[1]

From Data to Routes: A Comprehensive Approach to Public Transport Line Routing

BLIND REVIEW

***Abstract*—Modern public transport is an integral part of Smart Cities. Every day, millions of travellers use information about the current state of transport systems to plan their journeys. As the number of smart devices in these systems grows, so does the ability to provide more detailed information. These more detailed types of data also include geographically accurate routes of lines, and typically transport companies or coordinating authorities reach for solutions from the routing domain to obtain this type of data. This also includes the Integrated Transport System of the South Moravian Region (IDS JMK), where the need to complement the information provided to passengers with the geographically precise line routes has become apparent. However, the suitable data, which allows direct use of routing in this transport system, including the second largest city in the Czech Republic, did not exist. Thus, the purpose of this paper was to verify whether it is possible, with the help of preprocessing and subsequent data augmentation, to obtain geographically accurate line routes even from data sources that were not originally intended for this purpose. As a result, a state machine has been designed which, with the selected and implemented routing algorithm, allows to determine the exact structure of individual lines. Obtaining the exact structure of individual lines then enables the generation of geographically accurate routes, which was tested within the implementation of the proposed solution in cooperation with the coordinating authority for the IDS JMK, the Kordis company.**

***Index Terms*—Data augmentation, Data handling, Data processing, Public transportation, Vehicle routing**

I. INTRODUCTION

Routing as a concept covers a whole collection of problems and their solutions. Most often, these solutions consist in transforming a particular problem into an oriented graph [1]. With different variations and specifics, the individual solutions are then transferred between different domains from computer networking [2], to circuit boards [3], to logistics [4]. In addition to the most well-known areas where routing is used, there are also more specific fields of application, such as finding the most optimal route in different traffic networks.

The problem of finding the most optimal route in given conditions has been studied before [5]. But this problem came to attention only with one of the most important, and studied, combinatorial optimization problems—the vehicle routing problem [6]. Since this problem was defined in 1959 [6], the search for the most optimal route has been gradually decoupled and now it is a separate problem within the routing domain. While in the early days finding the most optimal route was solved using shortest path algorithms [7], current solutions incorporate a variety of optimization techniques [8]. In addition, current implementations of the algorithms also have to deal with dynamic parameters [9] such as weather or traffic congestion [10]. These optimizations make it possible to cover an ever-increasing number of routing requests in increasingly complex transport networks.

A similar problem in the routing domain is the area of public transport. In this area, routing is primarily used to solve two types of problems. The first type is the problem of optimising public transport systems, where routing is used in the design of the routes (Transit Network Design Problem-TNDP) [11]. As with routing in traffic networks, the individual algorithms must deal with external requirements for the optimal route, such as the exact order of the stops through which the route must pass [12]. The second type represents the planning of the actual connections within the transport system by passengers. In contrast to classical routing, time plays an important role here [7], since the nature of public transport systems is stochastic. Therefore, route searching engines use heuristics other than just the shortest distance [13] between the start and end points. In addition to these two main domains, routing in public transport is also used in specific cases, such as providing more accurate line route data in the form of geographically precise route, as shown on Figure 1. However, this subset of data belonging to ATIS (Advanced Traveller Information Systems) is only addressed occasionally.

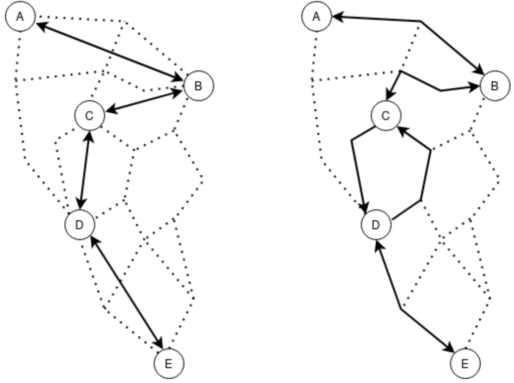


Fig. 1. While the simplified route data on the left is based on simply linking a specified sequence of stops, the geographically precise routing on the right is based on the underlying traffic net. In conjunction with traffic regulations, the out-and-back route may not always pass through the same nodes as in the case of the simplified routing.

