Anadolu Üniversitesi Bilim ve Teknoloji Dergisi A- Uygulamalı Bilimler ve Mühendislik
Anadolu University Journal of Science and Technology A- Applied Sciences and Engineering

2016 - Volume: 17 Number: 2
Page: 299 - 313
DOI: 10.18038/btda.06501
Received: 12 February 2016 Revised: 28 April 2016 Accepted: 25 May 2016

A MULTI-CRITERIA HEURISTIC ALGORITHM FOR PERSONALIZED ROUTE PLANNING

Sinem BOZKURT KESER[1, *], Ahmet YAZICI[1] , Serkan GÜNAL[2]

[1] Department of Computer Engineering, Eskisehir Osmangazi University Eskişehir, Turkey
[2] Department of Computer Engineering, Anadolu University Eskişehir, Turkey

ABSTRACT

This paper proposes a heuristic function for multi-criteria route planning problems. The Analytical Hierarchy Process (AHP) is used for the multi-criteria aggregation process both for actual and heuristic cost functions. Travel distance, travel time, safety and fuel consumption are considered to be the selected criteria. Additionally, while considering real data sets, road safety and fuel consumption models are developed. The proposed multi-criteria heuristic function is consistent; therefore, the A* algorithm finds optimal routes. The proposed algorithm is tested and compared with existing algorithms in the literature using a real dataset for a specific region in Eskisehir, Turkey.

Keywords: Multi-criteria optimization, Heuristic, A* search algorithm, Route planning, Driver preference

1. INTRODUCTION

Route guidance systems (RGSs) are increasingly being implemented in a wide range of vehicle navigation applications. RGSs provide pre-trip information about which route to choose, the departure time and whether to undertake the trip; additionally, en-route information can be provided about accidents, congestion, and adverse weather conditions [1]. The desired target point can be accessed by alternative ways using the shortest time, the safest way, the least fuel consumption, and other factors. For this purpose, as a component of RGSs, Route Planning Systems (RPSs) are used to calculate the least cost path between the start and target points.

The Classical RPSs usually calculate a route while considering only a single criterion, i.e., the travel distance or travel time. In the RPS literature, a shortest path algorithm is used to generate a route for various types of route planning applications. In [2], the route planning module in OSU-ACT Autonomous Vehicle planned routes for defined missions considers the minimum time criterion. In [3], an efficient hierarchical routing algorithm that finds the best routes using the minimum travel distance criterion is presented. This study proposes an improved heuristic that reduces the number of route computations and a network pruning technique that reduces the search space in the road network. In [4], the dynamic shortest path problem is solved in a network by using time-varying travel times. In this study, the A* algorithm is adapted to compute the minimum travel time path. In [5], an incremental search approach with novel heuristics based on a variation of the A* algorithm is proposed to calculate the shortest distance path between a moving object and its target in a dynamic network in which traffic conditions are updated in real time. In [6], k-shortest path algorithm is applied in proposed time-expanded graph model for public transport network. Heuristic search strategies can be used to improve the computational speed of shortest path algorithms for a single criterion [7]. In [8], shared nothing architecture is applied to decrease the computation time of finding minimum cost paths in large transportation networks. In this study, the shortest path algorithm is a variation of the A* algorithm. The concurrent version of the algorithm simultaneously explores the search space by

[*] Corresponding Author: sbozkurt@ogu.edu.tr

utilizing dynamically created agents across multiple disk nodes, and the parallel version of the algorithm breaks the problem into a set of smaller sub problems by exploiting a set of intermediate nodes that the shortest path passes through.

In all of the above studies, route planning is performed by considering a single criterion. However, drivers consider more than one criterion while planning a route. Therefore, multi-criteria route planning has also received increased attention in the RPS literature. In [9], a multi-criteria route planning strategy that considers road safety and travel time as user selected criteria is presented. In this study, the Analytical Hierarchical Process (AHP) and regular increasing monotone quantifier-guided ordered weighted averaging operators (OWA) are used to integrate the driver preferences (e.g., the relative importance and decision strategy) to form a cost for each edge of a network. In addition, the Dijkstra algorithm is used to calculate the optimal route. In [10], in the context of the multi-criteria optimization problem, fuzzy set theory is applied to the dynamic routing problem. In this study, four criteria, namely, the travel time, traffic density, road type, and route length, are used to cover the drivers' preferences, and a symmetric decision model is applied to solve the multi-criteria optimization problem. In [11], an RGS is proposed to help drivers in deciding an optimum route based on individual preferences. In this study, the Takagi-Sugeno-Kang inference system is used to calculate the cost of each edge according to driver preferences. Then, the Dijkstra algorithm is used to calculate the least cost path from the set origin to the destination. In [12], an adaptable personalized route planning system is proposed. To aggregate the driver preferences and decision strategies, pairwise comparison methods and quantifier-guided OWA operators are used. This model also determines a personalized route with a robust deviation shortest route algorithm. In [13], an ontology-based multi criteria decision model (AHP) is proposed to calculate personalized route based on user's preferences. In [14], personalized, context-aware driving preferences from trajectories are obtained considering the criteria; travel distance, travel time, and fuel consumption. In [15], Time-based Dynamic Weight for Dijkstra algorithm is proposed to compute the most efficient time and minimum fuel consumption based on real condition of the traffic profiles of the road network. In the above studies, heuristic costs are not considered during the route planning. Usually, traditional shortest path algorithms such as Dijkstra are used in these studies. It is well known that the use of heuristics increases the performance of route-planning algorithms [5]. However, to the best of the authors' knowledge, heuristic search strategies have not been used in this context yet.

