A Real-Time Vehicle Navigation Algorithm in Sensor Network Environments

C. L. Philip Chen, Fellow, IEEE, Jin Zhou, Student Member, IEEE, and Wei Zhao, Fellow, IEEE

Abstract—In a large-scale wireless sensor traffic network, collecting and processing of the global real-time traffic information are often unreliable. Making real-time navigation decision becomes an arduous task. To address this issue, an efficient wireless-sensor-network-based real-time vehicle navigation algorithm is proposed, in which multiple local traffic information is considered to make a navigation decision in a quick and accurate way. At the same time, a general distance metric is defined for the processing of both exact and fuzzy data. In addition, the algorithm can provide various navigation decisions according to the choice of different attributes to meet the diverse navigation requirements of drivers. Simulation results show the suitability and efficiency of the proposed algorithm.

Index Terms—Hybrid multi-attribute decision making, real-time vehicle navigation, wireless sensor networks (WSNs).

I. INTRODUCTION

In recent years, there has been a resurgence of interest in cyber-physical systems and intelligent transportation systems (ITSs) [1], [2]. This is directly attributed to the recent developments in vehicle navigation systems (VNSs), where there is an actual need to search the optimal routing path from a source to a destination within a reasonable time to achieve the least cost, i.e., travel time (TT) or travel distance (TD). Most of the research in this area focuses on developing increasingly efficient optimal algorithms. The Dijkstra [3] and the A* [4] algorithms are two well-known shortest path algorithms. Other variants of these two algorithms are further discussed and popularized in [5]–[7]. Various attempts to speed up the shortest path search for the case of single source and single target are reviewed by Wagner and Willhalm [8] and empirically compared by Bauer et al. [9].

However, existing approaches are mostly devised for static networks. They are not efficient when directly applied on dynamic shortest path planning in a real-time traffic environment. In practice, the navigation route needs to frequently be replanned according to the dynamics of the traffic conditions. Aside from the extensive computational studies on static shortest path algorithms [10], [11], some researchers pay more attention to the efficient reoptimization strategies on dynamic graphs [12]–[14].

Recently, a heuristic dynamic algorithm has been discussed in [15]. This algorithm considers both the changing locations of the vehicle and the changing traffic conditions in the navigation area. Based on a multipath routing strategy, Chen et al. [16] and Bell [17] propose multipath dynamic vehicle navigation algorithms.

Recently, with the development of microelectronic technologies, low-energy-consumption, low-cost, and relatively powerful wireless sensor networks (WSNs) have been implemented in ITSs [18], [19]. Small wireless sensors with integrated sensing, computing, and wireless communication capabilities offer tremendous advantages in traffic information collecting and vehicle tracking. A great deal of new energy-efficient and real-time solutions for traffic data sensing and collecting are discussed in [20]–[22]. At the same time, based on WSN techniques, much effort in vehicle navigation has been made. Huang et al. [23] put forward a distance-aware epidemic navigation algorithm using the vehicular ad hoc network. A new measure of TT reliability is introduced in [24] for dynamic routing. Chang et al. [25] apply three metrics (road traffic density, road class, and road distance) in ITSs to determine the optimal navigation route. Recently, Skog and Handel [26] have given a survey of car positioning and navigation technologies.

These dynamic vehicle navigation approaches are efficient in information collecting and traffic control, but there are still some problems, which are listed as follows.

1) The conventional traffic information center (TIC) collects global real-time traffic information through various media such as cable and wireless Internet and provides services for drivers, including navigation decisions. However, in a large-scale WSN-based traffic network, the conventional TIC approach may not be viable, because transmitting a large amount of global real-time traffic data is discouraged by the energy and bandwidth constraints of wireless sensors.

2) Most of the existing vehicle navigation algorithms try to pursue the minimum TD or the minimum TT. Usually, they cannot achieve an effective tradeoff.

3) Using only single traffic information or the simple mathematical model for the navigation decision making is not very sufficient. Comprehensive consideration of all the traffic information is needed to assist navigation.

An algorithm that can consider all kinds of local traffic information to make real-time navigation decisions in a quick and accurate way is needed. However, little has been reported on...
this type of dynamic vehicle navigation. Local traffic information such as road condition and vehicle status can be regarded as multiple attributes. As a result, the vehicle navigation can be considered as a multiattribute decision-making (MADM) problem [27]. MADM, which is a main research area that deals with complex decision problems, aims at implementing decent selections among alternatives that are associated with multiple attributes. Considering the navigation decision using multiple traffic information, we propose a novel WSN-based real-time vehicle navigation algorithm using the MADM method. In addition, because the real-time traffic information involved in the navigation decision making should not be all denoted by exact data, some of which are more suitable to be denoted by fuzzy data, a general distance metric is presented for the processing of both exact and fuzzy data in MADM. As a result, the proposed navigation algorithm provides a general methodology for vehicle navigation, i.e., it can provide various navigation decisions according to different attributes to meet diverse navigation requirements for drivers, for example, the least TD, the least TT, or a tradeoff of all conditions.

The rest of this paper is organized as follows. The WSN-based real-time vehicle navigation algorithm is presented in Section II. The real-time navigation decision-making strategy based on the hybrid MADM method is discussed in Section III. Section IV shows the simulation results. Finally, conclusions are given in Section V.

