IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 December 2023, accepted 6 February 2024, date of publication 13 February 2024, date of current version 20 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3365513

RESEARCH ARTICLE

A New Framework for Centralized Coordinated Multi-Vehicle Dynamic Routing

M. D. R. L. SILVA[1,2] **AND M. TANG**[1]
[1]School of Computer Science, Queensland University of Technology, Brisbane, QLD 4001, Australia
[2]Department of Computer Science, University of Sri Jayewardenepura, Nugegoda 10250, Sri Lanka

Corresponding author: M. Tang (m.tang@qut.edu.au)

ABSTRACT In the context of rapid urbanization, the problem of long travel times has become a significant problem for contemporary cities, both socially and economically. Although dynamic route guidance has emerged as a promising solution to minimize travel times in large road networks, existing route planning frameworks often lead to congestion on certain routes, as vehicles with the same travel itinerary tend to follow the same route due to lack of central coordination. To address this challenge, this paper introduces a novel framework for centrally coordinating all vehicles on a road network. The proposed framework aims to optimize the average travel time of all vehicles while considering the fairness of all vehicles. The effectiveness of this framework has been evaluated through simulations and compared with three popular benchmark frameworks using a well-known traffic scenario and a real-world traffic scenario. The experimental results have shown that this framework outperforms the benchmark frameworks.

INDEX TERMS Intelligent transportation system, centralized traffic assignment, dynamic route guidance, coordinated multi-vehicle routing, constrained combinatorial optimization.

I. INTRODUCTION

Travel time is becoming longer and longer in the modern world, driven primarily by the rapid increase in the number of travelers. This undesirable trend of growing traffic causes a huge waste of time, fuel, and productivity. Such losses can have negative effects on the economy of a country and can significantly impact the overall quality of life. According to the TomTom traffic index [1], in 2022 the average travel time per 10 km in the two largest cities, Sydney and Melbourne, has increased to 21 minutes and 30 seconds and 20 minutes and 30seconds, respectively.

Vehicle route can become challenging during special events, such as sporting or musical events, when a large number of vehicles exit a venue at the same time. In such a situation, it is common for vehicles to look for the shortest or quickest route between a common starting point (the venue of the event) and their destinations. As a result, all vehicles that have the same destination will use the same route, which can cause traffic congestion on the route, while some alternative routes are not used. This is because all vehicles are routed independently since there is no coordination between them. Addressing this problem requires a centrally coordinated approach that can take into account real-time traffic information. Therefore, the motivation behind this research work lies in introducing a new framework to provide a well-coordinated and guided route suggestion mechanism considering real-time traffic information.

Coordinating the route of multiple vehicles can be classified into *centralized coordination* and *distributed coordination*. In centralized coordination, the central server is responsible for acquiring information from the road network and generating route suggestions for all vehicles in the road network. For many years, prominent criticism of centralized systems has been about their limited scalability considering the high computational load, the high information exchange load, and the high delay with respect to large road networks. However, these limitations are being overcome by the tremendous growth shown in the area of cloud computing [2]. Recent surveys conducted by Kirimtat et al. [3] and Talebkhah et al. [4], further demonstrate how different data management technologies, such as IoT and cloud

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Xiao.

computing, can be used to process a large amount of data more efficiently. Furthermore, with the enhanced capabilities of the 5G network, it is expected to overcome the long delays experienced in getting a response from the central system [5].

In contrast, decentralized coordination requires that the individual user makes all route guidance decisions based on the information acquired from traffic management systems. However, the traffic management system has no control over route selection and the user is responsible for selection. Compared to centralized coordination, distributed coordination requires less computational power. However, since in distributed coordination, a vehicle communicates only with nearby vehicles, the knowledge of current traffic in the road network is incomplete, and this incompleteness may be reflected in the routing suggestions [6]. On the contrary, centralized coordination is capable of providing more reliable routing suggestions, as it can see the near future traffic congestion due to global knowledge of the traffic network. As a result, by properly analyzing traffic demand, it is possible to predict future traffic conditions that can then be used to optimize the vehicles' routes in the road network.

Motivated by the promising strengths of centralized computational feasibility and communication advancements, this paper introduces a new centralized coordinated multi-vehicle dynamic routing framework. This framework proactively optimizes traffic from the system's perspective by minimizing the average travel time of all vehicles while considering fairness in individual travel times for vehicles with similar itineraries. Differentiating from approaches such as [7], [8], [9], and [11], which propose route guidance systems to balance the travel time of the system and individual travel times, the proposed method adopts real-time traffic data and uses traffic-dependent travel times to limit the difference in individual experienced travel times, considering fairness among vehicles with similar itineraries.

The work presented in this paper makes multiple contributions to solving the assignment of routes in modern cities. The main contribution of this work is a novel centralized coordinated multi-vehicle dynamic routing framework. Another contribution of this this work is the way to consider the fairness of all vehicles in a road network.