In this study, a heuristic cost function is defined for multi-criteria route planning problems. Then, the A* search algorithm is used to calculate the optimal routes. The AHP is used to weight each criterion according to user preferences. Then, a linear combination is applied to the aggregation of the multi-criteria with their weights to calculate the cost of each edge. In applications, travel time, travel distance, safety and fuel consumption criteria are considered for the traffic network. Additionally, a real data sets road safety model is also developed for each edge of the network. Furthermore, Vehicle-Specific Power (VSP), a function of the vehicle speed, acceleration and road grade, is used in the fuel consumption calculation for each edge. The proposed approach is tested for a specific region in Eskişehir. The test results show that the proposed algorithm uses less memory and time compared with the counterpart algorithms in the literature.

The remainder of the paper is as follows. The concepts of multi-criteria route planning and the criteria used in this study are defined in Section 2. The proposed heuristic function and the integration of the A* algorithm are given in Section 3. To demonstrate the effectiveness of the proposed approach, tests are conducted, and the results are presented in Section 4. The conclusions are given in the final section.

2. MULTI-CRITERIA ROUTE PLANNING

Multi-criteria route planning calculates a route from a start point to a target point of a road network while considering more than one criterion. A road network can be represented as a directed graph $G(V,E)$ that is composed of a node set V and an edge set E . The edge set E is a subset of the set $V \times V$ Cartesian product. Here, $v_k \in V(k=1,2,...,n)$ and $(v_i,v_j) \in E$ $i,j=1,2,...,n, i \neq j$. In that graph $G(V,E)$, any node $v_k \in V$ can be identified by its coordinates (longitude, latitude) and altitude information, which can be represented by x_k, y_k and h_k, respectively. Each element (v_i,v_j) in E denotes a directed edge that joins nodes v_i and v_j (from v_i to v_j) and is associated with a cost C_{ij} according to the defined criteria. If edge (v_i,v_j) is in the set E , then node v_j is the neighbour of node v_i. The node series $(v_1,v_2,v_3,...,v_k)$ forms a path from the start node s to the destination node d ; $s=v_1, d=v_k$, and the edges $(s,v_2),(v_2,v_3),...,(v_{k-1},d)$ are in the set E . The cost of the path is equal to the sum of the edge costs, as represented by $\sum_{\mathrm{i,j=1}}^{\mathrm{k}} \mathrm{C_{ij}}, \mathrm{i} \neq \mathrm{j}$. The optimum path from node s to d ; is the least cost path according to the preferred criteria.

Multi-criteria route planning offers drivers a variety of criteria for route planning. The criteria could be the shortest travel distance, the shortest travel time, the safest path, or the minimum fuel consumption, for example. Drivers can prefer different routes although they have the same start and target points. Driver preferences play a major role in the multi-criteria route planning process. According to driver preference information, the weight of each criterion can be determined by using the AHP. The AHP is one of the most developed multi-criteria decision-making methods [16]. The AHP establishes a hierarchical formulation and allows the integration of different types of criteria in the decision-making process.

2.1. Criteria Used in Cost Calculation

Drivers can choose road safety, travel time, travel distance, travel reliability, quality of road, width, slope, or number of scenic landscapes as criteria when travelling on a route. In this study, the travel distance (d_{ij}), travel time (t_{ij}), safety (s_{ij}) and fuel consumption (f_{ij}) are considered in the application. The travel distance can be calculated using Euclidean distance. Travel time on any edge (v_i,v_j) can be calculated using Eq. (1),

$$t_{ij} = d_{ij} / sl_{ij}(s) \tag{1}$$

where $\mathrm{sl_{ij}}(\mathrm{m/s})$ is the speed limit on the edge (v_i,v_j).

Road safety is another essential parameter that can be used in the cost calculation. The road safety depends on various parameters, such as road classification and weather conditions. The risk can be reduced through greater insight into the impact of road classification [17]. In this study, functional road class and form of way data are used to determine the safety of each road segment. Functional road class and form of way are given in Table 1 and Table 2, respectively. These data are obtained from Tomtom navigation software.

Table 1. Functional road class

Road Class	Example
A1	major roads
A2	local roads

Table 2. Form of way

Road Form	Example
B1	Dual carriage way
B2	Single carriage way
B3	Secondary, roundabout, poor condition ways

By using these data, a flowchart for the road safety model is proposed in Figure 1.

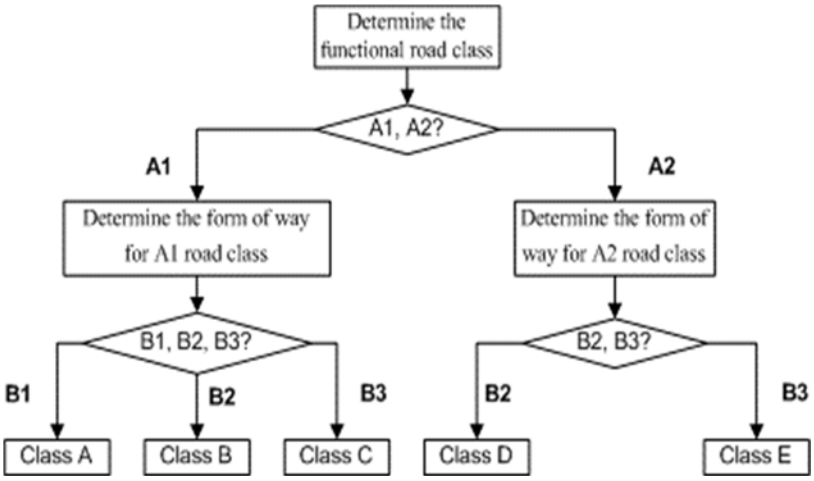
Determine the functional road class
A1, A2?
A1
A2
Determine the form of way for A1 road class
Determine the form of way for A2 road class
B1, B2, B3?
B2, B3?
B1
B2
B3
B2
B3
Class A
Class B
Class C
Class D
Class E