II. WIRELESS SENSOR NETWORK-BASED REAL-TIME VEHICLE NAVIGATION ALGORITHM

A. System Architecture

A WSN-based vehicle navigation system can be modeled as a directed graph \( G = (V, E) \) that consists of a set of intersection nodes \( V \) and a set of links \( E \) (a road exists between two intersections). The optimal navigation path from the source to the destination can be defined as a sequential list of directed links with the least cost, such as the TT and the TD.

Without loss of generality, given a sensor network that can transmit and receive signals, as shown in Fig. 1, we have the following two types of sensor nodes in our system: 1) the vehicle sensor nodes in the vehicles and 2) the intersection sensor nodes at the intersections. Each vehicle sensor node collects the host vehicle’s position, travel direction, and velocity data (obtained from different sensors such as the Global Positioning System and odometer installed in the vehicle [28]) and transmits them to the intersection sensor nodes. The intersection sensor nodes are responsible for collecting the information that comes from vehicle sensor nodes and computing the average vehicle velocity and the road density of each adjacent road. All the traffic information will be transmitted between sensor nodes using a simple hello protocol [29]. A two-layer hierarchical communication channel is established among sensor nodes, as shown in Fig. 2. One layer is the vehicle–intersection sensor network between the vehicle and intersection sensor nodes, and the other layer is the in-intersection sensor network among intersection sensor nodes.

B. Traffic Information

To achieve an efficient and robust routing operation, five kinds of traffic information are considered in our system as follows, and they are classified into the following two categories: 1) the benefit attributes (the-larger-the-better type) and 2) the cost attributes (the-smaller-the-better type).

- **Road distance**. It indicates the actual distance of a road. Navigation algorithms aim at achieving the least TD. It is a cost-type attribute.
- **Road width**. It denotes the number of lanes of a road. In this paper, we classify it into four priorities, i.e., 1, 2, 3, and 4, and the higher the lane status, the better the convenient travel condition. It is a benefit-type attribute.
- **Vehicle velocity**. It indicates the value of vehicle velocity on a road. It is important for the dynamic vehicle navigation, which aims at achieving the least TT. It is a benefit-type attribute.
- **Road density**. It stands for the number of vehicles per unit distance on a road. A high road density usually indicates the congested traffic, and vice versa. It is a cost-type attribute.
- **Navigation cost**. It refers to the expected cost (denoted as distance in this paper) for the driver to travel from an intersection position to the destination. This information can be obtained by an up–down broadcast protocol through the in-intersection sensor networks (the details are shown in the next sub-section). The cost is stored in the intersection sensor node. All intersection sensor nodes with navigation cost in the network form a cost field. When the vehicle approaches an intersection, it can obtain the navigation...
Among the five kinds of traffic information, the road distance and road width are fixed kinds that are initially kept in intersection sensor nodes, the vehicle velocity and road density are to be measured in real time, and the proposed navigation cost will be obtained through an up–down broadcast protocol at the beginning of navigation.

C. Real-Time Navigation Algorithm

This section gives a detailed description of the WSN-based vehicle navigation algorithm. The algorithm consists of four phases, as shown in Fig. 3.

1) Static Road Information Collecting Phase: The static road information, i.e., road distance and road width, should be loaded in the intersection sensor nodes in advance.

2) Navigation Cost Information Collecting Phase: In this phase, an up–down broadcast protocol, as shown in Fig. 4, is introduced to set up the navigation cost field. The protocol includes two steps. In the first step, a source vehicle’s navigation query will be transformed into an interest and transferred to the adjacent intersection sensor nodes. These intersection sensor nodes then flood the interest throughout the in-intersection sensor network toward the destination, as shown in Fig. 4(a), where the squared nodes represent vehicle sensor nodes, and the circles nodes are the intersection sensor nodes. A dashed circle represents the broadcasting area of a sensor. The arrows indicate the directions of information flow. When one intersection sensor node in the destination region matches the interest, it activates itself as the destination node $N_d$. The second step of the protocol, as shown in Fig. 4(b), is a reverse process of the first step. The destination node $N_d$ broadcasts an advertisement packet that announces a navigation cost of 0 ($\text{Cost}(N_d) = 0$). The rest of the intersection sensor nodes initially have a navigation cost of $\infty$ ($\text{Cost}(N_i) = \infty$ for $i \neq d$). Referring to Fig. 5, upon hearing an advertisement packet that contains the navigation cost from the sender $N_j$, the adjacent node $N_i$ calculates a new cost $\text{Cost}(N_i) + \text{LinkCost}(N_j, N_i)$, where $\text{LinkCost}(N_j, N_i)$ is the link distance between nodes $N_j$ and $N_i$. Node $N_i$ compares its current cost $\text{Cost}(N_i)$ with $\text{Cost}(N_j) + \text{LinkCost}(N_j, N_i)$ and chooses the smaller one as its new navigation cost. Then, node $N_i$ continues to broadcast an advertisement packet with the new cost toward the source vehicle. Eventually, every intersection sensor node may calculate the optimal navigation cost to the destination through flooding. The final navigation cost field is set up.

