

Analysis Urban Traffic Vehicle Routing Based on Dijkstra Algorithm Optimization



Truong-Giang Ngo, Thi-Kien Dao, Jothiswaran Thandapani, Trong-The Nguyen, Duc-Tinh Pham, and Van-Dinh Vu

Abstract The transportation cost, running vehicle time, duration, and distance cost, the route network of revealed urban vehicles, are considered to need to be analyzed meaningfully and reasonably planned. This paper suggests an analysis of urban traffic vehicle routing based on the Dijkstra algorithm optimization that is as a solution to the road selection and optimization under various constraints jointly built in the central metropolitan area. The leading metropolitan area road network's various constraints are to consider the road condition factors and the risk resistance in road planning. The analysis and realization are verified with the geocoding, network topology, and network analysis. The results show that the integration of network analysis technology and the Dijkstra algorithm realizes the urban vehicle route's optimization decision. Still, the improved Dijkstra algorithm reduces the number of node visiting and time complexity. Under the driving time and distance constraints, the speed limit, road hierarchy, and road condition are the line selection's restrictive factors. The graphical description could provide technical support and reference for the driver's driving strip and traffic management department for decision making.

Keywords Urban traffic · Dijkstra algorithm · Constraint conditions · Path analysis

T.-G. Ngo

Faculty of Computer Science and Engineering, Thuyloi University, Hanoi, Vietnam

T.-K. Dao · J. Thandapani · T.-T. Nguyen (✉)

Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, China

J. Thandapani

Anna University, Chennai, India

T.-T. Nguyen

Haiphong University of Management and Technology, Haiphong, Vietnam

D.-T. Pham

Center of Information Technology, Hanoi University of Industry, Hanoi, Vietnam

Graduate University of Science and Technology, Vietnam Academy of Science and Technology, Hanoi, Vietnam

V.-D. Vu

Faculty of Information Technology, Electric Power University, Hanoi, Vietnam

1 Introduction

Path observation is one of the research materials of an overview of urban traffic networks [1] and a part of the spatial analysis method's spatial analysis method [2]. It is commonly used in urban road maintenance, network coordination [3], traffic planning, and others [4]. The center of route analysis is the shortest path and the safest path, and the path optimization is to use the concept of tactical research planning to maximize the organizational efficiency and usage and road network, vehicle routing, delivery of goods, etc. [5]

In the case of confidence, there have been several studies on the shortest path analysis algorithm, e.g., Bellman algorithm [6], Dijkstra algorithm [7], and Dreyfus algorithm [8] have been the traditional algorithms. Nonetheless, under complexity, the shortest path problem can be loosely separated into four parts. This paper looks at the most straightforward path of random road length change, the shortest route with different cost functions, the shortest path with random road length change under the condition of time independence, and the shortest path of interval length.

Nevertheless, in the actual driving of urban vehicles, the above shortest path studies do not consider unknown considerations of the other road route conditions, e.g., rush hours, limited speed, or road situation [9]. Therefore, this study does more work focused on optimization allocation based on the Dijkstra algorithm. First, the urban road network is transformed into a driven graph, including nodes and connections, using the geographic information network mapping technology. The municipal road is transformed into a directed graph. Second, we consider several factors, e.g., the type of route, node stay time, speed limit, the direction of driving of vehicles, and other factors. A starting point to the endpoint is measured and evaluated with the time and cost to optimize global results. It means that the time or expense from the starting point to the parameter is calculated according to the time cost and distance.

Besides the expense restriction considerations and the calculated optimum paths, the shortest route scenarios under unconstrained conditions [10], hazard road condition, and speed limit mode are also considered for testing on the spot.

2 Related Work

2.1 Dijkstra Algorithm

Dijkstra algorithm is a standard algorithm for finding the shortest path in graph theory proposed by Dijkstra, a Dutch computer scientist, in 1959 [11]. The algorithm can find the shortest path from a point in a graph to any other vertex. Dijkstra algorithm divides network nodes into three parts: unlabeled in the network graph, all nodes are initialized as unlabeled nodes. In the process of searching, the nodes that are connected in any path and travel nodes are temporarily marked nodes. In each cycle, the shortest path length from the origin point is searched from the temporarily marked

nodes as the permanently marked nodes until the nodes are found. As shown in Fig. 1, assuming that the origin point of the node network is node 0 and the target point is node 4, the shortest path distance between node 0 and node 4 is calculated (the length between nodes is assumed).