Figure 1. Flowchart of the road safety model

In this flowchart, we first classify road segments according to the functional road class. Then, road segment classification is continued according to the form of way information. As a result, each road segment is classified into one of five classes; from class A to class E. Class A represents the safest road segment, whereas class E is the least safe segment. Road safety parameters change between 1 and 5 from class A to class E to identify the safety degree of each edge. As the number increases, the safety decreases. In the end, the safety cost of the edge (v_i, v_j) can be calculated by multiplying its square of safety degree and travel distance, as given in Eq. (2),

$$s_{ij} = s\deg_{ij}^2 d_{ij} \tag{2}$$

where

$$s\deg_{ij} = \begin{cases} 1 \ (s\deg_{ij} = A) \\ 2 \ (s\deg_{ij} = B) \\ 3 \ (s\deg_{ij} = C) \\ 4 \ (s\deg_{ij} = D) \\ 5 \ (s\deg_{ij} = E) \end{cases} \tag{3}$$

and $s\deg_{ij}$ is the safety degree of edge (v_i, v_j).

The fuel consumption for any road segment can be calculated considering the vehicle type, vehicle speed, acceleration and grade of the road segment. For this purpose, the Vehicle Specific Power (VSP) is used to determine the fuel consumption [18]. The VSP value $(kw/\,metric\ ton)$ can be calculated in [17] as

$$vsp_{ij} = avgspeed_{ij}[1.1a_{ij} + 9.81\sin(\arctan(g_{ij})) + 0.132] + 0.000302 avgspeed_{ij}^3 \tag{4}$$

where $g_{ij}, avgspeed_{ij}$, and a_{ij} are grade, average speed (m/s), and acceleration (m/s^2) on edge (v_i, v_j), respectively. Additionally, the numerical values 0.132 and 0.000302 are rolling resistance term coefficient (m/s^2) and drag term coefficient (m^{-1}), respectively.

As seen in Eq. (5), the VSP value of edge (v_i, v_j) is multiplied by its travel time value to estimate the fuel consumption cost of this edge.

$$f_{ij} = vsp_{ij}t_{ij} \tag{5}$$

The criteria given above are normalized and integrated to be comparable among each other. The maximum score method in [19] is used for this purpose. In this method, the normalization value is calculated by

$$x_i^{'} = \frac{x_i}{x_{\max}} \tag{6}$$

where $x_i^{'}$ is the normalized value, x_i is the raw value of a given criterion for the i^{th} criterion, and $x_{\max}$ is the maximum score of the criterion in the existing network. The results are obtained from the normalization range of 0 to 1.

Original and normalized values for each criterion for an edge(v_i, v_j) are shown in Table 3. In this table, $d_{ij}^{'}, t_{ij}^{'}, s_{ij}^{'}$ and $f_{ij}^{'}$ represent normalized values of each criteria.

Table 3. Criteria normalization.

Criteria	Range of Original Values	Normalized Values
Travel distance	$[0\ \infty)$	$[{}^{d_{\min}}/_{d_{\max}}\ 1], d_{ij}^{'} = d_{ij}/d_{\max}$
Travel time	$[0\ \infty)$	$[{}^{t_{\min}}/_{t_{\max}}\ 1], t_{ij}^{'} = t_{ij}/t_{\max}$
Safety	$[0\ \infty)$	$[{}^{s_{\min}}/_{s_{\max}}\ 1], s_{ij}^{'} = s_{ij}/s_{\max}$
Fuel consumption	$[0\ \infty)$	$[{}^{f_{\min}}/_{f_{\max}}\ 1], f_{ij}^{'} = f_{ij}/f_{\max}$

2.2. Cost Calculation of Each Edge

In multi-criteria route planning, criteria values are aggregated to calculate the cost of each edge of the traffic network. These criteria values are combined with some relative weights that represent driver preferences. These relative weights can be determined through the AHP. The AHP is one of the multi-criteria decision-making methods that are widely used in the literature [12].

The AHP procedure for relative weight calculation is explained through an example with four criteria that is used in this study. In the AHP, a scale of numbers is required to make a pairwise comparison. For this purpose, the linguistic terms equal, moderate, strong, very strong and extreme importance are enumerated using the numbers 1, 3, 5, 7 and 9, respectively. The pairwise comparison for the criteria is given in Table 4. In this table, the diagonal elements are set to 1. The upper triangle part of this table is formed according to driver preferences. Each element of the lower triangle is set to the inverse of these pairwise comparisons automatically. As seen in Table 4, a ‘very strong’ linguistic term is selected to judge the importance of the travel time over the safety, which is enumerated with 7. Then, the importance of the safety over the travel time is set to 1/7 automatically.
The driver preferences in Table 4 are transformed into the relative weights as follows:

Table 4. Normal pairwise comparison of criteria

	Travel distance	Travel time	Safety	Fuel consumption
Travel distance	1	3	1/5	1/7
Travel time	1/3	1	1/7	1/9
Safety	5	7	1	1/3
Fuel consumption	7	9	3	1

Construct a matrix A from Table 4 as

$$A = \begin{bmatrix} 1.00 & 3.00 & 0.20 & 0.14 \\ 0.33 & 1.00 & 0.14 & 0.11 \\ 5.00 & 7.00 & 1.00 & 0.33 \\ 7.00 & 9.00 & 3.00 & 1.00 \end{bmatrix}$$

Then, the following equation

$$\frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}} \tag{7}$$

is applied to each cell of matrix A to obtain the following matrix D:

$$D = \begin{bmatrix} 0.075 & 0.150 & 0.046 & 0.090 \\ 0.025 & 0.050 & 0.033 & 0.070 \\ 0.375 & 0.350 & 0.230 & 0.210 \\ 0.525 & 0.450 & 0.691 & 0.630 \end{bmatrix}$$

After this step, the column vector U is created by the arithmetic mean of each row of the D matrix.

$$U = \begin{bmatrix} 0.090 \\ 0.044 \\ 0.291 \\ 0.574 \end{bmatrix}$$

Each entry of the column vector U corresponds to the relative weights. In this example, the relative weights $u_1 = 0.090, u_2 = 0.044, u_3 = 0.291$ and $u_4 = 0.574$ correspond to the travel distance, the travel time, the safety, and the fuel consumption, respectively, where $\sum_{i=1}^{4} \mathrm{u_i} = 1$.