3) Real-Time Local Traffic Information Collecting Phase: The real-time local traffic information is sensed and collected in the vehicle–intersection sensor network in this phase, and these data will be used to guide the navigation decision making. Each intersection sensor node periodically broadcasts messages, including its ID and position. Normally, the vehicle sensor nodes are in the listening state. When a vehicle sensor node comes...
into the broadcast range of an intersection sensor node and receives its message, the vehicle sensor node will switch to the active state. Then, it collects the host vehicle’s location, travel direction, and velocity data and sends the information to the intersection sensor nodes nearby. The intersection sensor nodes receive the real-time road information from the vehicle sensor nodes around and compute the average vehicle velocity and the road density of each adjacent road.

4) Real-Time Navigation Decision-Making Phase: When the vehicle approaches an intersection, the vehicle sensor node can promptly obtain all the traffic information from the intersection sensor nodes nearby using a simple hello protocol. Multiple traffic attributes can be used to make a real-time navigation decision, which help the driver select one of the roads that are connected to the intersection (the details will be described in the following section). The phase will be repeated until the end of navigation.

In this algorithm, the static road information and the navigation cost information are obtained at the beginning of the navigation and will not be changed in the navigation process; only real-time traffic information at each adjacent road is needed to be collected, transmitted, and processed. Moreover, when the vehicle approaches an intersection, all the traffic information can rapidly be obtained from the intersection sensor nodes by using a simple hello protocol. As a result, less memory is needed compared with the global optimal navigation methods, and the real-time performance of navigation can be achieved.

III. REAL-TIME NAVIGATION DECISION-MAKING STRATEGY

When the vehicle approaches an intersection, it will select one of the adjacent intersection nodes as its direction of next move. Five kinds of traffic information, i.e., road distance, road width, vehicle velocity, road density, and navigation cost, are considered attributes, with which alternatives’ performance are measured. Then, the node selection or the navigation decision can be considered to be a MADM problem and concisely expressed in a decision matrix format $A = [a_{ij}]_{n \times m}$ as

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ a_2 & a_2 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_n & \cdots & a_n \end{bmatrix}_{n \times m}$$

where $a_1, a_2, \ldots, a_n$ are traffic attribute vectors for the $n$ possible adjacent intersection sensor nodes among which the decision maker should choose. Each intersection node has $m$ attributes, and $a_{ij} (1 \leq i \leq n, 1 \leq j \leq m)$ is the $j$th traffic attribute value of the $i$th intersection node.

The technique for order preference by similarity to ideal solution (TOPSIS) has been proven to be a useful technique in dealing with MADM problems in many applications [30]. It is based on the concept that the chosen alternative should have the shortest distance from the positive ideal solution and the farthest distance from the negative ideal solution. Unfortunately, the traditional TOPSIS method is used only for exact data. However, in vehicle navigation, not all of the traffic attributes involved in the decision problem should be denoted by exact numbers; some of these attributes are more suitable to be denoted by fuzzy numbers such as vehicle velocity and road density. A more realistic approach is to use hybrid MADM to tackle this problem [31]. However, one limitation of this hybrid method is that it uses different metrics for different attributes in calculating the distance of two attributes. In this paper, we further extend the concept of TOPSIS and develop a new method called Hybrid-TOPSIS, where a general distance metric is defined for processing both exact and fuzzy data. In the following sections, we start with some basic definitions. Then, the process of the Hybrid-TOPSIS method is presented.

A. Preliminaries

**Definition 1:** A fuzzy set $a$ in a universe of discourse $X$ is characterized by a membership function $\mu_a(x)$ that is associated with each element $x$ in $X$, a real number in the interval $[0, 1]$. The function value $\mu_a(x)$ is called the grade of the membership of $x$ in $a$ [32], [33].

In this paper, fuzzy data are denoted as a triangular fuzzy number $a$, which is a fuzzy set defined by a triplet $(e_1, e_2, e_3)$, as shown in Fig. 6. The membership function $\mu_a(x)$ is defined as (2) [34]

$$\mu_a(x) = \begin{cases} 0, & x \leq e_1 \\ \frac{x - e_1}{e_2 - e_1}, & e_1 < x \leq e_2 \\ \frac{e_3 - x}{e_3 - e_2}, & e_2 < x \leq e_3 \\ 0, & x > e_3 \end{cases}$$

Seven linguistic terms, such as low and high, can be expressed by triangular fuzzy numbers, as shown in Fig. 7.

**Definition 2:** A triangular fuzzy number $a$ can be defined as one case of an $L-R$ fuzzy number [34]. The left and
right parts of the membership function can be expressed as 
\( \mu_u^L(x) \) and \( \mu_u^R(x) \), respectively, and \( x_u^L(u) \) and \( x_u^R(u) \) are their corresponding inverse functions.

In the decision matrix \( A \), the attribute value may be an exact number or a triangular fuzzy number, and the corresponding distance equations are usually shown as (3) and (4), shown below, respectively. A general distance metric should be defined to unify the processing of exact and fuzzy data.

**Definition 3:** Let \( a_{ij} \) and \( a_{kj} \) be two exact attribute numbers. Then, the distance between them is defined as
\[
d_1(a_{ij}, a_{kj}) = c |a_{ij} - a_{kj}| \tag{3}
\]
where \( c \) is a constant that can be neglected for the optimal problem. We set \( c = 2 \) according to Theorem 2 in the Appendix.