Suppose each node has a pair of labels (w_i, p_j) , w_i is the length of the shortest path from the origin point to the target point, w_j is the shortest path length from the origin point s to any node j (the shortest path from the vertex to itself is the zero path (the path without arc), and its length is equal to zero), and p_j is the length from the origin point s to the node j . The shortest path from the origin point I to the target node j is solved. The basic process is as follows:

- (1) Initialization. The origin point is set to
 - (i) If the shortest path length $w_s = 0$, p_s is null;
 - (ii) The path length of all other nodes is $w_i = \infty$, $p_i = s$;
 - (iii) Mark the origin point S . mark all marked nodes $k = s$, and set other nodes as unmarked.
- (2) Verify the distance from all marked nodes K to their directly connected unlabeled nodes J , and set

$$w_j = \min\{w_j, w_k + d_{kj}\} \tag{1}$$

where w_j is the shortest path length of unlabeled node j , w_k is the shortest path length of labeled node k , and d_{kj} is the direct connection distance from node k to node j .

- (3) Select the next point. From all unlabeled nodes w_j , select the smallest labeled node i , that is, $w_i = \min w_j$ (all unlabeled nodes j), and node i is selected as the point in the shortest path and set as the marked point

Fig. 1 Meshed network of the route

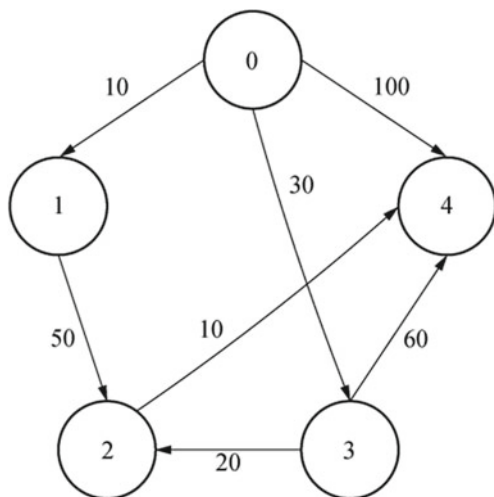


Table 1 List of an example of the shortest path of node 0 to node 4 processing

Cycle	Node set s	Number of marked nodes K	Distance d between node 0 and 1 $D\{0\}$	Distance d between node 0 and 1 $D\{1\}$	Distance d between node 0 and 2 $D\{2\}$	Distance d between node 0 and 3 $D\{3\}$	Distance d between node 0 and 4 $D\{4\}$
Initialization	{0}	–	0	10	∞	30	100
1	{0,1}	1	0	10	60	30	100
2	{0,1,3}	3	0	10	50	30	90
3	{0,1,3,2}	2	0	10	50	30	60
4	{0,1,3,2,4}	4	0	10	50	30	60

- (4) Find the previous point of node i . find the point j' directly connected to node i from the marked node. As the previous point, set $i = j'$
- (5) If all nodes have been marked, the algorithm will find the shortest path; otherwise, mark $k = i$ and go to step (2). Table 1 shows a list of an example of the shortest path of node 0 to node 4 processing.

2.2 Algorithm Optimization

The optimization is implemented based on the Dijkstra algorithm as follows: firstly, the temporary set NB_s of the starting point, the neighbor nodes k with the smallest distance is selected as the transition point, and at the same time, it is assigned to the identification set S . The union of the temporary set of all nodes in the identification set s and the difference set of the identification set $(\cup NB_s S, i \in S)$, a node W_k with the minimum path distance value is selected as the next transition point and is assigned to the identification set S . Repeat the above process until all nodes are identified $|s| = n$, and the algorithm ends. Let NB_i be the temporary set of nodes i , S be the identification set, w_i be the shortest path length from source point s to node j , P_i be the previous node of j point in the shortest path from s to j , and D_{ij} is the distance from node i to node j .