In order to obtain an accurate routing of the lines, it is necessary to know the exact order of the stops in order to construct the routing task itself. However, since the input data coming from the internal ATIS system did not contain all the necessary metadata to determine the exact stop order, it was not possible to use the routing directly. Similarly, the complexity of the system itself proved to be a problem, with only rarely one correct stop order existing within the lines.

---

[1] *BLIND REVIEW*

Our solution, therefore, was to design a state machine with a built-in routing algorithm that simultaneously decodes the order of stops within a line while constructing all variants of its geographically accurate route.

The verification of the proposed concept then consisted in the implementation of the designed state machine and routing algorithm, while the whole concept was implemented as a complex geographic information system. As a result, the proposed solution was able to operate with an average accuracy of 85%, while it was able to successfully handle the wide class of anomalies. The result confirms the suitability of using routing in the problem of data augmentation provided within ATIS. Thus, the main benefit lies mainly in increasing the satisfaction of the travelling public.

II. ROUTING IN TERMS OF PUBLIC SERVICE

In the broad area that routing addresses, routing paths in transportation networks is one of people's most commonly used applications. In particular, path routing algorithms have to deal with two conflicting requirements, where on the one hand is the requirement for the most optimal route and on the other hand is the requirement for the shortest computation time. With the gradual growth in the size, the rising complexity of transportation networks and the increasing number of requirements, traditional algorithms have gradually become unusable [8]. However, thanks to various optimizations, today's algorithms are capable of handling single-route calculations in milliseconds.

One of the oldest and simplest basic algorithms that have been used in individual car path routing is the Dijkstra’s algorithm [14], which is essentially a blind search of the state space. The improvement in the efficiency of this algorithm lies in bidirectional search, where the number of nodes visited decreases significantly. Other basic techniques include goal-oriented algorithms, such as A* algorithm [7]. Their essence is the use of certain heuristics to shrink the state space towards the desired goal. Although the use of heuristics such as the geographic distance to the next stop does not yield satisfactory results in the case of the A* algorithm [7], some optimizations based on this algorithm such as precomputation of distances between points [15] still make this approach competitive.

While the basic approach works with a complete state space, the second group of algorithms for routing works with the concept of partitioning the state space into multiple parts. This concept consists in a partition into cells, where the shortest distances within each cell are precomputed and then shortcut arcs [16] are introduced between the cells. This makes it possible to speed up some parts of the routing algorithms [7]. Hierarchical algorithms are also based on partitioning, although not of the state space, to distinguish transport networks according to the importance of individual nodes [17]. Then, the algorithm first searches a smaller set of more important nodes, such as highways, in search of a suitable route, and then traces the exact route around the destination at less important nodes [7]. Modifications of this approach are then algorithms that work with virtual nodes with precomputed distances and distance tables [18]. In addition to these approaches, there are also algorithms that combine different techniques to further increase the speed of the routing itself. Besides, there are extensions whose goal may not be to obtain the shortest path, but, for example, to obtain multiple alternative routes [7]. The implementation of these algorithms then finds applications not only in the field of public transport.

While the search for the most suitable route within individual transport options is mainly limited by physical infrastructure, the search for the most suitable route within public transport systems is also limited by time. This is due to the fact that in public transport systems, certain sections of the network are only available at certain times, in other words, individual stop connections made at precise times [7]. Thus, the search for the most optimal route does not consist in routing in physical space, but in the space formed by the abstract structure of public transport systems. This fact is particularly reflected in the way routing is handled in public transport.