**Definition 4:** Let \( a_{ij} \) and \( a_{kj} \) be two triangular fuzzy attribute numbers. Then, the distance between them [34] is defined as
\[
d_2(a_{ij}, a_{kj}) = \int_0^1 \left( |x_{a_{ij}}^L(u) - x_{a_{kj}}^L(u)| + |x_{a_{ij}}^R(u) - x_{a_{kj}}^R(u)| \right) du. \tag{4}
\]

**Definition 5:** Let \( a_{ij} \) be a general attribute number for the \( j \)th attribute of the \( i \)th alternative. Then, we denote it as a triplet \( a_{ij} = (a_{ij}^{(1)}, a_{ij}^{(2)}, a_{ij}^{(3)}) \), subject to \( a_{ij}^{(1)} \leq a_{ij}^{(2)} \leq a_{ij}^{(3)} \). If \( a_{ij} \) is an exact number, then \( a_{ij} = a_{ij}^{(1)} = a_{ij}^{(2)} = a_{ij}^{(3)} \), whereas if \( a_{ij} \) is a triangular fuzzy number, then \( a_{ij}^{(1)} = e_1 \), \( a_{ij}^{(2)} = e_2 \), and \( a_{ij}^{(3)} = e_3 \).

**Definition 6:** Let \( a_{ij} \) and \( a_{kj} \) be two general attribute numbers. Then, the general distance between them is defined as
\[
d(a_{ij}, a_{kj}) = \int_0^1 \left[ \left( a_{ij}^{(2)} - a_{ij}^{(1)} - (a_{kj}^{(2)} - a_{kj}^{(1)}) \right) \cdot u + \left( a_{ij}^{(1)} - a_{ij}^{(2)} \right) \right] du \nonumber
\]
\[
+ \int_0^1 \left[ \left( a_{ij}^{(2)} - a_{ij}^{(3)} - (a_{kj}^{(2)} - a_{kj}^{(3)}) \right) \cdot u + \left( a_{ij}^{(3)} - a_{ij}^{(2)} \right) \right] du. \tag{5}
\]

As aforementioned, \( d_1 \) is defined as the distance of two exact numbers, whereas \( d_2 \) is defined as the distance of two fuzzy numbers. In our algorithm, we define \( d \) to represent the distance in either case. The distance \( d \) is a generalization of \( d_1 \) and \( d_2 \). The proof of the validity and generality of \( d \) can be found in Theorem 1 and Theorem 2 in the Appendix. In the following Hybrid-TOPSIS decision scheme, \( d \) is used as the general distance metric of two traffic attributes.

**Definition 7:** Let \( w \) be numerical data. Then, \( a_{ij} \) is a general attribute number, and the basic operation \( w \cdot a_{ij} \) is defined as
\[
w \cdot a_{ij} = \left( w \cdot a_{ij}^{(1)}, w \cdot a_{ij}^{(2)}, w \cdot a_{ij}^{(3)} \right). \tag{6}
\]

**B. Hybrid-TOPSIS Decision Scheme**

The process of the Hybrid-TOPSIS decision scheme is shown as follows.

1. Normalize the decision matrix \( A = [a_{ij}]_{n \times m} = [(a_{ij}^{(1)}, a_{ij}^{(2)}, a_{ij}^{(3)})]_{n \times m} \) to get the normalized decision matrix \( R = [r_{ij}]_{n \times m} = [(r_{ij}^{(1)}, r_{ij}^{(2)}, r_{ij}^{(3)})]_{n \times m} \).

   Normalization seeks to obtain comparable scales, which is a linear transformation, as defined in
\[
r_{ij} = \left( r_{ij}^{(1)}, r_{ij}^{(2)}, r_{ij}^{(3)} \right) = \left( a_{ij}^{(1)} - a_{ij}^{(2)}, a_{ij}^{(2)} - a_{ij}^{(3)}, a_{ij}^{(3)} \right), \quad j \in S_1
\]
\[
r_{ij} = \left( r_{ij}^{(1)}, r_{ij}^{(2)}, r_{ij}^{(3)} \right) = \left( a_{ij}^{(1)} - a_{ij}^{(2)}, a_{ij}^{(2)} - a_{ij}^{(3)}, a_{ij}^{(3)} \right), \quad j \in S_2 \tag{7}
\]
where \( a_{ij}^{(3)} = \max \{ a_{ij}^{(1)} \}, a_{ij}^{(1)} = \min \{ a_{ij}^{(1)} \}, S_1 \) is the set of benefit attributes (the-larger-the-better type), \( S_2 \) is the set of cost attributes (the-smaller-the-better type), and \( 1 \leq i \leq n, 1 \leq j \leq m \).

2. Calculate the weight vector \( w = [w_1, w_2, \ldots, w_m] \) for each attribute based on the maximizing deviation method [35].