- (1) Initialize identity collection $S = \{s\}$, $w_i = d_{si}$ $i \in NB_s$ otherwise $w_i = \infty$ $i \in NB_s$; $P_i = s$.
- (2) If the distance between k temporary set node and origin point is the smallest $d_{sk} = \min d_{si}$, then $S = S \cup \{k\}$, $j \in NB_s$
- (3) Modify the w_i value in k temporary set node $NB_k = S$, $w_i = \min\{w_i + d_{ki}\}$; if w_i value changes, then $P_j = k$, $j \in NB_s - S$
- (4) Select the minimum value of w_i in the marked temporary node, set a and classify it into s , $w_k = \min w_i$, $S = S \cup \{k\}$ if $|s| = n$, the node has been identified, and then the algorithm is terminated; otherwise, go to step (3).

For the network graph with n nodes, the Dijkstra algorithm needs a total of $n - 1$ iterations to solve the shortest path, and a new node is added to the temporary node set in each iteration. Because the number of nodes not in the temporary node set is $n - i$ in the i th iteration, then $n - i$ is needed in the i th iteration. The number of nodes processed is

$$\sum_{i=1}^{n-1} (n - i) = \frac{n(n - 1)}{2} \quad (2)$$

When the number of network nodes in Eq. (2) is n , the time complexity of the path solution is $O(n^2)$, which is the time complexity of the shortest path from the origin point to the other nodes in the network graph. With the increase of the total number of nodes n , the calculation efficiency and storage efficiency decrease.

The time complexity $O(n^2)$ depends on the number of elements in the temporary set NB_s of the transition point k , and then the space complexity of the optimization algorithm is $O(n \times P)$, where p is the space occupied by the node object under the adjacency matrix storage structure $N \times N$, which is a constant. According to the number of points and road sections, the average out degree e of nodes can be obtained, i.e.,

$$e = \frac{m}{n} \quad (3)$$

where m is the number of road sections.

Generally, in the geographic information system network diagram, $e \in [1, 4]$, because step (3) and step (4) are search and V_i ($i = 1, 2, 3, \dots, n$). The time complexity of step 4 is $O(m)$, i.e., $O(n \times e)$.

3 Model Construction

We assume the road network of a center city by generating randomly to facilitate the experiment. The vector elements of options are imported into the local geographic database, e.g., the expressway, national road, provincial road, and county road. The topological grade is specified as vector elements of the participating road network. The tolerance limit value, such as the speed limit and traffic restriction and turning model, is set up, and the road arc segment node is established. The cost and constraint variables are set according to the connection rule between the road network node and arc segment. The cost variable is path cost expense, and the attribute field of the constraint variable is a Boolean value, namely a single line or double line, respectively.

$$\begin{cases} R_w = (N, R, L_R) \\ R = \{(x, y) | x, y \in N, \text{ and } L(x, y)\} \\ L_R = \{l_{xy} | (x, y) \in R\} \end{cases} \quad (4)$$

where R_w represents the road network; N represents the node set; R represents the set of road sections, whose elements are ordered pair (x, y) , and indicates that there is a directed path from node x to node y ; L_R represents the weighted length of the section, and its elements $L(x, y)$ denote the weighted length of the directed section (x, y) ; l_{xy} is the length of any section from node x to node y .

The weighted length L_R of a road segment refers to the optimal path which is solved by integrating various dynamic and static attribute constraints according to the multi-objective function and planning requirements, rather than the actual distance or length of the path. Therefore, the optimal path not only refers to the shortest distance in the general geospatial sense. It can also be extended to time, cost, and line capacity.

Some constraints are also considered when nodes or sections in the network cannot operate due to some emergencies, such as traffic accidents, or the original optical path needs to be modified. For example, the road is under maintenance, the distance of the optimal path changes, and maintenance status. If there is a traffic accident at the intersection, it is temporarily not passable; that is, the nodes in the network cannot run, thus, prolonging vehicles' travel time.

The comprehensive road resistance C of any path is calculated by the weighted summation method

$$C = \sum_{i=1}^n C_i = \sum_{i=1}^n (D_{i1}a_1 + D_{i2}a_2 + \dots + D_{ij}a_j) \quad (5)$$

where C_i is the comprehensive road resistance of section i , D_{ij} is the score of the road resistance factor, and a_j is the weight of j road resistance factor. A parameter is used to adjust the weight according to the situation of environmental conditions.

$$a_j^t = \mu \times a_j^{t-1} \quad (6)$$

where μ is an adjusting variable based on the scenario of environmental conditions.