The remainder of the paper is structured as follows. Section II discusses related work. Section III introduces the proposed centralized coordinated multi-vehicle dynamic routing framework. Section IV gives the details of the experimental design followed by the experimental results in Section V. The article ends with conclusions and potential future work in Section VI.

II. RELATED WORK

For many years, the Vehicle Route Guidance System (RGS) has been investigated by numerous researchers in terms of optimal route assignment. Almost a great deal of previous research in vehicle RGS has focused on static RGS, or SRGS. These SRGS are based on algorithms such as Dijkstra [12], Bellman-Ford [13], and A* search [14], which are the optimal algorithms to find the shortest path in a graph. These algorithms are successfully used in SRGS, but they are suitable for use in dynamic RGS, or DRGS.

More recent attention has been paid to dynamic RGS, or DRGS, due to their ability to react to changes in traffic in road networks [15], [16], [17]. However, most commercial DRGS, such as Google, Microsoft, and TomTom DRGS, are reactive systems that cannot actively prevent traffic congestion [11], [20], [24]. This is because a reactive DRGS generates the optimal path independently for each of the vehicles in the road network based on current traffic information. Therefore, a reactive DRGS can cause congestion to switch between roads back and forth when a significant number of vehicles select the same route known as the "route flapping" [18]. Alternatively, many proactive DRGS [18], [19], [20], [21] are proposed in which optimal routes are suggested considering both historical traffic information and real-time traffic information.

In the literature, much research on DRGS has focused on user perspective guidance [17], [22], [23]. However, user-perspective DRGS still suffer from developing traffic congestion due to the selfish nature of route selection. On the contrary, the system perspective route guide focuses on reducing the total travel time for all vehicles in a road network using [24], [25], [26], [27], and [28]. The system-optimal DRGS reduce the average travel time of those vehicles in a road network by overlooking the traffic distribution in the road network.

To satisfy the user and system performance, coordination among vehicles in a road network is necessary. In 2019, Zhu et al. [29] proposed an edge-assisted distributed routing framework, which enables virtual agents on behalf of vehicles to interact with others to make real-time routing decisions. A similar distributed approach was adopted in [24] where traffic distribution optimization was carried out locally by each car based on local knowledge collected from nearby vehicles. A decentralized approach to anticipatory vehicle routing that was particularly useful in a large-scale dynamic environment was presented in [30]. Another look at the decentralized system can be found in [31]. In this work, the cooperative vehicle routing problem is considered at intersections of the road network. Few more studies on distributed systems can be found in [32] and [33].

On the other hand, centralized DRGS uses a central system to provide route suggestions. This method provides more robust and reliable solutions at the overall network level. In 2014, a centralized proactive DRGS was introduced to provide driver route guidance when traffic congestion was predicted [19]. Furthermore, the simulation results showed that proactive route guidance leads to lower reroutes for each rerouted vehicle. However, their work could not optimize traffic from the point of view of the system.

In the work presented in [34], a cooperative centralized approach is discussed to determine the routes of all vehicles, thus improving the efficiency of the entire traffic system. However, during simulations, the proposed method could not find good solutions in the case of a large number of vehicles due to its limitation in scalability. Research carried out in [18] presents five reroute strategies to compute alternative routes for vehicles. The introduced rerouting strategies proactively push individually tailored routes to vehicles when there are signs of congestion. Similar work can also be found in [35], which presents a participatory navigation system. Data, such as location samples and route choice decisions, were collected from road vehicles to estimate traffic speed and future traffic flow at the level of the road segment. However, both of these approaches do not concern themselves with a globally optimal solution, but instead try to optimize the local user's route choice based on the information collected at the given time.