   Let \( w' = [w'_1, w'_2, \ldots, w'_m] \) be the initial weight vector to the attribute set. Referring to [35], we should choose the weight vector \( w' \) to maximize all deviation values for all the attributes. To do so, we can construct a nonlinear programming model as follows:
\[
\max F(w') = \sum_{j=1}^m \sum_{i=1}^n w'_j d(r_{ij}, r_{kj})
\]
subject to
\[
\sum_{j=1}^m w'_j = 1, \quad 0 \leq w'_j \leq 1. \tag{8}
\]

To solve the aforementioned model, let
\[
G(w', \xi) = \sum_{j=1}^m \sum_{i=1}^n w'_j d(r_{ij}, r_{kj}) - \frac{1}{2} \xi \left( \sum_{j=1}^m w'_j^2 - 1 \right) \tag{9}
\]
denote the Lagrange function of the constrained optimization problem, where \( \xi \) is a Lagrange multiplier. Then, the partial derivatives of \( G \) are computed as
\[
\frac{\partial G(w')}{\partial w'_j} = \sum_{i=1}^n \sum_{k=1}^m d(r_{ij}, r_{kj}) + \xi w'_j = 0, \quad \text{for } 1 \leq j \leq m \tag{10}
\]
\[
\frac{\partial G(w')}{\partial \xi} = - \frac{1}{2} \left( \sum_{j=1}^m w'_j^2 - 1 \right) = 0. \tag{11}
\]

Based on (10), we can derive that
\[
w'_j = \frac{\sum_{i=1}^n \sum_{k=1}^m d(r_{ij}, r_{kj})}{\xi}, \quad \text{for } 1 \leq j \leq m. \tag{12}
\]
Putting (12) into (11), we have
\[
\sum_{j=1}^{m} \left( \frac{\sum_{i=1}^{n} \sum_{k=1}^{n} d(r_{ij}, r_{kj})}{\xi} \right)^2 = 1 \quad (13)
\]
\[
\xi = \sqrt{\sum_{j=1}^{m} \left( \frac{\sum_{i=1}^{n} \sum_{k=1}^{n} d(r_{ij}, r_{kj})}{\sum_{i=1}^{n} \sum_{k=1}^{n} d(r_{ij}, r_{kj})} \right)^2} \quad (14)
\]
Putting (14) into (12), we have
\[
w_j' = \frac{\sum_{i=1}^{n} \sum_{k=1}^{n} d(r_{ij}, r_{kj})}{\sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{n} d(r_{ij}, r_{kj})} \quad \text{for } 1 \leq j \leq m.
\]

The final \( w_j \) is derived from the normalization of \( w_j' \) as
\[
w_j = \frac{w_j'}{\sum_{j=1}^{m} w_j'} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{n} d(r_{ij}, r_{kj})}{\sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{n} d(r_{ij}, r_{kj})}, \quad \text{for } 1 \leq j \leq m.
\]

3) Construct the weighted normalized decision matrix \( Z = [z_{ij}]_{n \times m} \) as (17) and identify positive ideal \( (z^*) \) and negative ideal \( (z^-) \) solutions as
\[
z_{ij} = w_j r_{ij} = \left( w_{ij}^{(1)}, w_{ij}^{(2)}, w_{ij}^{(3)} \right) \quad (17)
\]
\[
z^* = [z^*_1, z^*_2, \ldots, z^*_m], \quad z^- = [z^-_1, z^-_2, \ldots, z^-_m] \quad (18)
\]
where \( z^*_i = (1, 1, 1), z^-_i = (0, 0, 0), 1 \leq i \leq n, 1 \leq j \leq m. \)

4) Obtain the \( m \)-D Euclidean distance of each intersection node to \( z^* \) and \( z^- \) as \( s^* \) and \( s^- \), i.e.,
\[
s^*_i = \sqrt{\sum_{j=1}^{m} d(z_{ij}, z^*_j)^2}, \quad \text{for } 1 \leq i \leq n \quad (19)
\]
\[
s^-_i = \sqrt{\sum_{j=1}^{m} d(z_{ij}, z^-_j)^2}, \quad \text{for } 1 \leq i \leq n. \quad (20)
\]

5) For each intersection node, calculate the closeness \( c_i \) according to
\[
c_i = \frac{s^-_i}{s^-_i + s^*_i}, \quad \text{for } 1 \leq i \leq n. \quad (21)
\]

We rank all the adjacent intersection nodes according to their closeness \( c_i \). The node with the maximum \( c_i \) is selected as the best node for navigation.

Note that the proposed navigation decision-making scheme can be used for reaching different navigation goals. Different attribute combinations in a decision matrix can provide different navigation paths to meet diverse navigation requirements of drivers. For example, if we choose the road distance and navigation cost as the traffic features in the decision-making

scheme, the goal of the least TD can be achieved. Similarly, taking vehicle velocity, road density, and navigation cost as the traffic features will lead to the least TT. The most effective and compromised navigation route can be obtained by using multiple attributes for decision making. The simulation results of different navigation goals can be found in the following section.

IV. SIMULATION RESULTS
To demonstrate the suitability and efficiency of the proposed algorithm, an \( 8 \times 8 \) grid network followed by two real-world urban road networks from Beijing and New York are employed for simulation. Simulation of Urban Mobility (SUMO) [36], a suite of microscopic and continuous road traffic simulation software tools developed by the German Aerospace Center, is used to model the movement and behavior of vehicles in traffic networks and helps us obtain dynamic traffic data that are similar to those in the real world. The following four navigation requirements are discussed:

- least TD;
- least TT;
- best lane status (the maximum road width);
- comprehensive consideration of multiple attributes.

The following three performance metrics are introduced for comparative analysis: 1) TD; 2) TT; and 3) the mean of velocity (MV).