The upper limit measure is used to calculate the D_{ij} a score of any road section, and the calculation formula is

$$D_{ij} = d_{ij} / d_{j,\max} \quad (7)$$

where D_{ij} is the score of j road resistance factor of section i ; d_{ij} is the actual value of road resistance factor; $d_{j,\max}$ is the maximum actual value of J road resistance factor.

Algorithm 1. The adjusted optimal path based on the Dijkstra algorithm

Input: A bi-directed graph with weights $D_{ij} = (N_{ij}; C_{ij})$, the set of all nodes N , and the set of weighted edges of connected nodes C . The source and target node n_s and n_t , respectively.

Output: Shortest path and the length from the source node n_s to the target node n_t

1: Initialization

Visited nodes $Ne = ns.d(ns) = 0$, where $d(nti)$ is the minimum cost of the shortest path from n_s to n_{ti} . $Nu = N - Ne$, where N_u is the set of nodes to n_s with the undetermined shortest path. $d(nti) = \min(C(ns; n_{ti}))$, if n_{ti} is a successor to n_s , or else, $d(n_{ti}) = 1$, where $C(i; j)$ is the cost between node i and node j as Eq. (5)

2: Repeat the following steps until there are no nodes in the set N_u

2.1 Put the node in N_u that has the minimum cost to the old n_s as the new source n_s . Move the new source node n_s to N_e ; adjust the weight according to the situation of environmental conditions as Eq. (6)

2.2 Set $d(n_{ti}) = d(n_s) + \min(C(ns; n_{ti}))$

3: Find the shortest path from the source node n_s to the target node n_t

4: Return $d(nt)$, N_e

The initial work of the Dijkstra algorithm is only dealing with the shortest path between two points. Mathematically, these points must be represented by nodes in the graph network.

Bellman Ford implemented the possibility of fixing

4 Results and Discussion

In order to test the performance of the proposed scheme, some lines assuming roads are generated randomly in a specified metro urban area based on the grid and sample points under constrains of the condition roads and obstacles. Table 2 lists the comparison of theoretical and actual driving time of the optimized route. It is seen that the travel time of the optimized route, the real travel time is longer, and the difference between the minimum and maximum travel time is 1.2 min and 2.9 min, respectively. In the obstacle mode, the driving time of line 4 is 1.0 min longer than that of line 3. The driving time of line 4 is 1.0 min longer than that of line 3. In the speed limit mode, the actual driving time of urban vehicles inline 5 is limited by road congestion, traffic capacity, and traffic information, which is 1.2 min longer than the real driving time on line 6 is 2.3 min longer than the theoretical driving time of line 6. In the case of a 30 s delay at the intersection node, the actual driving distance is more reasonable, and the real driving time is longer. Therefore, the node delay, road conditions, and speed limit are the main factors affecting the urban vehicle route selection. The number of nodes in the optimal path plays a leading role in the driving time and distance.

Figure 2a displays the setting for generating points in the road network. Figure 2b shows the generated obstacle maps and the roads' points as road network grade and speed limit mode, randomly prioritizing provincial roads, national highways, and urban trunk roads and then choosing urban secondary trunk roads and township roads. Figure 2c displays the proposed method's optimal paths close to the actual travel time and distance cost. Drivers would be advised as a priority to these kinds of

Table 2 Obtained results of the optimization of the Dijkstra algorithm for actual driving time and vehicle routing

Serial number	Line selection conditions	Distance (m)	Theoretical driving time (min)	Actual driving time (min)	Travel time difference (min)	Number of nodes
Line 1	The shortest distance in unlimited road class and speed mode	4377.8	10.7	11.9	1.2	19
Line 2	Minimum time in unlimited road class and speed mode	4385.2	9.7	11.5	1.8	17
Line 3	Shortest distance in obstacle mode	4610.3	10.7	12.1	1.4	20
Line 4	Shortest time in obstacle mode	4630.6	10.2	13.1	2.9	18
Line 5	The shortest distance under speed limit mode	4455.2	10.9	13.2	2.3	17
Line 6	The shortest time under speed limit mode	4739.3	10.5	12.0	1.5	21
Line 7	The delay time of 30 s at the node is the shortest	4463.3	15.9	18.4	2.5	17

conditions as assuming that the speed limit of trunk roads is 30 km/h, and of crowded streets is 35 km/h. In the speed limit mode, the shortest path needs to drive 4455.3 m, and the travel time cost is 13.2 min.