The basis of the solution is the modelling of public transport systems into graphs with a more suitable structure. One possibility is the modelling using the temporal sequence of events in these systems. The individual events, departures of a particular trip from a particular stop, are then interconnected in the graph in the direction of the flow of time [19]. An important element is then the representation of transfers between connections, which is solved by linking individual nodes. Another possibility is to model the system in a graph in which the individual vertices represent the stops and the connections between them the individual links [20]. The exact departure times are then encoded in travel time functions. Speeding up the search then logically led to an attempt to use the same algorithms as in the case of individual transport. This effort has resulted in the use of a variety of methods ranging from A* [21] to precompute shortest paths over subsets of stops and connections [22]. However, in addition to finding the most convenient connection, routing in public transport can also be applied to geographic precise line routes.

The geographically precise route of the current connection or the whole line is additional information provided within ATIS systems, which is well reflected by the fact that this data is an optional part of the most widely used public transport data sharing format, GTFS. Research [23] that addresses this issue then finds application not only in the generation of geographically accurate routes [24], but also, for example, in the automated generation of entire schemes of public transport systems [25]. However, the important fact remains that the result is always the most likely option for a given route [24]. The individual solutions then use different meta-information, such as the one-way streets or the orientation of the vehicles, to improve the accuracy. An important element is also the input network base maps, which are most often derived from open data sources such as OpenStreetMaps. It is the results of these works that have laid the basis for solving this kind of problem within the IDS JMK system.

III. PROBLEM DEFINITION

The integrated transport system of the South Moravian Region covers the second largest city and one of the largest metropolitan areas in the Czech Republic. This system consists of more than 300 lines of road and rail transport. Obtaining geographically accurate routes was thus a non-trivial task that involved solving several related problems.

Since the first base input of routing algorithms is a well-defined state space, in the case of public transport the availability of good quality up-to-date traffic maps is essential. However the problem with normally available traffic maps is that there is a delay between changes to the physical infrastructure and the updating of the traffic maps. This problem is most evident in the case of new industrial and residential developments where the area is served by public transport but the available transport maps still do not cover the area. In addition, in some cases the public transport infrastructure requires a certain degree of detail compared to the normal map.

The second base input to routing algorithms is the determination of the start and end nodes. In the case of public transport, the input is the exact logical order of the stops through which the line passes. The resulting route is then formed by the outputs of the routing tasks run for each pair of stops. However, the input data format does not contain all the metadata needed to determine the correct stop order. This fact made it impossible to use any routing algorithm directly. In addition, analysis of the data structure itself showed that there is only exceptionally a single stop order within each line. Thus, in addition to the root route, there are also different variations in the form of, for example, multiple final stops or parallel routes. An example of the comparison between the input format and the actual system structure is shown on Figure 2.

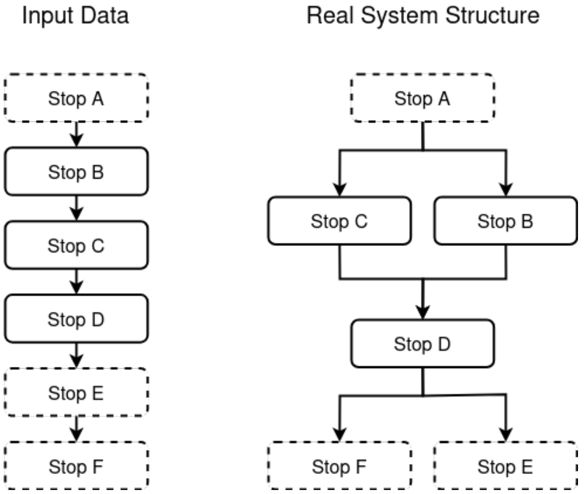


Fig. 2. A simplified example of the input data format converted into a schematic representation of the actual route. The input data contains only information about the terminal stops, here represented by dotted boxes. In order to obtain the real structure of the route of the line, it is necessary to calculate the relationships between the stops in real space.