A well-designed \( 8 \times 8 \) grid network, as shown in Fig. 8, is first given as an example to show the validity of the proposed algorithm. In the figure, an intersection node is represented by a solid circle, and the node number is displayed nearby. A link between two intersection nodes is denoted by a black line, and its length indicates the road distance. All links can be traveled in either direction, leading to 224 directional paths. To show the effects of different traffic conditions on each navigation requirement, some parameters are provided as follows. The length of the horizontal link is 200 m, and the length of the vertical link is 100 m. The number of lanes is set to 4 for the roads on (9–16) and (2–58). Here, \( (a-b) \) denotes the line from node \( a \) to node \( b \). The number of lanes for other roads is set to 1. The traffic on (1–8) and (9–16) is set as congested, whereas the
traffic on (17–24) and (25–32) is set as relatively crowded. The maximum velocity on each road is 40 km/h.

At first, the traffic lights are not considered in the simulation. When the start of navigation is set at node 1 and the destination is set at node 8, four navigation paths are marked with different colors, and the marks are shown in Fig. 8. The corresponding navigation performances are listed in Table I. Without considering road traffic, path 1 (the carmine line) can achieve the minimum TD (1400 m), but with the maximum TT (282.11 s) and the lowest MV (17.87 km/h). On the contrary, path 2 (the green line) chooses the smoothest road traffic, resulting in the shortest TT (198.02 s) and the highest MV (40.00 km/h). However, it has the longest TD (2200 m). Giving the highest priority to the road width, path 3 (the blue line) chooses the route with more lanes. The balanced results (TD of 1800 m, TT of 213.27 s, and MV of 30.38 km/h) can be obtained by comprehensively considering multiple traffic attributes for navigation decision making. The resulting path is highlighted by the red line in Fig. 8. For comparison, we also apply two well-known algorithms (the Dijkstra algorithm [3] and the dynamic A* algorithm [12]) to search the optimal routes. As shown in Table I, the result of the Dijkstra algorithm is same as path 1, and the result of the dynamic A* algorithm is same as path 2. In conclusion, our algorithm can not only obtain an effective and compromised route by considering multiple traffic information to make navigation decision but select a route with the least TD or the least TT similar to traditional navigation approaches as well.

To consider the effects of traffic lights on the navigation algorithm, we then add them at intersections in the simulation. The time delay of the red signal is set as a random number that is distributed within 0 ～ 20 s. In the simulation, the navigation algorithm is performed for ten times. We use the mean of travel time (MTT) as the performance metric instead of TT due to the randomness of the red signal. The simulation results are shown in Table II. Because only local traffic information is utilized in our algorithm to make navigation decisions, the traffic lights and their phase settings have little impact on the selection of navigation paths, except for prolonging the TT of each selected route. This has also been reflected in the simulation result.

In this paper, two real-world urban road networks from Beijing and New York, as shown in Figs. 9 and 10, are employed in the simulations. They are both obtained from http://www.openstreetmap.org/. Figs. 11 and 12 illustrate the corresponding traffic networks for simulations. The traffic network of Beijing contains 115 nodes and 321 links, whereas the traffic network of New York includes 154 nodes and 274 links. The coordinate of each node and the length of each link in the traffic networks are in accordance with real map data. The gray parts in Figs. 11 and 12 indicate the heavy traffic situations in these areas. Here, dark gray denotes the crowded traffic, and light gray stands for the relatively crowded condition. In particular, because the roads of New York are mostly one-way streets, black arrows are added to indicate the travel directions of roads in Fig. 12. In these two examples, SUMO is also used to simulate the traffic conditions, and the maximum velocity

---

**Table I**

<table>
<thead>
<tr>
<th>Paths</th>
<th>Requirements</th>
<th>TD (m)</th>
<th>TT (s)</th>
<th>MV (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel Distance (as Dijkstra algorithm [3])</td>
<td>1400.00</td>
<td>282.11</td>
<td>17.87</td>
</tr>
<tr>
<td>2</td>
<td>Travel Time (as dynamic A* algorithm [12])</td>
<td>2200.00</td>
<td>198.02</td>
<td>40.00</td>
</tr>
<tr>
<td>3</td>
<td>Lane Status</td>
<td>1600.00</td>
<td>318.38</td>
<td>18.09</td>
</tr>
<tr>
<td>4</td>
<td>Multi-attribute</td>
<td>1800.00</td>
<td>213.27</td>
<td>30.38</td>
</tr>
</tbody>
</table>

**Table II**

<table>
<thead>
<tr>
<th>Paths</th>
<th>Requirements</th>
<th>TD (m)</th>
<th>MTT (s)</th>
<th>MV (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel Distance</td>
<td>1400.00</td>
<td>316.11</td>
<td>15.94</td>
</tr>
<tr>
<td>2</td>
<td>Travel Time</td>
<td>2200.00</td>
<td>275.02</td>
<td>28.80</td>
</tr>
<tr>
<td>3</td>
<td>Lane Status</td>
<td>1600.00</td>
<td>357.38</td>
<td>16.12</td>
</tr>
<tr>
<td>4</td>
<td>Multi-attribute</td>
<td>1800.00</td>
<td>283.27</td>
<td>22.88</td>
</tr>
</tbody>
</table>
of roads is also set as 40 km/h. The source and destination of the navigation are marked with a solid circle and a triangle, respectively. The navigation results are listed in Tables III–VI. The results shown in Tables III and V are very similar to the results of the 8 × 8 grid network when the traffic light information is not included. One interesting finding is that, in Table VI, the least TT path may no longer be the unique path, because the randomness of the real-time traffic lights information is included at the intersections. This result demonstrates the benefits of our multiple-criteria decision-making-based navigation algorithm.