In the AHP procedure, pairwise comparisons of the criteria are accomplished. If there are more than two criteria, there could be inconsistency. For example, the travel distance might be judged as important as the travel time and the travel time as important as the safety, but the travel distance is not ranked to be as important as the safety. Then, inconsistency occurs in the pairwise comparison. To assure the consistency of the pairwise comparison, the consistency ratio (C.R.)

$$C.R. = \frac{\lambda_{\max} - n}{R.I.(n-1)} \tag{8}$$

is used. For consistency, the value of *C.R.* should be less than 0.1. In Eq. (8), $\lambda_{\max}$ is the largest eigenvalue of the pairwise comparison matrixA, n is the number of criteria to be compared, and *R.I.* is the random consistency index. The *R.I.* is a constant value for the criteria that are defined in the literature. The values of *R.I.* depend on n and are given in Table 5.

Table 5. R.I. values

n	R.I.	n	R.I.
1	0.00	7	1.32
2	0.00	8	1.41
3	0.58	9	1.45
4	0.90	10	1.49
5	1.12	11	1.51
6	1.24	12	1.48

In our example, the consistency ratio is calculated as 0.0997, which is less than 0.1 and is acceptable. After obtaining the relative weight for each criterion, a linear combination is applied to the aggregation of the multi-criteria with their weights. Thus, different types of criteria are provided together, and a single cost function is achieved. According to the above equations, an edge cost C_{ij} on edge $(v_i, v_j), i \neq j$, in the network *G(V,E)* can be calculated by Eq. (9).

$$c(v_i, v_j) = C_{ij} = \sum_{l=1}^{k} (x_{ij}^{l})' u_l = (x_{ij}^{1})' u_1 + (x_{ij}^{2})' u_2 + ... + (x_{ij}^{k})' u_k \tag{9}$$

where $(x_{ij}^{l})' (l = 1,2,...,k)$ are the normalized values for the selected criteria on the edge (v_i, v_j), which are calculated in Eq. (6).

The last step is to apply a least-cost path algorithm on the network created above.

2.3. Route Planning Algorithms

The least-cost path algorithm can be used to calculate the multi-criteria route after calculating the aggregated cost of each edge. The Dijkstra algorithm is the most widely used algorithm in route planning [20]. Typically, this algorithm finds the least cost path from an initial node to all of the other nodes in the graph. If the algorithm is terminated when the target node is found, it finds the optimum path between an initial node and a destination node. This approach was developed for weighted graphs; the weight of the each edge must be equal to zero or greater than zero.

The Dijkstra algorithm maintains two lists, the open list and the closed list. The open list keeps nodes that must be examined, while the closed list keeps nodes that have already been examined. The Dijkstra algorithm builds a shortest path from the source node to all of the other nodes. Initially, the source is assigned to the open list. At each step, the node with the least cost (the minimum cost from the start node to the selected node) is selected from the open list, the nodes reached are added to the open list, and the selected node is transferred from the open list to the closed list. When the open list is empty or the destination node is added to the closed list, then the algorithm terminates. The original Dijkstra algorithm has a complexity of $O(n^2)$, where n represents the number of nodes in the graph. However, when using a priority queue data structure, the complexity of the algorithm can be reduced to $O(\mathrm{m} + \mathrm{nlogn})$, where m represents the number of edges in the graph.

In [9], the Dijkstra algorithm is used to calculate the multi-criteria route. Because the Dijkstra algorithm is designed to calculate the least cost path from a single source node to all of the other nodes in the network, it could expand a larger number of nodes for a single destination node. In the literature, more effective algorithms for single criterion and single origin/destination pair shortest path problems are proposed [21]. These algorithms use heuristic functions to obtain a better performance. To the best of the authors' knowledge, these algorithms are not used for multi-criteria single source/destination path planning problems. In the following section, a heuristic search algorithm is proposed to solve the multi-criteria route planning problem.

3. PROPOSED METHOD

Multi-criteria route planning approaches result in more realistic routes for drivers. The reason is that the relative importance of each criterion is determined by the driver preferences. Then, criteria values are aggregated to form a single cost function. These aggregated cost values are used to calculate the route from a source to a destination. In the literature, the Dijkstra algorithm is widely used to find the least cost path for this route planning. For single source/destination route planning problems, the A* algorithm is more effective than the Dijkstra algorithm [5]. The reason is that the Dijkstra algorithm finds the minimum costs from the source node to all of the other nodes, whereas the A* algorithm finds the minimum cost from the source node to the destination node. To achieve this goal, the A* algorithm uses a heuristic function to search for the most promising nodes first for the given destination node. The heuristic function is an estimate of the cost of the cheapest path from the current node to the goal node.