The third factor that affects routing tasks is the algorithm itself for finding the connection of the start and end nodes in the state space. Due to the characteristics of the input data, the use of the existing solution appeared to be unsatisfactory. Therefore, it was necessary to select and implement a suitable routing algorithm. In the selection process it was necessary to take into account the difficulty of algorithms implementation as well as its reliability. On the other hand, the less stringent computational speed requirements as well as the smaller scope of the intended transport networks simplified this choice.

The nature of these three problems required the development of a new solution, which is based on the well-studied field of routing.

IV. PROPOSED SOLUTION

The aim of our solution was to design a complex tool that would be able to automate the process of creating geographically accurate IDS JMK lines as much as possible. In addition to the routing itself, solving this problem required the possibility of specific modifications of the road and rail network itself. The result of our work is thus a complex geographic information system consisting of two interconnected tools.

The first tool is a map editor, which allows editing of the underlying traffic maps both in terms of infrastructure and traffic constraints. The input is pre-processed traffic maps of the South Moravian Region, cleaned of redundant data such as railway sidings or country roads. By separating the map data itself from open data sources, it is possible both to incorporate real-time changes to the transport infrastructure into the maps and to create more detailed network mapping, as in the case of bus stations. This results in specifically adapted transport maps, which serve as the basis for the second tool.

The second tool is used for the actual routing of lines over the map bases. In addition to parsing the input data about the line, it contains two main parts. The first part includes the partially optimised routing algorithm itself, which computes routes for pairs of stops. The second part is a state machine that both decodes the order of the stops and executes the routing tasks. Using the input data and the results of the routing tasks, the state machine then detects different route variations and generates complex routes for each line as its output. An overall view of the routing engine diagram is captured on Figure 3.

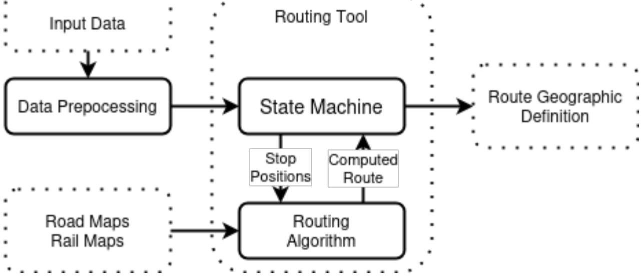


Fig. 3. The overall scheme of the routing tool shows that the state machine calls the routing algorithm for different pairs of stops. The resulting route is then generated from the evaluation of the input data and the results of the routing tasks.

The routing algorithm always computes the shortest route between two stops, which is due to the nature of transport systems where the shortest route is most often the correct option to connect two stops. At its core is the algorithm A*, using the estimated geographical distance between the starting stop, the current point in the network, and the ending stop as a heuristic. The state space search itself is then bounded by two constraints. The first is the traffic regulations resulting from map bases such as one-way streets, i.e. one-way links in the graph. The second are infrastructure constraints where, particularly in the case of rail traffic, it is not possible to continue in certain directions within certain junctions as this would mean an unrealistic wrapping of the resulting route. Since the use of pure heuristics would lead to high temporal and spatial complexity, 2 penalization mechanisms are implemented within the algorithm.

The first mechanism consists in adjusting the ranking of each option based on the evolution of the heuristic function value. While the evaluation improves as the value decreases and hence as the target is approached, the overall evaluation of the option gets worse as the value of the heuristic function stagnates or increases. As the result, options whose evaluation continuously becomes worse after a certain number of iterations are then eliminated. This mechanism can overcome complex shapes or workarounds in traffic networks so in the result, the shortest route is returned while the searched state space is reduced as much as possible. An example of the use of the first penalty mechanism is shown in Figure 4.