In recent literature, there is a trend towards integrating system optimal approaches with user optimal approaches to achieve system optimal routes while considering fairness among users. In 2016, Angelelli et al. [7] introduced a linear programming-based approach to balance the trade-off between the system and user perspectives in proactive route guidance. This approach assigns paths to users, ensuring a certain level of fairness among vehicles with similar starting and ending locations. The method initially sets a limit on user travel inconvenience, referred to as maximum travel inconvenience, in terms of the maximum increase allowed relative to the shortest path. Subsequently, the method generates a set of eligible paths for each OD pair, which comprises Origin-Destination paths with a travel inconvenience that does not exceed the specified maximum travel inconvenience. However, drawbacks include the initial generation of eligible paths and the limitation of maximum user travel inconvenience relative to the shortest path, without incorporating real-time traffic information. This approach lacks dynamic generation of eligible paths, which is crucial for suggesting paths to vehicles based on real-time traffic conditions. Another limitation is the omission of traffic-dependent arc travel times, which prevents accurate modeling of travel times along paths containing one or more congested arcs. The authors expanded on their research in 2018 by introducing a heuristic algorithm approach to decrease the computational time required to generate eligible paths [8]. Subsequently, in 2020, they proposed a linear programming model incorporating a traffic-dependent latency function [10]. However, in each of these approaches, the observed unfairness in terms of travel times on the restricted path set may exceed the specified level of unfairness due to the prior generation of eligible paths. This inherent drawback was overcome by the authors in their research carried out in 2020 [9]. The proposed approach embedded path selection in the optimal formulation of the system to control the unfairness experienced by travelers. The most recent research by Ho et al. in 2023 [11] proposed adopting a road segment capacity-aware routing approach that can effectively facilitate collaboration among vehicles to optimize both system and individual performances. The proposed approach updates the remaining capacity of relevant road segments each time a vehicle updates its path to allow collaboration among vehicles. However, only a selected number of vehicles are dynamically rerouted by evaluating the distance between the current position of the vehicle and the congested road. Moreover, while the adaptation of the kSP algorithm appears useful in finding multiple alternative paths, including the best path, there might be scenarios where the algorithm does not guarantee the absolute best path in terms of travel time or other metrics. This is particularly crucial in dynamic routing, where conditions can change rapidly. Furthermore, the performance of kSP algorithms can be sensitive to the choice of parameters, such as the value of 'k' (the number of shortest paths to be computed), and selecting an inappropriate value for 'k' may lead to suboptimal results. Therefore, in contrast to the aforementioned approaches in the literature, the proposed approach in this paper computes a route plan for all vehicles in the traffic network. The method generates two sets of routes to choose the best set of paths based on current traffic conditions. Furthermore, it incorporates the remaining capacity of the road segments each time a new routing plan is generated, ensuring collaboration between vehicles traveling on the network.

In the literature, out of many extensions proposed to improve the efficiency of coordinated centralized multi-vehicle dynamic routing problem, dynamic generation of candidate routes for each vehicle by incorporating real-time traffic data to calculate the road segment travel times, incorporating explicit measurements to ensure individual travel time fairness among vehicles with same travel needs, and consideration of capacity of the each road segment when generating routes to ensure that the road segments are not overutilized can be highlighted as essential extensions to be considered in proposing an efficient coordinated centralized multi-vehicle dynamic routing algorithm. Furthermore, it is evident that most of the research carried out on multi-vehicle route guidance focused on decentralized approaches considering scalability and computational limitations. However, with recent advances in high-performance computing facilities, such as cloud computing, and more advanced communication technologies, such as 5G /6G, centrally coordinated route guidance is feasible. Therefore, to bridge the aforementioned gaps, in this paper, we will propose a centrally coordinated multi-vehicle dynamic route guidance framework that adopts the traffic assignment to the dynamic nature of traffic considering all individual drivers' experience while optimizing the traffic from system perspective.

III. THE CENTRALIZED COORDINATED MULTI-VEHICLE DYNAMIC ROUTING FRAMEWORK

This section presents a new centralized coordinated multi-vehicle dynamic routing framework. In this section, we will introduce the architecture of the framework and the components of the framework and then discuss how all vehicles in a road network are dynamically routed in a centralized and coordinated way.

A. ARCHITECTURE OF THE FRAMEWORK

As depicted in Fig. 1, our framework comprises two components: *candidate routes generator* and *coordinated multi-vehicle dynamic route generator*. The candidate routes generator is a one-time process that is tasked with generating a list of feasible potential routes between each pair of predefined locations in the road network. The coordinated multi-vehicle route generator operates periodically to produce a new optimized route plan for all vehicles in the road network. This periodic operation is essential due to the dynamic nature of traffic conditions that change over time in the road network.

1) CANDIDATE ROUTES GENERATOR

In a road network, a road can be divided into several segments, known as *road segments*. Each end of a road segment is termed a *predefined location* or simply a *location* within this framework. The candidate route generator is used to create a set of candidate routes between every pair of these locations.

In a vast road network, there may be numerous routes between a pair of locations. However, not all routes are considered candidate routes due to their excessive travel time. Consequently, these unsuitable routes are excluded during the route generation process. This candidate route generator has multiple advantages. First, it ensures that the framework avoids generating an all-vehicle route plan where some vehicles endure long travel times, an undesirable scenario aiming to minimize the average travel time of all vehicles, a concern in some traffic assignment frameworks. Second, this strategy significantly reduces the search space for the traffic assignment problem. Third, it prevents redundant computations, since finding a route between a pair of locations is a fundamental operation repeated numerous times during dynamic traffic assignment in this framework. *Algorithm* 1 outlines the process of generating a set of candidate routes for each pair of locations in a road network.

To compute a set of such candidate routes, first of all, we use Floyd-Warshall's algorithm [13] to find the shortest travel time between each pair of locations in the road network when there is no traffic in the road network. To do this, we use a weighted directed graph to represent the road network, where the weight of the edges is the shortest travel time of the corresponding road segment. The shortest travel time is calculated using the speed limit to divide the distance of the road segment. Then, we use a branch and bound breadth-first search algorithm to find a list of routes from each location to all other locations in the road network such that the travel time is bounded by $\lambda \times T_{i,j}$, where $T_{i,j}$ is the shortest travel time from location i to location j, $i \neq j$, and λ is a control parameter that is typically greater than 1 and less than 3.