Theorem 1: A heuristic $h(n)$ is said to be consistent if, for each node n and every successor $n^{'}$ of n generated by an action, the estimated cost of reaching the goal from n is no greater than the step cost of reaching $n^{'}$ plus the estimated cost of reaching the goal from $n^{'}$:

$h(n) \leq c(n,n^{'}) + h(n^{'})$. (triangle inequality)

where $c(n,n^{'})$ is the actual cost of reaching $n^{'}$ from n.

Proof: Let $h^{*}(n)$ be the cost of the least cost path from n to the goal node. It will be proven by induction on the number of steps to the goal that $h(n) \leq h^{*}(n)$.

Base case: If there are 0 steps to the goal from node n, then n is the goal and, therefore,

$$h(n) = 0 \leq h^{*}(n).$$

Induction step: If n is i steps away from the goal, then there must be some successor $n^{'}$. Here, $n^{'}$ is on the optimal path from n to the goal, and $n^{'}$ is $i-1$ steps away from the goal. Therefore,

$$h(n) \leq c(n,n^{'}) + h(n^{'})$$

However, by the induction hypothesis, $h(n^{'}) \leq h^{*}(n^{'})$. Therefore,

$$h(n) \leq c(n,n^{'}) + h^{*}(n^{'}) = h^{*}(n)$$

because $n^{'}$ is on the optimal path from n to the goal□

Theorem 2: Every consistent heuristic is also admissible.

Proof: Let $h(n)$ be any consistent heuristic, and let $c(n,n^{'})$ be the corresponding step cost of moving from node n to node $n^{'}$. Then, $h(n) \leq c(n,n^{'}) + h(n^{'})$ by the definition of consistency. To show that h is admissible, we must show that $h(n) \leq h^{*}(n)$, where h^{*} is the actual cost of n.

Suppose that there is no path from n to the goal state. In this case, $h^{*}(n)$ is infinite and satisfies $h(n) \leq h^{*}(n)$ for any finite $h(n)$.

Next, suppose that there is some path from n to a goal state g. Let us represent the intermediate nodes of a shortest path with the variables $\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_1$. Thus, the least cost path from n to g is expressed as $(\mathrm{n}, \mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_1, g)$. Now, consider $h(\mathrm{v}_1)$. Because h is consistent, $\mathrm{h}(\mathrm{v}_1) \leq \mathrm{c}(\mathrm{v}_1, \mathrm{g}) + \mathrm{h}(\mathrm{g}) = \mathrm{c}(\mathrm{v}_1, \mathrm{g})$, since because $\mathrm{h}(\mathrm{g}) = 0$.

Similarly,

$$h(v_{l-1}) \leq c(v_{l-1}, v_l) + h(v_l) \leq c(v_{l-1}, v_l) + c(v_l, g)$$

By induction on 1, we obtain

$$h(v_k) \le c(v_k, v_{k+1}) + \ldots + c(v_{l-1}, v_l) + c(v_l, g)$$

for any k between 1 and l-1. However, because $(\mathrm{n}, \mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_l, g)$ is a least cost path,

$$h^*(v_k) = c(v_k, v_{k+1}) + \ldots + c(v_{l-1}, v_l) + c(v_l, g)$$

which implies that $\mathrm{h}(\mathrm{v}_k) \le \mathrm{h}^*(\mathrm{v}_k)$, as desired □

If a consistent heuristic can be defined in multi-criteria route planning, better results would be obtained by using the A* algorithm. However, to the best of the authors' knowledge, heuristic functions are not used for multi-criteria route planning yet.

3.1. Heuristic Cost Calculation for Multi-criteria Route Planning

The A* algorithm uses a heuristic function to improve the search efficiency. It expands the successor nodes that are closer to the goal node in terms of the determined cost. In the following, we explain the heuristic cost calculation for each criteria. Let $\mathrm{v}_\mathrm{j}(\mathrm{x}_\mathrm{j}, \mathrm{y}_\mathrm{j}, \mathrm{h}_\mathrm{j})$ denote the node that is a heuristic value to be calculated, and let $\mathrm{v}_l(\mathrm{x}_l, \mathrm{y}_l, \mathrm{h}_l)$ denote the destination node. The heuristic function for the travel distance is calculated using Euclidean distance equation.

Euclidean distance corresponds to the straight-line distance from node v_j to v_l. The straight-line distance is admissible because this distance is the shortest possible length of any route that connects node v_j with the goal node v_l. Thus, it is a lower bound of the actual route distance from node v_j to node v_l i.e., $h_d(\mathrm{v}_\mathrm{j}) \le \mathrm{h}_\mathrm{d}^*(\mathrm{v}_\mathrm{j})$, where $\mathrm{h}_\mathrm{d}^*(\mathrm{v}_\mathrm{j})$ is the actual route distance from node v_j to node v_l. Thus, the straight-line cannot be an overestimate. According to the definition of consistency given in Theorem 1, the straight-line distance is also a consistent heuristic.
The heuristic cost calculation for the travel time criterion is given in Eq. (10).

$$h_t(v_j) \le h_d(v_j) / sl_{\max} \tag{10}$$

Where $\mathrm{sl}_{\max}$ is the maximum speed limit on the traffic network. As seen in Eq. (10), the Euclidean distance $h_d(v_j)$ divided by the maximum speed $\mathrm{sl}_{\max}$ can be used as a heuristic value. Because the admissibility of $\mathrm{h_d(v_j)}$ is proven, the heuristic value $h_t(v_j)$ cannot be an overestimate. Additionally, it is consistent.