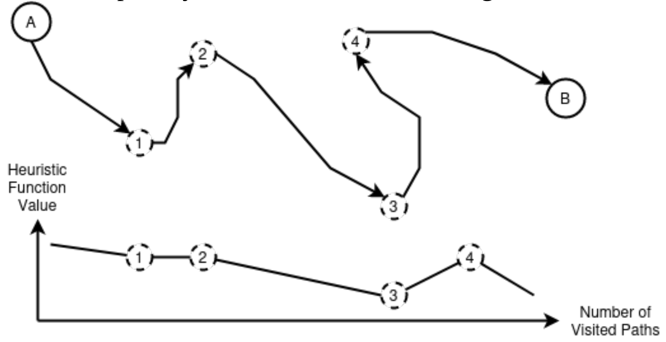


Fig. 4. A penalty mechanism based on the evolution of the value of the heuristic function allows to reduce non-perspective options. The value of the heuristic function is continuously decreasing, the investigated option is approaching the target. The problem occurs only after the 3rd measured point, when the option becomes unpromising. However, due to the previous evolution, the option is not reduced and eventually reaches the goal of the routing task at point B.

The second mechanism consists in tracking the individual nodes visited and the total length of each option. In this case, the ranking of an option is degraded if another option passing through a given node has a shorter actual geographic length from the initial stop than the option under consideration. These mechanisms significantly affect the resulting routes, which the algorithm returns to the second main part of the routing tool, the state machine.

The reduced speed requirements of the routing algorithm made it possible to use a relatively simple routing algorithm. However, as the problem definition implies, it was not possible to determine from the input data the exact order of the stops underlying the routing tasks. In addition, individual lines contain different variations of stop order. As a solution to this problem, we proposed a state machine. This state machine basically always works within a single line and direction with a single root route and its variants. The data analysis shows that the route variants can be divided into three groups:

1) The first group is the case with multiple end stops, where the line does not terminate at a single point, with the split occurring at a previously unspecified point along the route.
2) The second case is the occurrence of a terminal stop in the middle of the route, where in some cases it is part of the root route, while in other cases it lies outside the root route together with a previously unknown number of stops that are also not part of the root route.
3) The third case is then the splitting of the route into two parallel sections with their own stops, where both the beginning and the end of this splitting is unknown. In addition, the possibility of multiple platforms on individual stops is also included in the route design.

As Figure 5 shows, realistic routes mostly consist of a succession of these variants.

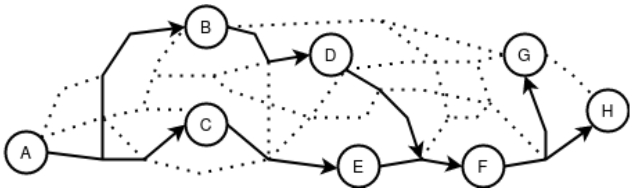


Fig. 5. Real routes consist of different sequences of the three groups described. In this case it is a variant from group number 3 followed by a variant from group number 1.

The proposed state machine uses a routing algorithm and simultaneously constructs the root route and its variants. Our basic assumption was that the probability of a line route bending 180 degrees at a single location is close to zero. The occurrence of these breaks then serves as a cue for finding route variants. Since the proposed state machine itself uses the routing algorithm for two stops each time, it can detect these breaks by comparing the results of the algorithm between pairs of stops and looking for cases where the route from a stop continues in the same direction it came from at the previous pair of stops. In this case, the state machine detects the start of the route variant and tries to calculate the route variant. In the same way, the machine detects the end of the variant and its termination, merging with the root route. As a result, there are different orders of states for different cases, with the goal of obtaining the complex root route and all its variants.

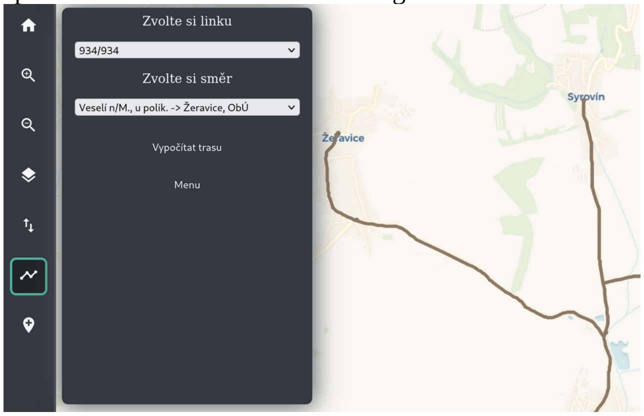
A well-studied routing area simplifies the acquisition of geographically accurate line routes. On the other hand, in order to be able to use routing, the exact order of stops through which the line passes must be known in advance, which in the case of input data describing the IDS JMK presents a specific problem. Our proposed solution exploits the capabilities of the routing domain and, together with the proposed state machine for determining the correct order of stops, thus enables the retrieval of geographically accurate routes of individual lines in their full form.