Figure 2d shows the shortest distance between Hoabinh park and the Linhnam-Hoang Mai, Hanoi, as an example of the most straightforward under the condition of daily road status and vehicle speed in the middle afternoon. Assuming that Hoa Binh the park, Hanoi is the initial source point and the Linhnam-Hoangmai, Hanoi as the target node, the optimal distance, and time path between Hoa Binh park and

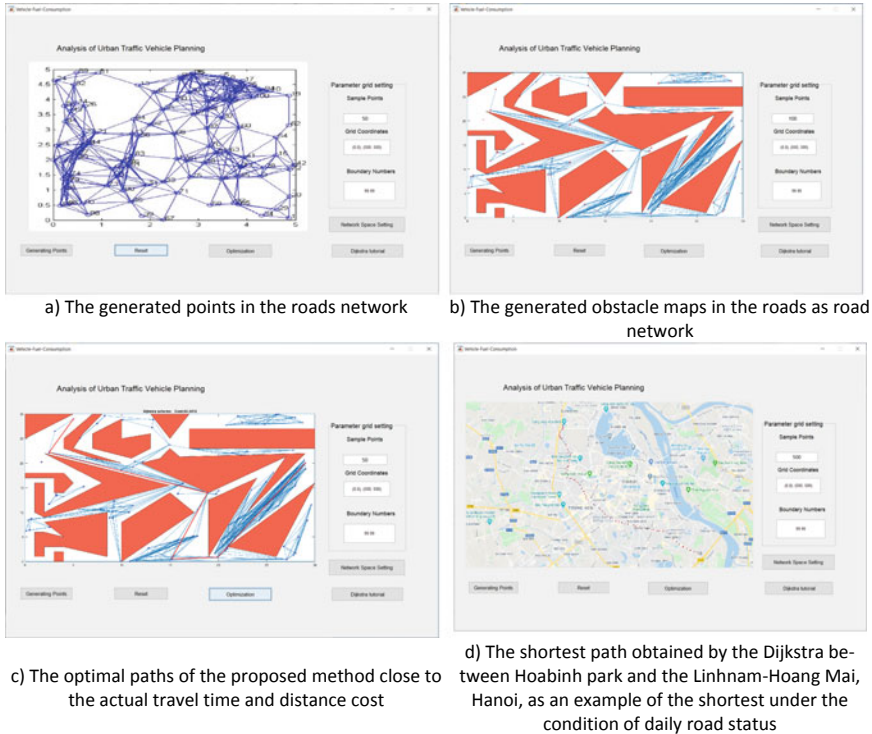


Fig. 2 Proposed scheme’s optical paths based on the Dijkstra for analysis of urban traffic vehicle routing in terms of the travel time and distance cost

the Linhnam–Hoangmai can be calculated at any intersection. The route length is 19,377.8 m, and the time cost is 25.7 min, while the route distance under time constraint mode is 19,374.9 m, and the minimum driving time is 24.7 min. In actual driving, route selection is affected by drivers’ preference for the shortest time and shortest distance. Figure 3 shows a comparison of the suggested approach’s outcomes regarding the average optimization error rate with the A* algorithm and dynamic programming schemes to analyze urban traffic vehicle paths. It can be seen that the suggested approach based on Dijkstra algorithm optimization produces the lowest average converge error rate in comparison with other methods.

Table 3 illustrates the comparison of the proposed scheme’s results with the A-star algorithm and dynamic programming methods. It is seen that the proposed approach produces the results efficiently for the analysis of urban traffic vehicle routing planning.

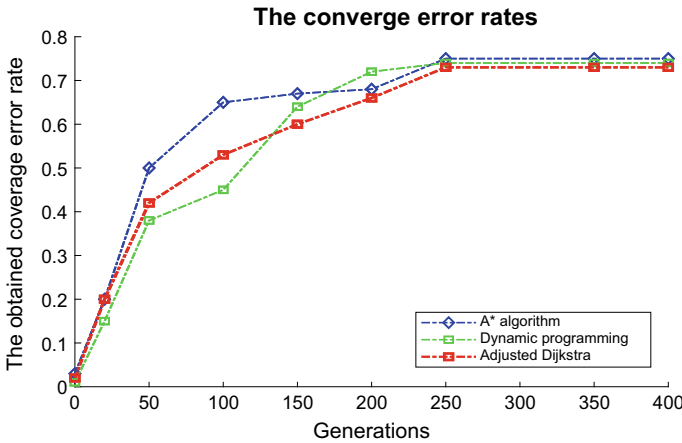


Fig. 3 Comparison of the suggested approach’s outcomes regarding the average optimization error rate with the A* algorithm and dynamic programming schemes to analyze urban traffic vehicle paths