Heuristic cost calculation for safety is given in Eq. (11).

$$\mathrm{h}_\mathrm{s}(\mathrm{v}_\mathrm{j}) = \mathrm{sdeg}_{\min}^2 \mathrm{h}_\mathrm{d}(\mathrm{v}_\mathrm{j}) = 1^2 \mathrm{h}_\mathrm{d}(\mathrm{v}_\mathrm{j}) = \mathrm{h}_\mathrm{d}(\mathrm{v}_\mathrm{j}) \tag{11}$$

According to the road safety model that is given by Eq. (3), the numbers 1 to 5 are used to represent the safety degree of each edge. The square of this value is multiplied by the distance of the edge to obtain the safety value of this edge (see Eq. (2)). In Eq. (11), the smallest safety degree $\mathrm{sdeg}_{\mathrm{ij}} = 1$ in the network is selected for the heuristic cost calculation for safety. Then, the square of this value is multiplied by the Euclidean distance $h_d(v_j)$. As a result, we have a lower bound for the real cost value for the safety from node v_j to the goal node, and it is never overestimated. Because the triangle inequality given in Theorem 1 is validated by $h_s(v_j)$, this heuristic is also a consistent heuristic.

The heuristic model for the fuel consumption is given in Eq. (13).

$$h_{vsp}(v_j) = 0.132 avgspeed_{\min} + 0.000302 avgspeed_{\min}^3 \tag{12}$$

$$h_f(v_j) = h_{vsp}(v_j) h_t(v_j) = (0.132 + 0.000302 avgspeed_{\min}^2) h_d(v_j) \tag{13}$$

Here, we relax the fuel consumption calculation by setting the grade and acceleration to 0 and $\text{avgspeed}_{\min}$ to the minimum average speed in the network. This cost is always less than the actual fuel consumption cost. Therefore, the cost value of the relaxed problem (see Eq. (13)) is an admissible heuristic for the original problem. Because the derived heuristic is an exact cost for the relaxed problem, it must obey the triangle inequality and is therefore consistent.

The heuristic value for each criterion is normalized as follows:

$$h_d^{'}(v_j) = {h_d(v_j)}/{d_{\max}}, h_t^{'}(v_j) = {h_t(v_j)}/{t_{\max}}, h_s^{'}(v_j) = {h_s(v_j)}/{s_{\max}}, h_f^{'}(v_j) = {h_f(v_j)}/{f_{\max}} \tag{14}$$

Note that the maximum values used in Eq. (14) are the same as the values in Table 3 that are used for the actual cost normalization.

Then, we also apply the same aggregation that is used for the actual cost calculation (see Eq. (9)) to the normalized heuristic function values for the node v_j as follows:

$$h(v_j) = h_d^{'}(v_j)u_1 + h_t^{'}(v_j)u_2 + h_s^{'}(v_j)u_3 + h_f^{'}(v_j)u_4 \tag{15}$$

This linear equation can be generalized to the k criteria given in Eq. (16).

$$h(n) = \sum_{l=1}^{k} h_l^{'}(n)u_l = h_1^{'}(n)u_1 + h_2^{'}(n)u_2 + ... + h_k^{'}(n)u_k \tag{16}$$

where n represents the node that is a heuristic value to be calculated.

Theorem 3: The multi-criteria heuristic cost value given by Eq. (16) is consistent.
Proof: The multi-criteria heuristic function value is calculated as a linear combination of the heuristic value of each criterion that is proven to be consistent. Because the relative weights $u_1,...,u_k$ are the same as the aggregation of actual criteria costs, the resulting heuristic is also consistent □

3.2. The A* Search Algorithm for Multi-criteria Route Planning

A* is a search algorithm that is used to find a path from the initial node to the goal node. It employs a heuristic function to select the best node to process next. If the heuristic function is consistent, then it finds the optimum path. The A* algorithm uses the evaluation function

$$f(n) = g(n) + h(n) \tag{17}$$

to calculate the optimum path. In Eq. (17), $g(n)$ is the actual multi-criteria cost of an optimal path from the initial node to node n, $h(n)$ is the multi-criteria heuristic cost of an optimal path from node n to the goal node calculated using Eq. (16), and $f(n)$ is the estimate of the best solution that goes through node n.

Corollary 1: If h(n) is consistent, then the values of $f(n) = g(n) + h(n)$ along any path are non-decreasing.

Proof: Suppose that n' is a successor of n; then, $g(n') = g(n) + c(n, n')$, and

$$f(n') = g(n') + h(n') = g(n) + c(n, n') + h(n') \geq g(n) + h(n) = f(n) \square$$

Corollary 2: If A* uses a consistent multi-criteria heuristic, then A* results in the optimal path for the multi-criteria search.

Proof: Suppose that A* uses a consistent multi-criteria heuristic function. Then, according to Corollary 1, the values of $f(n)$ along any path are non-decreasing. In this case, a node cannot be re-expanded. In other words, if A* expands a node, then the optimal path to that node has already been found .□

Pseudo code of the A* algorithm for the proposed method is shown in below.

```
Algorithm: A* algorithm for multi-criteria route planning
Input parameters:
V: node set
E: edge set
s:  source node
d: destination node
A: relative importance matrix
A* Algorithm(){
for each $v_k$ in V do
    $g(v_k) = \infty$
    $parent(v_k) = null$
end for
$g(s) = 0$
$f(s) = g(s) + h(s)$
$OpenList.Add(s)$
$ClosedList = \varnothing$
while $OpenList != \varnothing$ do
    $v_k = OpenList.FindMin()$
    $ClosedList.Add(v_k)$
    $OpenList.\mathrm{Re}move(v_k)$
    if $\mathrm{v_k = d}$
        break
    end if
    for each $v_j$ in $v_k.neighborList()$
      if $g(v_j) > g(v_k) + C_{kj}$
            $g(v_j) = g(v_k) + C_{kj}$
                  1.          $f(v_j) = g(v_j) + h(v_j)$
            $parent(v_j) = v_k$
            if $v_j \notin OpenList$
                  $OpenList.Add(v_j)$
            end if
      end if
    end for
end while
```