### TABLE III

**Statistics of Different Navigation Requirements on the Road Network of Beijing**

<table>
<thead>
<tr>
<th>Paths</th>
<th>Requirements</th>
<th>TD (m)</th>
<th>TT (s)</th>
<th>MV (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel Distance</td>
<td>663.53</td>
<td>145.20</td>
<td>16.45</td>
</tr>
<tr>
<td>2</td>
<td>Travel Time</td>
<td>1015.69</td>
<td>133.42</td>
<td>27.41</td>
</tr>
<tr>
<td>3</td>
<td>Lane Status</td>
<td>1152.31</td>
<td>232.61</td>
<td>17.83</td>
</tr>
<tr>
<td>4</td>
<td>Multi-attribute</td>
<td>840.93</td>
<td>138.52</td>
<td>21.85</td>
</tr>
</tbody>
</table>

### TABLE IV

**Statistics of Different Navigation Requirements on the Road Network of Beijing with Traffic Lights**

<table>
<thead>
<tr>
<th>Paths</th>
<th>Requirements</th>
<th>TD (m)</th>
<th>TT (s)</th>
<th>MV (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel Distance</td>
<td>1229.14</td>
<td>211.19</td>
<td>20.95</td>
</tr>
<tr>
<td>2</td>
<td>Travel Time</td>
<td>1694.72</td>
<td>152.54</td>
<td>40.00</td>
</tr>
<tr>
<td>3</td>
<td>Lane Status</td>
<td>1272.86</td>
<td>211.09</td>
<td>21.71</td>
</tr>
<tr>
<td>4</td>
<td>Multi-attribute</td>
<td>1368.83</td>
<td>164.33</td>
<td>29.99</td>
</tr>
</tbody>
</table>

### TABLE V

**Statistics of Different Navigation Requirements on the Road Network of New York**

<table>
<thead>
<tr>
<th>Paths</th>
<th>Requirements</th>
<th>TD (m)</th>
<th>TT (s)</th>
<th>MV (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel Distance</td>
<td>1229.14</td>
<td>267.19</td>
<td>16.56</td>
</tr>
<tr>
<td>2</td>
<td>Travel Time</td>
<td>1694.72</td>
<td>223.54</td>
<td>27.29</td>
</tr>
<tr>
<td>3</td>
<td>Lane Status</td>
<td>1272.86</td>
<td>256.09</td>
<td>17.89</td>
</tr>
<tr>
<td>4</td>
<td>Multi-attribute</td>
<td>1368.83</td>
<td>219.33</td>
<td>22.47</td>
</tr>
</tbody>
</table>

### TABLE VI

**Statistics of Different Navigation Requirements on the Road Network of New York with Traffic Lights**

<table>
<thead>
<tr>
<th>Paths</th>
<th>Requirements</th>
<th>TD (m)</th>
<th>TT (s)</th>
<th>MV (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel Distance</td>
<td>1229.14</td>
<td>267.19</td>
<td>16.56</td>
</tr>
<tr>
<td>2</td>
<td>Travel Time</td>
<td>1694.72</td>
<td>223.54</td>
<td>27.29</td>
</tr>
<tr>
<td>3</td>
<td>Lane Status</td>
<td>1272.86</td>
<td>256.09</td>
<td>17.89</td>
</tr>
<tr>
<td>4</td>
<td>Multi-attribute</td>
<td>1368.83</td>
<td>219.33</td>
<td>22.47</td>
</tr>
</tbody>
</table>

### V. Conclusion

This paper has developed a novel WSN-based real-time vehicle navigation algorithm, in which the two-layer sensor network architecture is introduced, five kinds of traffic attributes are defined, and four phases of the proposed algorithm are discussed. The hybrid MADM method has been presented for real-time navigation decision making, where a new general distance metric has been proposed for the processing of both exact and fuzzy data. Based on multiple traffic information and navigation requirements, including the least TD, the least TT, and the best lane status, the proposed algorithm selects the most effective and compromised navigation route. Simulation results demonstrate the good navigation performance of the proposed algorithm in the WSN-based real-time traffic networks.

### Appendix

**Theorem 1:** For any general attribute numbers $a_{ij}$, $a_{kj}$, and $a_{lj}$, the general distance (5) satisfies the following conditions.

1) $a_{ij} = a_{kj} \Leftrightarrow d(a_{ij}, a_{kj}) = 0$.
2) $d(a_{ij}, a_{kj}) = d(a_{kj}, a_{ij})$.
3) $d(a_{ij}, a_{kj}) \geq d(a_{kj}, a_{lj}) \geq d(a_{ij}, a_{lj})$. 