Starting from a location, the branch and bound breadth-first search algorithm explores all routes that can be reached within a given time. Initially, we put the starting location into the queue. Then, we repeat the following step until the queue becomes empty. First, we remove the head element, which is a location, from the queue and check all locations that are associated with this location with a road segment one by one. If the travel time from the starting location i to this location j is less than or equal to $\lambda \times T_{i,j}$, then we add this location j to the queue.

Algorithm 1 Candidate routes generation

Input: A road network, including a set of locations in the road network; and λ, which is a multiplication factor

Output: A list of candidate routes for each pair of locations in the road network

1: Use Floyd-Warshall algorithm to find the shortest travel time between each pair of the locations
2: **for** each of the locations, i **do**
3: Use a branch and bound breadth-first search algorithm to find all the routes from this location to all the other locations in the road network such that the travel time from this location i to another location j is less than or equals to $\lambda \times T_{i,j}$, where $T_{i,j}$ is the shortest travel time from location i to location j, and $i \neq j$
4: **end for**

2) COORDINATED MULTI-VEHICLE DYNAMIC ROUTE GENERATOR

Another component in our framework is the coordinated multi-vehicle dynamic route generator. Given the current location of each of the vehicles in the road network and the current traffic in the road network, the coordinated multi-vehicle dynamic route generator is used to generate a new route plan for all the vehicles currently in the road network such that the average travel time of the vehicles is minimal based on the current traffic in the road network.

The traffic in a road network is dynamic by its nature, as there might be new vehicles joining the traffic and there might be some vehicles reaching their destination. As a result, a route plan that was optimal previously might not be optimal anymore. Thus, the coordinated multi-vehicle dynamic route generator is used periodically to calibrate the route plan of all the vehicles.

When generating a coordinated route plan for all the vehicles, first of all, we need to get the location of all the vehicles currently in the road network, and then use this information to get the number of vehicles currently traveling on each road segment. Once we get the current number of vehicles on each road segment, we can calculate the travel time for each road segment based on the current number of vehicles on each road segment.

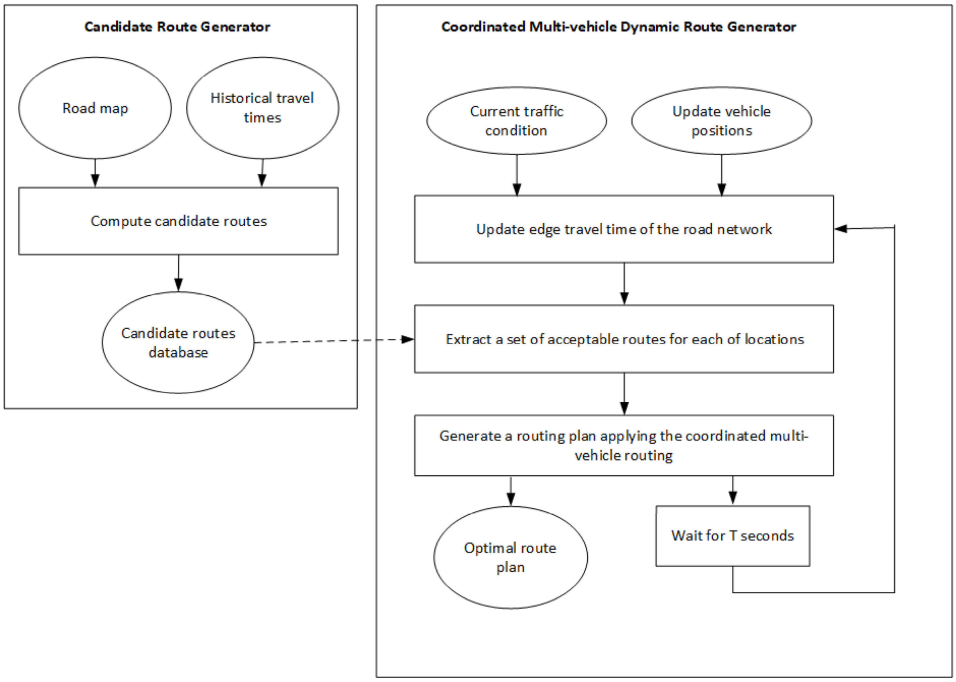

FIGURE 1. A new centralized coordinated multi-vehicle dynamic routing framework.