V. RESULTS AND EVALUATION

Since the design of our solution was based on the requirements of the coordinating authority of the IDS JMK, the Kordis company, the testing of the proposed and implemented solution was carried out in close cooperation with this company. The testing itself then focuses on two main areas. The first area was reliability and the ability of proposed heuristic to deal with various deviations in decoding the stop sequence into real line structure. The second area involved monitoring and evaluating the accuracy of the constructed routes by comparing them with real line routes.

The structure of the studied integrated transport system of the South Moravian Region consists of more than 4200 stops and 340 lines. This includes various types of transport ranging from trams, passenger trains to express bus lines, with a total of more than 650 routing tasks on average per system state. In addition to the underlying maps, the input data consists of internal Kordis datasets that contain the locations of the stops as well as the individual lines.

The implementation itself consists of two separate tools, an API and a database, which together form a single geographic information system. For the database implementation, the PostGIS framework was used, while the application interface uses NodeJS technology with the Express framework. Both tools are then implemented using the Angular library. The source code as well as the documentation for the whole system can be found in the public repository[2]. An example of the implemented GIS[3] is then shown in Figure 6.



934/934

Veselí n/M., u polik. -> Žeravice, ObÚ

Fig. 6. An example from the implemented routing tool shows how the final geographic definition of the route of a particular line for one direction looks like. In this example, one root route and its two variations are visible.

The first area of testing involved the reliability of the proposed heuristic. The input was the set of all IDS JMK lines, which represented 680 separate routing tasks. The ratio between correct and incorrectly completed tasks was then investigated. A correctly completed task is considered to be where the state machine creates a route structure that contains all stops, with each stop having at least one neighbour. The results then showed that, on average, 96% of the routing tasks were successfully completed, resulting in a geographic definition of the line route. In particular, the lines crossing the municipal area of the South Moravian Region were found to be problematic, where it was necessary to supplement the input transport maps. Problems were then caused to a lesser extent by marginal cases with duplicate platforms or a high number of alternative routes for one stop.

While the first area of testing consisted of verifying the functionality of the proposed heuristic, the second area of testing focused on its accuracy. The results of the routing tasks from the previous part of the testing were used as input. The first step was to determine the significant points through which each line passes, based on the integration of data from different sources, such as the IDS JMK traffic monitoring application. The order in which the line passes through these points was then determined. The second step was the evaluation of the accuracy of individual routes. For each route, the ratio between the points visited in the correct order and the points that were not visited or were visited in the wrong order was calculated. As a result, the proposed heuristic was able to generate outputs that matched the real routes on average 88%. Bus routes in particular proved to be problematic, where the proposed heuristic failed to address a small class of specific stop and platform combinations in the input data. An overall summary of the results of both parts of the testing is then shown in Table 1.

Type of transport	**Routing reliability**	**Route precision**
Rail	98%	95%
Buses	93%	80%
Trolleybuses	97%	87%
Trams	94%	89%

TABLE I

OVERALL SUMMARY OF THE RESULTS

Two findings in particular emerge from the overall testing results. The first highlights the fact that it is more difficult to find the correct sequence of stops than the actual connections between these stops in transport networks. The second insight points to the fact that the proposed solution is less successful in the road transport domain. This is due to the fact that road networks are more complicated, in addition to the fact that regional bus routes generally have a more complex structure, richer in different variations of individual routes.