The inputs of the A* algorithm are the source and destination nodes, the relative importance matrix and the network of the environment model. The A* algorithm maintains two sets, the open list and the closed list. The open list keeps track of those nodes that must be examined, while the closed list keeps track of nodes that have already been examined. Initially, the open list contains only the initial node, and the closed list is empty. A* has a main loop that repeatedly gets the node that has the lowest $f(n)$ value from the open list. If n is the goal node, then we are done, and all that is left to do is to return the solution by backtracking from node n. Otherwise, we remove node n from the open list and add it to the closed list. Next, we generate all of the possible successor nodes of node n. For each successor node of noden, if it is already in the closed list, then we can discard the node. Similarly, if it is already in the open list and there has been an equal or lower f estimate, then we can discard this node. Otherwise, it is added to an open list with its updated f value. Since the A* algorithm guarantees finding a path if one exists, it is complete. Its optimality is also proven by Corollary 2. The complexity of A* depends on the multi-criteria heuristic function. In the worst case, it is the same as the Dijkstra algorithm.

4. ILLUSTRATIVE EXAMPLE

The proposed multi-criteria route planning algorithm is tested and compared with the Dijkstra algorithm for a specific region in the city of Eskişehir. Figure 2 shows a picture of the selected region, which has different functional road classes (main roads, local roads) and road forms (dual carriageway, major roads, intersections, and poor condition roads). This situation leads to the formation of edges that have varying safety degrees and travel time. The selected region also has road segments with different grades that result from different fuel consumption at the same travel velocity.

Figure 2. Representation of the test environment

The corresponding network of the environment is given in Figure 3. This picture is obtained from Tomtom navigation software. In this study, intersections and streets are represented by nodes and edges, respectively. The network of the test region has 1165 nodes and 2895 edges. For each edge, the values of the speed limit, average speed, functional road class and road form are available. The Tomtom database stores each node with its latitude and longitude information. The information on each node is transferred to a local x, y coordinate system. In addition, the altitude information on each node is obtained through a web service using latitude and longitude information. Each node $v_k(x_k, y_k, h_k)$ is represented within the x, y, z coordinate system. Five of these nodes are selected for the test and comparison of the algorithms. They are shown in Figure 3 with dark circles.

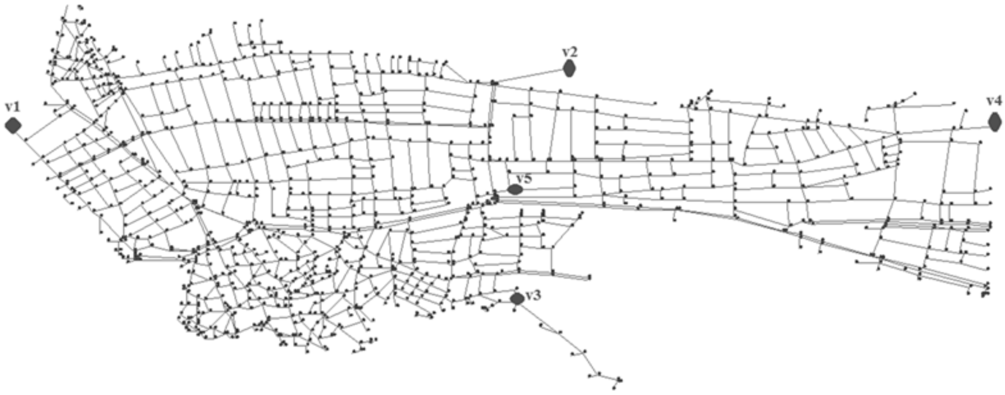

Figure 3. Network structure of the test environment

The effects of driver preferences in calculating optimal routes were studied in Bozkurt et. al. (2012), using the Dijkstra algorithm. In this study, it is assumed that the driver selects an "equal" importance for the travel time, distance, safety, and fuel consumption criteria; therefore, the relative weights are calculated by $u_1 = 0.25, u_2 = 0.25, u_3 = 0.25$ and $u_4 = 0.25$, respectively. Then, using these driver preferences, the five nodes are used for the comparison of the A* search for multi-criteria route planning and the Dijkstra algorithm. For each test node, the routes to all of the other remaining nodes in the network are calculated using both algorithms. The algorithms are compared in terms of the number of nodes added to a closed list and the calculation time. The results are shown in Table 6 and Table 7.

In Table 6, for each starting test node, the maximum value, the average value and the variance of the number of nodes that are added to the closed list are given for both the Dijkstra and A* algorithms. As shown in Table 6, a smaller number of nodes are added to the closed list by the A* algorithm with the proposed multi-criteria heuristic. The last column of Table 6 shows the percentage improvement in the average values with the proposed multi-criteria heuristic.

Table 7 is formed similar to Table 6 except that each cell of the table contains time values that are taken by the algorithms to be in milliseconds. As seen in the last column of this table, the A* algorithm performs better than the Dijkstra algorithm.

Both methods produce the same optimal routes for the same start and goal nodes. But the A* algorithm with the proposed heuristic function adds less number of nodes in the closed list (i.e. use less memory) during the optimal route calculation. Thus, the proposed A* algorithm for multi-criteria route planning has used less memory and time. According to the test results, the proposed multi-criteria heuristic function improves the search efficiency.