The travel time of a road segment depends on the current traffic flow of the road segment, and therefore, it is crucial to compute the actual travel time experienced on each road segment. Furthermore, as Sheffi states in [36], it is realistic to assume steady-state traffic flow behavior during rush hours. Therefore, to calculate the travel time experienced on a segment of roads at time t, the Bureau of Public Roads (BPR) formulation [37] is used.

$$T_l(t) = T_{free}\,[1 + \mu \left(\frac{n(t)}{C}\right)^{\beta}] \tag{1}$$

In the equation, T_{free} is the free flow travel time, n is the total number of vehicles on the road segment at time t, C is the maximum feasible flow of the respective road segment, μ and β are constant parameters, and $T_l(t)$ is the travel time of the road segment l at time t. The two constant parameters μ and β are usually set to 0.15 and 4 as advanced by the US Bureau of Public Roads [38]. Moreover, the values selected on two constant parameters implies that the practical capacity is the flow at which the travel time is 15% higher than the free-flow travel time. Further, the current number of vehicles is obtained from the traffic data collected by the service, and (2) is the equation used to calculate the maximum feasible flow of a road segment.

$$C = \frac{d}{(V_{len} + g_{min})} \tag{2}$$

In the equation, d is the distance from the road segment, V_{len} is the average vehicle length, and g_{min} is the minimum gap between two vehicles.

Based on updated travel times for all road segments, we can calculate the travel time of all precomputed candidate routes and exclude routes whose travel time is greater than $\alpha \times T_{i,j}$, where α is the ratio of the current travel time of all vehicles and the average travel time of all vehicles without traffic, and $T_{i,j}$ is the travel time from location i to location j without traffic. This algorithm is called "acceptable route list generation". Algorithm 2 is the description of this algorithm.

Algorithm 2 Acceptable Routes Selection

Input: A list of candidate routes between a pair of locations (i, j) and a control parameter α
Output: A list of acceptable routes between location i and location j

```
for each candidate route from i to j do
    calculate its travel time based on current traffic
    if the travel time is less than or equals to α × T*_{i,j} then
        add the route to the acceptable route list
    end if
end for
```

It is crucial to be proactive in excluding routes affected by traffic congestion. For example, during traffic congestion in one of the road segments along a route, the travel time for that route would increase significantly. Therefore, it is necessary to actively exclude such routes from the candidate set and generate a set of acceptable routes that are tailored to the current traffic conditions of the road network.

The utilization of an acceptable set of routes, reflective of real-time traffic conditions, is crucial before applying a coordinated multi-vehicle routing algorithm to find a route plan for each of the vehicles in the road network.

In this framework, the coordinated multi-vehicle routing dynamic problem is transformed into the following computational problem: Given a number of vehicles and a list of acceptable routes for each of the vehicles, select one route for each vehicle from its acceptable route list such that the average travel time of all the vehicles is minimal. The average travel time of all vehicles is calculated by summing up the total travel time of all the vehicles on the road network and then divide the sub by the number of vehicles on the road network. Algorithm 3 is a hill-climbing algorithm for solving this computational problem.

Algorithm 3 Heuristic Algorithm for Coordinated Multi-Vehicle Dynamic Routing

Input: A list of candidate routes for each pair of locations; the information about all the vehicles currently in the road network
Output: An optimized route plan for all vehicles currently in the road network

calculate the ratio of current average travel time between location and average travel time without traffic, α
for each pair of locations **do**
    Use Algorithm 2 to extract a list of acceptable routes
end for
improved = true
while improved **do**
    improved = false
    for each vehicle **do**
        Calculate the travel time of the vehicle from its current position to the end of the road segment i
        Randomly select an acceptable route from the list of acceptable routes from location i to its destination location j
        Recalculate the travel time of this vehicle and all other vehicles that share a road segment with this vehicle
        if the new total travel time of these vehicles is less than the old total travel time of these vehicles **then**
            Use the new route to replace the old route of this vehicle
            improved = true
        end if
    end for
end while

Before calculating the average travel time of all vehicles, the total travel time of each vehicle to reach its destination location at time t was calculated. However, the current position of a vehicle at time t is not necessarily a location of the road network, but an intermediate position of a road segment. Therefore, the starting location of a vehicle is considered as the location at the end of the current road segment. We used (3) to calculate the total travel time of a vehicle at time t.

$$V_T(t) = \sum_{\forall\, l \in V_{i,j}} T_l(t) + \delta_{k,i}(t) \tag{3}$$

where $T_l(t)$ is the travel time of the road segment l at time t, $V_{i,j}$ is the route of the vehicle from the location i to the location j, and $\delta_{k,i}(t)$ is the travel time of the vehicle from its current position k to location i at the end of the current road segment at time t. We used (4) to calculate the travel time from the current position of the vehicle to the end of the road segment.

$$\delta_{k,i}(t) = \frac{T_l(t)}{d_l} * d_{k,i} \tag{4}$$

where $T_l(t)$ is the travel time of the road segment in which the vehicle is on at time t, d_l is the distance of the road segment and $d_{k,i}$ is the distance from the current position to the location i at the end of the road segment.