VI. DISCUSSION

The result of our work is a tool enabling automated generation of geographically accurate routing of IDS JMK lines. We have verified our proposed solution by its implementation and subsequent testing over a selected subset of lines. The result is a complex state machine with a built-in routing algorithm that can generate geographically precise line routes with an average accuracy of 85%. In addition to enhancing the information provided to passengers, the solution thus proposed shows one way in which public transport routing can take advantage of more widely available data formats, not

---

[2] Blind review

[3] Blind review

originally designed for this purpose. On the other hand, there are still bottlenecks in the proposed solution, such as duplication of platforms within stops, which the tool is not yet able to fully deal with.

Further development of the proposed solution should be directed mainly in two areas. On the one hand, there is a need for computational optimizations, for example, in the form of selection and implementation of a more suitable routing algorithm, as well as the introduction of indexing. The second area of development could then be automated testing of the results of routing tasks, which could be based on the analysis of historical data on the location of public transport vehicles. These improvements could lead to a greater degree of automation and ultimately to the redirection of valuable human resources to more important tasks.

VII. CONCLUSION

The purpose of this work was to verify the possibility of extracting geographically accurate routes from a compressed format that did not contain all the metadata necessary for this task. Our solution consisted in designing and implementing a state machine, which uses a routing algorithm that simultaneously detects the root route of a line and its variants, while as a result is generating a geographically accurate representation of the whole line and its variants. We validated the proposed solution within the implemented tool, which was tested in collaboration with the Kordis company. As a result, the proposed solution shows a way to solve routing problems in the public transport domain when direct use of routing algorithms is not possible. The proposed tool then enables the automated acquisition of previously unavailable data, thus contributing to the enhancement of the information provided to the travelling public, and therefore contributing to the increase of the travelling public's satisfaction.

ACKNOWLEDGMENT

Hide due to the Blind Review.

REFERENCES

[2] D. Schultes, “Routing in Road Networks with Transit Nodes,” Encyclopedia of Algorithms, pp. 1875–1879, 2016, doi: 10.1007/978-1-4939-2864-4_353.

[3] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg, “Compact distributed data structures for adaptive routing,” Proceedings of the twenty-first annual ACM symposium on Theory of computing - STOC ’89, 1989, doi: 10.1145/73007.73053.

[4] J. Fawcett and P. Robinson, “Adaptive routing for road traffic,” IEEE Computer Graphics and Applications, vol. 20, no. 3, pp. 46–53, 2000, doi: 10.1109/38.844372.

[5] FISHER, Marshall. Vehicle routing. *Handbooks in operations research and management science*, 1995, 8: 1-33.

[6] L. Rosenfeld, “Unusual problems and their solutions by digital computer techniques,” Papers presented at the February 7-9, 1956, joint ACM-AIEE-IRE western computer conference on - AIEE-IRE ’56 (Western), 1956, doi: 10.1145/1455410.1455441.

[7] P. Toth and D. Vigo, Eds., “The Vehicle Routing Problem,” Jan. 2002, doi: 10.1137/1.9780898718515.

[8] D. Delling, “Route planning in transportation networks,” Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Nov. 2018, doi: 10.1145/3274895.3282802.

[9] Q. Liu, P. Hou, G. Wang, T. Peng, and S. Zhang, “Intelligent route planning on large road networks with efficiency and privacy,” Journal of Parallel and Distributed Computing, vol. 133, pp. 93–106, Nov. 2019, doi: 10.1016/j.jpdc.2019.06.012.

[10] A. Sadeghi-Niaraki, M. Varshosaz, K. Kim, and J. J. Jung, “Real world representation of a road network for route planning in GIS,” Expert Systems with Applications, vol. 38, no. 10, pp. 11999–12008, Sep. 2011, doi: 10.1016/j.eswa.2010.12.123.

