the road network and make the "optimal route" provided by the system unreliable; in fact, more vehicles will be directed to already congested areas and traffic jams will worsen.

6. Conclusion

Graph-based Route Planning for Map Navigation is carried out by combining the structure of roads, geographical information and the discovery of the shortest path. Physical roads can be modelled as weighted directed graphs where the nodes are intersections, the edges are road segments, and the weights are distances, travel times, turning penalties, tolls or congestion. Classical methods are still good because their assumptions and computational characteristics are well-known. Dijkstra's algorithm is for non-negative edge weights; A* can reduce the search space with an admissible heuristic; Bellman-Ford supports negative edges but should not have reachable negative cycles; Floyd-Warshall can find all-pairs shortest paths in relatively small networks. Bidirectional Search and Contraction Hierarchies are used in the construction of large-city networks.

According to the above research, route quality is not determined solely by the search algorithm. OpenStreetMap data, GPS trajectories, roadside sensor data, origin-destination records, etc., should be cleaned, matched, fused and updated to obtain reliable edge weights. Therefore, the problems of dynamic navigation are as follows: incomplete map data, delayed traffic updates, GPS errors, high computational cost, conflict between user goals and location privacy. In the evaluation, not only the path length and travel time but also the query latency, route stability, robustness to changes in traffic, and the quality of alternative routes should be considered.

In the future, good graph search will be combined with more accurate travel-time prediction, uncertainty estimation, privacy-preserving data processing and preference-aware multi-objective optimisation. Lightweight dynamic models can update only the graph area that has changed and are thus more responsive. More open evaluation rules and realistic large-scale data will also be available for the comparison of methods. Graph theory and spatial data science have together provided an excellent foundation for intelligent navigation, but problems in data quality, algorithmic efficiency, dynamic changes and user needs still need to be addressed in order to achieve good application results.

References

[1] Tian, Z., Yao, H., & Shao, Y. (2025). A review of urban path planning algorithms in intelligent transportation systems. *Algorithms, 18*(11), 676. https://doi.org/10.3390/a18110676

[2] Abirami, S., Pethuraj, M., Uthayakumar, M., & Chitra, P. (2024). A systematic survey on big data and artificial intelligence algorithms for intelligent transportation system. *Case Studies on Transport Policy, 17*, 101247. https://doi.org/10.1016/j.cstp.2024.101247

[3] Chen, X., Xiang, L., Jiao, F., & Wu, H. (2023). Detecting turning relationships and time restrictions of OSM road intersections from crowdsourced trajectories. *ISPRS International Journal of Geo-Information, 12*(9), 372. https://doi.org/10.3390/ijgi12090372

[4] Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., & Huang, Y. (2010). T-Drive: Driving directions based on taxi trajectories. In *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 99-108). ACM. https://doi.org/10.1145/1869790.1869807

[5] Teimouri, F., Richter, K.-F., & Hochmair, H. H. (2023). Analysis of route choice based on path characteristics using Geolife GPS trajectories. *Journal of Location Based Services, 17*(3), 271-297. https://doi.org/10.1080/17489725.2023.2229285

[6] Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems, 65*, 126-139. https://doi.org/10.1016/j.compenvurbsys.2017.05.004

[7] Bachechi, C., & Po, L. (2022). Road network graph representation for traffic analysis and routing. In *Advances in Databases and Information Systems: 26th European Conference, ADBIS 2022, Turin, Italy, September 5-8, 2022, Proceedings (Lecture Notes in Computer Science, Vol. 13389)*, pp. 75-89. Springer. https://doi.org/10.1007/978-3-031-15740-0_7

[8] Li, B., Cai, Z., Wei, H., Su, S., Cao, W., Niu, Y., & Wang, H. (2026). A quality enhancement method for vehicle trajectory data using onboard images. *Geo-spatial Information Science, 29*(1), 546-571. https://doi.org/10.1080/10095020.2025.2497402

[9] Jiang, L., Chen, C., Chen, C., Huang, H., & Guo, B. (2022). From driving trajectories to driving paths: A survey on map-matching algorithms. *CCF Transactions on Pervasive Computing and Interaction, 4*(3), 252-267. https://doi.org/10.1007/s42486-022-00101-w

[10] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics, 4*(2), 100-107. https://doi.org/10.1109/TSSC.1968.300136

[11] Chen, Z., Feng, B., Yuan, L., Lin, X., & Wang, L. (2023). Fully dynamic contraction hierarchies with label restrictions on road networks. *Data Science and Engineering, 8*(3), 263–278. https://doi.org/10.1007/s41019-023-00227-6

[12] Geisberger, R., Sanders, P., Schultes, D., & Vetter, C. (2012). Exact routing in large road networks using contraction hierarchies. *Transportation Science, 46*(3), 388–404. https://doi.org/10.1287/trsc.1110.0401

[13] Cabrera, N., Huertas, J. A., & Medaglia, A. L. (2022). Pedestrian evacuation planning: Unveiling evacuation routes via column generation. *EURO Journal on Transportation and Logistics, 11*, 100089. https://doi.org/10.1016/j.ejtl.2022.100089