---

**Fig. 11.** Different navigation paths on the road network of Beijing.

**Fig. 12.** Different navigation paths on the road network of New York.
Proof: We have the following cases.

1) Let \( a_{ij} \) and \( a_{kj} \) be two general attribute numbers; then

If \( a_{ij} = a_{kj} \), we have \( a_{ij}^{(1)} = a_{kj}^{(1)} \), \( a_{ij}^{(2)} = a_{kj}^{(2)} \), and \( a_{ij}^{(3)} = a_{kj}^{(3)} \); therefore, we have \( d(a_{ij}, a_{kj}) = 0 \).

If \( d(a_{ij}, a_{kj}) = 0 \), we have

\[
1 \int \left| \left( a_{ij}^{(1)} - a_{ij}^{(2)} - a_{kj}^{(1)} - a_{kj}^{(2)} \right) \right| du = 0
\]

\[
1 \int \left| \left( a_{ij}^{(1)} - a_{ij}^{(3)} - a_{kj}^{(1)} - a_{kj}^{(3)} \right) \right| du = 0. \tag{22}
\]

As \( 0 \leq u \leq 1 \), we have

\[
\left( a_{ij}^{(2)} - a_{ij}^{(1)} - a_{kj}^{(2)} - a_{kj}^{(1)} \right) = 0
\]

\[
\left( a_{ij}^{(2)} - a_{ij}^{(3)} - a_{kj}^{(2)} - a_{kj}^{(3)} \right) = 0. \tag{23}
\]

for \( \forall u \in \{0, 1\} \).

When we let \( u = 0 \) and \( u = 1 \), we have \( a_{ij}^{(1)} = a_{ij}^{(2)} = a_{ij}^{(3)} \); therefore, we have \( a_{ij} = a_{kj} \).

2) Let \( a_{ij} \) and \( a_{kj} \) be two general attribute numbers. Then, we have

\[
d(a_{ij}, a_{kj}) = 1 \int \left| \left( a_{ij}^{(2)} - a_{ij}^{(1)} - a_{kj}^{(2)} - a_{kj}^{(1)} \right) \right| du + 1 \int \left| \left( a_{ij}^{(3)} - a_{ij}^{(2)} - a_{kj}^{(3)} - a_{kj}^{(2)} \right) \right| du
\]

\[
= \left| a_{ij}^{(2)} - a_{kj}^{(2)} \right| + \left| a_{ij}^{(3)} - a_{kj}^{(3)} \right| \tag{24}
\]

3) Let \( a_{ij} \), \( a_{kj} \), and \( a_{ij} \) be general attribute numbers. Then, we have

\[
d(a_{ij}, a_{kj}) + d(a_{kj}, a_{ij}) = 1 \int \left| \left( a_{ij}^{(2)} - a_{ij}^{(1)} - a_{kj}^{(2)} - a_{kj}^{(1)} \right) \right| du + 1 \int \left| \left( a_{ij}^{(3)} - a_{ij}^{(2)} - a_{kj}^{(3)} - a_{kj}^{(2)} \right) \right| du
\]

\[
= 1 \int \left| \left( a_{ij}^{(2)} - a_{ij}^{(1)} - a_{kj}^{(2)} - a_{kj}^{(1)} \right) \right| du + 1 \int \left| \left( a_{ij}^{(3)} - a_{ij}^{(2)} - a_{kj}^{(3)} - a_{kj}^{(2)} \right) \right| du
\]

\[
= d(a_{ij}, a_{kj}). \tag{25}
\]

This completes the proof.

\[\blacksquare\]

Theorem 2: Equation (5) is the general form of (3) and (4).

Proof: If \( a_{ij} \) and \( a_{kj} \) are two exact attribute numbers, we have \( a_{ij} = a_{ij}^{(1)} = a_{ij}^{(2)} = a_{ij}^{(3)} \) and \( a_{kj} = a_{kj}^{(1)} = a_{kj}^{(2)} = a_{kj}^{(3)} \). Therefore, we have

\[
d(a_{ij}, a_{kj}) = \left( a_{ij}^{(1)} - a_{kj}^{(1)} \right) = 0
\]

\[
d(a_{ij}, a_{kj}) = \left( a_{ij}^{(3)} - a_{kj}^{(3)} \right) = 0. \tag{26}
\]

If \( a_{ij} \) and \( a_{kj} \) are two triangular fuzzy attribute numbers and we let \( x_{a_{ij}}^{L}(u) \) and \( x_{a_{kj}}^{R}(u) \) be membership function of \( a_{ij} \) and \( x_{a_{ij}}^{L}(u) \) and \( x_{a_{kj}}^{R}(u) \) are membership function of \( a_{kj} \), and we have

\[
x_{a_{ij}}^{L}(u) = a_{ij} \]

\[
x_{a_{ij}}^{R}(u) = a_{ij} \]

\[
x_{a_{ij}}^{L}(u) = a_{ij} \]

\[
x_{a_{ij}}^{R}(u) = a_{ij} \]

Then, we have

\[
d(a_{ij}, a_{kj}) = \int \left( x_{a_{ij}}^{L}(u) - x_{a_{kj}}^{L}(u) \right) du + \int \left( x_{a_{ij}}^{R}(u) - x_{a_{kj}}^{R}(u) \right) du
\]

\[
= d_{2}(a_{ij}, a_{kj}). \tag{28}
\]

This completes the proof.

\[\blacksquare\]

REFERENCES