[11] Vi Tran Ngoc Nha, S. Djahel, and J. Murphy, “A comparative study of vehicles’ routing algorithms for route planning in smart cities,” 2012 First International Workshop on Vehicular Traffic Management for Smart Cities (VTM), Nov. 2012, doi: 10.1109/vtm.2012.6398701.

[12] P. Heyken Soares, C. L. Mumford, K. Amponsah, and Y. Mao, “An adaptive scaled network for public transport route optimisation,” Public Transport, vol. 11, no. 2, pp. 379–412, Jul. 2019, doi: 10.1007/s12469-019-00208-x.

[13] P.-L. Bourbonnais, C. Morency, M. Trépanier, and É. Martel-Poliquin, “Transit network design using a genetic algorithm with integrated road network and disaggregated O–D demand data,” Transportation, vol. 48, no. 1, pp. 95–130, Aug. 2019, doi: 10.1007/s11116-019-10047-1.

[14] A. Bozyigit, G. Alankus, and E. Nasiboglu, “Public transport route planning: Modified dijkstra’s algorithm,” 2017 International Conference on Computer Science and Engineering (UBMK), Oct. 2017, doi: 10.1109/ubmk.2017.8093444.

[15] E. W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” Edsger Wybe Dijkstra, pp. 287–290, Jul. 2022, doi: 10.1145/3544585.3544600.

[16] A. Botea, “Ultra-Fast Optimal Pathfinding without Runtime Search,” Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 7, no. 1, pp. 122–127, Oct. 2011, doi: 10.1609/aiide.v7i1.12443.

[17] D. Delling, A. V. Goldberg, T. Pajor, and R. F. Werneck, “Customizable Route Planning in Road Networks,” Transportation Science, vol. 51, no. 2, pp. 566–591, May 2017, doi: 10.1287/trsc.2014.0579.

[18] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter, “Exact Routing in Large Road Networks Using Contraction Hierarchies,” Transportation Science, vol. 46, no. 3, pp. 388–404, Aug. 2012, doi: 10.1287/trsc.1110.0401.

[19] T. Akiba, Y. Iwata, K. Kawarabayashi, and Y. Kawata, “Fast Shortest-path Distance Queries on Road Networks by Pruned Highway Labeling,” 2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 147–154, Dec. 2013, doi: 10.1137/1.9781611973198.14.

[20] F. Schulz, D. Wagner, and K. Weihe, “Dijkstra’s Algorithm On-Line: An Empirical Case Study from Public Railroad Transport,” Lecture Notes in Computer Science, pp. 110–123, 1999, doi: 10.1007/3-540-48318-7_11.

[21] G. Stølting Brodal and R. Jacob, “Time-dependent Networks as Models to Achieve Fast Exact Time-table Queries,” Electronic Notes in Theoretical Computer Science, vol. 92, pp. 3–15, Feb. 2004, doi: 10.1016/j.entcs.2003.12.019.

[22] Y. Disser, M. Müller–Hannemann, and M. Schnee, “Multi-criteria Shortest Paths in Time-Dependent Train Networks,” Lecture Notes in Computer Science, pp. 347–361, doi: 10.1007/978-3-540-68552-4_26.

[23] H. Bast, S. Funke, and D. Matijevic, “Ultrafast shortest-path queries via transit nodes,” The Shortest Path Problem, pp. 175–192, Jul. 2009, doi: 10.1090/dimacs/074/07.

[24] R. Geisberger and C. Vetter, “Efficient Routing in Road Networks with Turn Costs,” Lecture Notes in Computer Science, pp. 100–111, 2011, doi: 10.1007/978-3-642-20662-7_9.

[25] H. Bast, P. Brosi, and S. Storandt, “Efficient generation of geographically accurate transit maps,” Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Nov. 2018, doi: 10.1145/3274895.3274955.

[26] H. Bast and P. Brosi, “Sparse map-matching in public transit networks with turn restrictions,” Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Nov. 2018, doi: 10.1145/3274895.3274957.