Finally, we used (5) to calculate the average travel time of all vehicles at time t.

$$T_{ave}(t) = \frac{\sum_{i=1}^{n} V_{T_i}(t)}{n(t)} \tag{5}$$

where $V_T[i]$ is the total travel time of vehicle i to reach its destination at time t, n is the total number of vehicles in the road network at time t, and $T_{ave}(t)$ is the average travel time of all the vehicles at time t.

IV. DESIGN OF EXPERIMENTS

The performance of the proposed framework is evaluated by experiments. The main goal of the experiments is to answer the question of: How is the performance of the proposed framework compared to other methods? In this section, we elaborate how the experiments are designed. We first introduce the road traffic networks used for the experiments and the baseline methods used to compare the performance of the proposed framework.

A. METHODOLOGY

The proposed framework is evaluated by experiments, which are carried out on SUMO 1.16.0 [39]. SUMO is an open-source, portable, microscopic, and continuous multimodal traffic simulation package that is used for modeling, analyzing, and evaluating performance and managing urban road traffic networks. The proposed framework is evaluated by comparing its performance with that of three dynamic vehicle routing benchmark methods for two traffic scenarios in SUMO through simulation experiments. One traffic scenario is a popular benchmark traffic assignment problem (Section IV-C); another is a real-world traffic scenario (SectionIV-B).

For each of the two traffic scenarios, we start by getting each vehicle started traveling from its predefined departure position towards its predefined destination along the fastest route that can be taken at the time of departure. SUMO records the route in a file. Since the traffic in the road network is dynamically changing, at a regular time interval t, we use the proposed coordinated multi-vehicle dynamic routing algorithm to regenerate a new route for each of the vehicles in the traffic road network such that the average travel time of the vehicle is minimal, and we update the route for each of the vehicles and update the route stored in the file. We continue to do this until the experiment is completed. After the experiment finishes, we convert the recorded routes stored in the file into a simulation configuration of SUMO, and run SUMO to work out the average travel time. We repeat the process for the same traffic scenario with the three benchmark methods and calculate their average travel time.

We evaluated the proposed centralized coordinated multi-vehicle dynamic routing framework in average travel time of all vehicles in the road network. In the experiments, the proposed method is compared with three other baseline methods; iterative, incremental, and fastest route method and further described in Section IV-D in the same environment for the same traffic scenarios. The travel time of a vehicle n is calculated using the SUMO travel time function, which is $TT_n = \frac{L_n}{V_n}$, the ratio between the route length (L_n) and the average speed (V_n) of the vehicle n. Furthermore, we used $TT_n^* = \frac{\sum_{\forall n} TT_n}{N}$ to calculate the average system travel time of all vehicles currently traveling in the road network, where N denotes the total number of vehicles in the road network. TT_n denotes the travel time of the vehicle n. The average travel time of all vehicles is used as the measurement standard for evaluation methods. Our method is implemented in C # and all the experiments are carried out on a laptop with four cores (1.80 GHz Intel Core(TM) i7-8550U CPU) and 8 GB RAM.

B. ROAD NETWORKS

Two road networks with different road geometry are used in the simulation experiments. One is the road network depicted in Fig. 2, which is taken from [40]. This road network contains 13 intersections and 23 links. To simplify the experiment, all the roads in this road network are defined as bidirectional of two lanes. The cost of the link in the road network is the travel time in minutes between two ends of the link. Table 1 presents the link attributes of the network.

Another road network is a road network in the city of Bologna, Italy, which is frequently used as a test problem in the research on DTA. The road network includes the area between the two main streets Andrea Costa and Pasubio (Fig. 3). This road network was originally prepared by iTetris ("An Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions"), but later was made public in [41]. The Bologna traffic network includes 159 junctions and 268 road segments. The data set includes details of the network topology and information about each road segment, including the number of lanes, length, and free-flow travel speed. Furthermore, the same types of road have the same attribute values except for the length. The characteristics of the road network are summarized in Table 2.

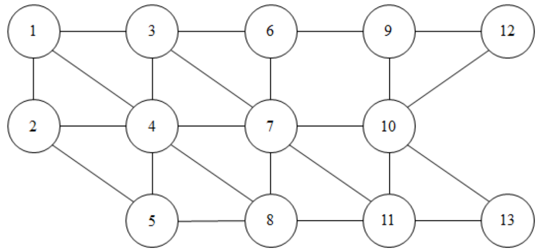

FIGURE 2. A road network adopted from [40].

TABLE 1. Link attributes.

Link	Max. speed (mins)	Free flow travel time (mins)
1-2	60	7
1-3	80	5
1-4	80	15
2-4	60	11
2-5	80	11
3-4	80	7
3-6	80	11
3-7	80	9
4-5	100	7
4-7	80	7
4-8	80	9
5-8	80	7
6-9	100	13
7-8	100	9
7-10	60	3
7-11	60	13
8-11	100	3
9-10	60	9
9-12	60	2
10-11	100	9
10-12	100	12
10-13	60	12
11-13	60	2

TABLE 2. Characteristics of the Bologna road network.