Table 3 Comparison of the results of the proposed scheme with the A-star algorithm and dynamic programming methods

Approach	The obtained best cost	The averaged the optimal cost	The executed time
A* algorithm [12]	19,378	20,263	2670
Dynamic programming algorithm [13]	19,375	20,623	2580
Proposed scheme	19,368	20,233	2376

5 Conclusion

This study proposed an analysis of urban traffic vehicle routing based on the Dijkstra algorithm optimization for suggesting road selection and optimization under various constraints jointly built in the central metropolitan area. The optimal path cost depends on the number of path nodes, link constraints, and speed limit conditions. In this planned scheme of the road network, the specified for both origin point and the target point, the Dijkstra’s obtained optimal path was implemented under distance and time cost constraints. The analysis and realization are verified with the geocoding, network topology, and network analysis tools. The results show that the integration of network analysis technology and the Dijkstra algorithm realizes the urban vehicle route’s optimization decision. Also, the Dijkstra algorithm reduced the number of node visiting and time complexity. Under the driving time and distance

constraints, the speed limit, road hierarchy, and road conditions are the line selection's restrictive factors. The graphical description could provide technical support and reference for the driver's driving strip and traffic management department for decision making.

References

1. Dao T, Nguyen T, Pan J, Qiao Y, Lai Q (2020) Identification failure data for cluster heads aggregation in WSN based on improving classification of SVM. *IEEE Access* 8:61070–61084. <https://doi.org/10.1109/ACCESS.2020.2983219>
2. dan Stucky J (1998) On applying viewshed analysis for determining least-cost paths on digital elevation models. *Int J Geogr Inf Sci* 12:891–905
3. Dao T, Yu J, Nguyen T, Ngo T (2020) A hybrid improved MVO and FNN for identifying collected data failure in cluster heads in WSN. *IEEE Access* 8:124311–124322. <https://doi.org/10.1109/ACCESS.2020.3005247>
4. Bang S, Heo J, Han S, Sohn H-G (2010) Infiltration route analysis using thermal observation devices (TOD) and optimization techniques in a GIS environment. *Sensors* 10:342–360
5. Magzhan K, Jani HM (2013) A review and evaluations of shortest path algorithms. *Int J Sci Technol Res* 2:99–104
6. Awerbuch B, Bar-Noy A, Gopal M (1994) Approximate distributed bellman-ford algorithms. *IEEE Trans Commun* 42:2515–2517
7. Jianya YYG (1999) An efficient implementation of shortest path algorithm based on dijkstra algorithm. *J Wuhan Tech Univ Surv Mapp* 3
8. Fuchs B, Kern W, Wang X (2007) Speeding up the Dreyfus-Wagner algorithm for minimum Steiner trees. *Math Methods Oper Res* 66:117–125
9. Nguyen T-T, Qiao Y, Pan J-S, Chu S-C, Chang K-C, Xue X, Dao T-K (2020) A hybridized parallel bats algorithm for combinatorial problem of traveling salesman. *J Intell Fuzzy Syst* 38:5811–5820. <https://doi.org/10.3233/jifs-179668>
10. Nguyen T-T, Dao T-K, Kao H-Y, Horng M-F, Shieh C-S (2017) Hybrid particle swarm optimization with artificial bee colony optimization for topology control scheme in wireless sensor networks. *J Internet Technol* 18. <https://doi.org/10.6138/JIT.2017.18.4.20150119>
11. Dijkstra EW (1959) A note on two problems in connexion with graphs (PDF). *Numerische Mathematik* 1:269–271. <https://doi.org/10.1007/BF01386390>. S2CID 123284777
12. Duchoň F, Babinec A, Kajan M, Beňo P, Florek M, Fico T, Jurišica L (2014) Path planning with modified a star algorithm for a mobile robot. *Procedia Eng* 96:59–69
13. Psaraftis HN, Tsitsiklis JN (1993) Dynamic shortest paths in acyclic networks with Markovian arc costs. *Oper Res* 41:91–101