Table 6. Number of nodes in the closed list for Dijkstra and A* algorithms

	DIJKSTRA			A*			Improvement Percentage in Terms of Avg. Values
	Node Number in the Closed List			Node Number in the Closed List			
	Max	Avg.	σ	Max	Avg.	σ	
v_1	1140	571	328.95	1103	426.92	271.90	25%
v_2	1140	571	328.95	1128	385.20	304.78	33%
v_3	1140	571	328.95	741	290.62	175.46	49%
v_4	1140	571	328.94	1127	455.74	278.53	20%
v_5	1140	571	328.94	1125	339.65	269.52	41%

Table 7. Computation time for Dijkstra and A* algorithms

	DIJKSTRA			A*			
	Computational Time Values			Computational Time Values			Improvement Percentage in Terms of Avg. Values
	Max	Avg.	σ	Max	Avg.	σ	
v_1	11.13	5.41	3.20	10.53	3.99	2.61	26%
v_2	10.77	5.40	3.05	10.60	3.66	2.86	32%
v_3	11.10	5.56	3.17	7.18	2.86	1.70	49%
v_4	10.90	5.40	3.07	10.67	4.33	2.59	20%
v_5	10.94	5.42	3.09	10.67	3.23	2.56	40%

5. CONCLUSION

In this study, a multi-criteria route planning algorithm is proposed. The travel distance, travel time, safety, and fuel consumption are considered to be the selected criteria, and models for safety and fuel consumption are proposed. The AHP is used for a multi-criteria aggregation process for both the actual and heuristic cost calculations. To the best of authors' knowledge, only actual costs are used for calculating the optimal routes in the multi-criteria route planning. In this study, a heuristic function is proposed to improve the search efficiency. The proposed heuristic function is consistent; therefore, the A* algorithm finds optimal routes. The test results show that the proposed algorithm adds less number of nodes in the closed list. So, it uses less memory and time compared with the most often used counterpart algorithms.

In future work, the proposed method can be developed by improving the proposed multi-criteria heuristic function. This would reduce the number of nodes that examined during the route planning process.

REFERENCES

[1] Herbert W, Mili F. Route Guidance: state of the art vs. state of the practice. In: IEEE 2008 Intelligent Vehicles Symposium; 2008; Eindhoven, NETHERLANDS: IEEE. pp. 988-995.

[2] Fu L, Yazici A, Ozguner U. Route planning for OSU-ACT autonomous vehicle in DARPA urban challenge. In: IEEE 2008 Intelligent Vehicles Symposium; 2008; Eindhoven, NETHERLANDS: IEEE. pp. 928-933.

[3] Jagadeesh GR, Srikanthan T, Quek KH. Heuristic techniques for accelerating hierarchical routing on road networks. IEEE Transactions on Intelligent Transportation Systems 2002; 3: 301-309.

[4] Chabini I, Lan S. Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. IEEE Transactions on Intelligent Transportation Systems 2002; 3: 60-74.

[5] Huang B, Wu Q, Zhan FB. A shortest path algorithm with novel heuristics for dynamic transportation networks. International Journal of Geographical Information Science 2007; 21: 625-644.

[6] Olczyk A, Galuszka A. Finding routes in a public transport network. A case study. In: IEEE 19th International Conference on Methods and Models in Automation and Robotics (MMAR); 2014; Miedzyzdroje: IEEE. pp. 800-803.

[7] Fu L, Sun D, Rilett LR. Heuristic shortest path algorithms for transportation applications: state of the art. Computers & Operations Research 2006; 33: 3324-3343.

[8] Safar M. Minimum cost path for a shared nothing architecture. International Arab Journal of Informational Technology 2005; 2: 281-290.

[9] Bozkurt S, Yazici A, and Keskin K. A multicriteria route planning approach considering driver preferences. In: IEEE International Conference on Vehicular Electronics and Safety (ICVES); 2012; İstanbul, Turkey: IEEE. pp. 324-328.

[10] Wahle J, Annen O, Schuster C, Neubert L, Schreckenberg M. A dynamic route guidance system based on real traffic data. European Journal of Operational Research 2001; 131: 302-308.

[11] Lin IC, Chou SY, Hsu HY. Developing adaptive driving route guidance systems based on fuzzy neural network. In: IEEE 2009 International Conference on Systems on Man and Cybernetics (SMC 2009); 2009; Antonio, TX: IEEE. pp. 4293-4298.

[12] Nadi S, Delavar MR. Multi-criteria, personalized route planning using quantifier-guided ordered weighted averaging operators. International Journal of Applied Earth Observation and Geoinformation 2011; 13: 322-335.

[13] Niaraki AS, and Kim K. Ontology based personalized route planning system using a multi-criteria decision making approach. Expert Systems with Applications 2009; 36: 2250-2259.

[14] Yang B, Guo C, Ma Y, Jensen CS. Toward personalized, context-aware routing. The International Journal on Very Large Data Bases 2015; 24: 297-318.

[15] Rosyidi L, Pradityo HP, Gunawan D, Sari RF. Timebase dynamic weight for Dijkstra Algorithm implementation in route planning software. In: International Conference on Intelligent Green Building and Smart Grid (IGBSG); 2014. pp. 23-25.

[16] Saaty TL. The Analytic Hierarchy Process, Planning, Priority Setting. Resource Allocation 1980; McGraw-Hill, New York, NY, USA.

[17] Malenkovska TM, Donceva R, Bunevska J. Role of functional classification of highways in road traffic safety. Transport Problems 1980; 4: 97-104.

[18] Coelho MC, Frey HC, Rouphail NM, Zhai H, Pelkmans L. Assessing methods for comparing emissions from gasoline and diesel light-duty vehicles based on microscale measurements. Transportation Research Part D-Transport and Environment 2009; 14: 91-99.

[19] Malczewski J. GIS and Multicriteria Decision Analysis. Wiley 1999, New York, NY, USA.

[20] Dijkstra EW. A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1959, 1: 269-271.

[21] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 1968; 4: 100-107.