Parameter	Value
Total number of junctions	159
Total number of roads	268
Vehicle density (per Km2)	3751.7094
Total number of vehicles	11000
Minimal gap of vehicle	2.5m
Average length of vehicles	5m

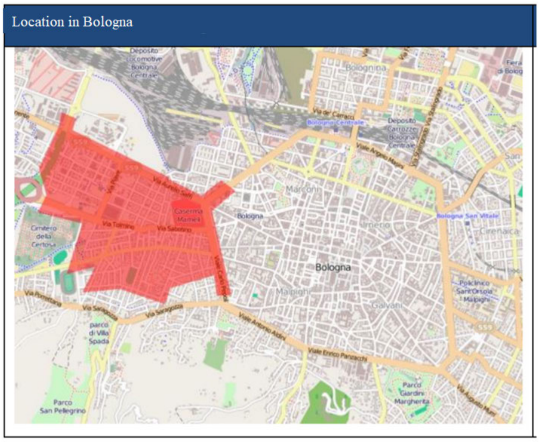

FIGURE 3. A road network from Bologna, Italy.

TABLE 3. Traffic Demand (per hour).

Origin node	Destination node	Vehicles (#)
1	12	600
1	13	300
2	12	400
2	13	400

C. TRAFFIC SCENARIOS

During our experiments, we have used two different traffic scenarios to accommodate the two traffic networks used. The following section summarizes the details of the traffic scenarios used.

1) TRAFFIC SCENARIO 1

This traffic scenario is based on the first road network depicted in Fig. 2, in which the authors defined four pairs of origin-destination and distributed the total traffic demand of 1700 vehicles as in Table 3. However, not all the demand of traffic is loaded onto the network at once. Total demand is loaded to the network incrementally by a fraction of the demand in four stages for each link. The fractions that we use in the experiments are 0.4, 0.3, 0.2, and 0.1 for the links 1-12, 1-13, 2-12 and 2-13, respectively. Traffic is loaded into the road network in a time interval of 300s.

2) TRAFFIC SCENARIO 2

The traffic scenario 2 used in this paper was adopted by [42]. The scenario was obtained by joining two areas around a football stadium and a hospital. In addition to the traffic demand for the morning traffic demand, the joined scenario also includes the traffic demand for a football game. During this period, 11000 vehicles enter the road network at a steady rate of all origin-destination pairs with given departure times. The demand profile comes from the study of [41]. Fig. 4 shows the traffic demand profile of the road network over time. In this traffic scenario, we perform traffic assignment at a time interval of 600s.

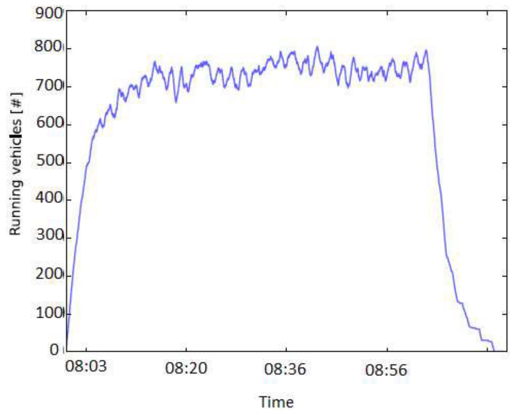

FIGURE 4. Traffic demand profile of the Bologna road network.

TABLE 4. Parameters used to create traffic scenario 1.

OD pair	Time interval (s)				Vehicles
	0-60	300-360	600-660	900-960	
1-12	240	180	120	60	600
1-13	160	120	80	40	400
2-12	120	90	60	30	300
2-13	160	120	80	40	400
Vehicles	680	510	340	170	1700

D. BENCHMARK METHODS

To demonstrate the effectiveness of the proposed framework, we compare it with three baseline methods; iterative assignment method, incremental assignment method, and fastest route method. The baseline methods are briefly summarized below:

- Iterative method: the algorithm starts with some initial set of routes and gradually proceeds to improve the travel times of vehicles. The goal of the iterative method is to find a route for each vehicle such that each vehicle cannot reduce its travel cost (usually the travel time) by using a different route. It does so iteratively by applying the last known edge costs to derive the travel costs of the next routing step [43].
- Incremental method: In incremental assignment, each vehicle will compute the fastest route at the time of departure and then recalculate the route based on the instantaneous travel time of each respective road link. It does not attempt to generate an optimal route solution, rather it prevents all vehicles blindly selecting the same route to travel.
- Fastest route method: This algorithm generates the routes with the fastest travel time without considering re-routing.

V. EXPERIMENT RESULTS

In this section, we compare the experimental results obtained from the proposed framework with those of the three benchmark methods described in Section IV-D. The three

TABLE 5. Statistics of the experimental results - Scenario 1.

Tool (Method)	Mean travel time (s)	Std. deviation	Min. travel time (s)	Max. travel time (s)
iterative method (duaIterate)	45.436	0.068	45.313	45.575
incremental method (one-shot)	46.221	0.051	46.130	46.332
fastest route (DUARouter)	47.023	0.0817	46.799	47.184
our method	44.983	0.116	44.748	45.207

TABLE 6. Statistics of the experimental results - Scenario 2.

Tool (Method)	Mean travel time (s)	Std. deviation	Min. travel time (s)	Max. travel time (s)
iterative method (duaIterate)	1128.575	38.155	1047.393	1193.234
incremental method (one-shot))	1167.311	37.615	1101.752	1290.101
fastest route (DUARouter	1414.834	41.235	1324.405	1497.769
our method	1102.524	42.7460	990.312	1163.711

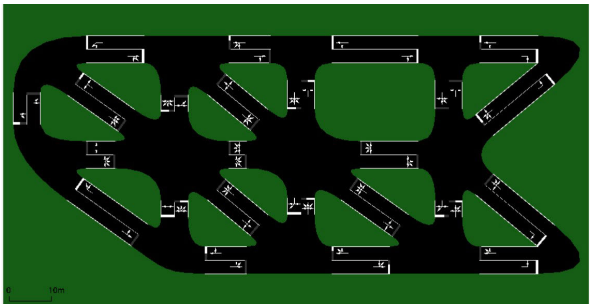

FIGURE 5. Road network of scenario 1 modeled by SUMO.

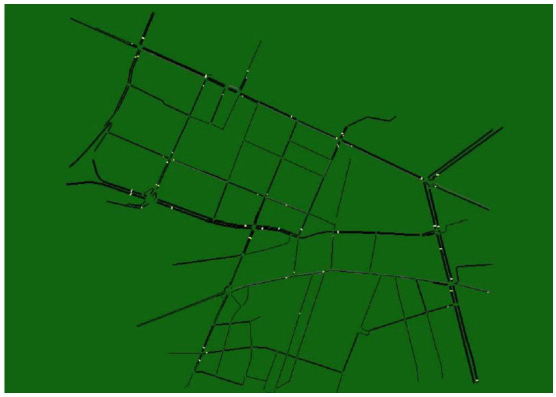

FIGURE 6. Road network of Bologna city modeled by SUMO.

benchmark methods were implemented using tools provided by the SUMO generator. The fastest route was calculated using the *duarouter* tool which is based on Dijkstra's shortest path algorithm using the free-flow edge traversal time as the edge weights. The iterative method was performed using the *duaiterate* tool which iteratively calls duarouter which finds routes for vehicles in a network with the last known edge costs (starting with empty-network travel times) and then calls *SUMO* to simulate "real" travel times result from the calculated routes. The result edge costs are used in the next routing step. This process is called iteratively to a fixed number of determined dynamically depending on the used options. Finally, *oneshot-assignemnt* tool in SUMO is used to implement the incremental assignment method with periodic re-routing.

For traffic scenario 1, we use the NetEdit tool in SUMO to create the traffic network depicted in Fig. 2 into a SUMO usable format (Fig. 5) and *OD2trip.py* script in SUMO to generate the demand file (trip file). We load the traffic demand at a steady rate by deploying all vehicles within the first minute of every time interval as shown in Table 4. The generated trip file includes the starting position, destination, and departure time of each vehicle. All four assignment methods use the same network topology and the same traffic flows for consistency.

Table 5 shows the comparison of the mean and standard deviation of our proposed framework with the other three assignment methods. The fastest route assignment is carried out with the default parameters of the *duarouter* tool in SUMO. In *duarouter*, the default parameters include the Dijkstra algorithm to calculate the shortest route with free-flow travel time. Furthermore, in the iterative DTA, we set the number of iterations to 20, and in the incremental assignment, the link time update interval is set to 300s. Similarly, both our proposed framework and iterative DTA update route assignment to optimize traffic at every 300 seconds time interval.

As can be seen from the statistics of the experimental results for Traffic Scenario 1 in Table 5 that among the four methods, our method has the shortest average travel time. In fact, the maximum travel time of the 30 runs of our method is still shorter than the minimum average travel time of the 30 runs of any of the other three methods.

In traffic scenario 2, we use the network file made available by [41] (Fig. 6) to perform the simulations. The trip file is created by extracting the OD pairs and departure times from the route file provided by the same study. For Traffic Scenario 2, the iterative DTA uses 30 iterations and the link time update interval for the increment assignment is set to 600 seconds. In addition, for Traffic Scenario 2, the proposed framework and iterative DTA update the route assignment every 600 seconds. Table 6 shows the statistics of the experiments for scenario 2.

It can be seen from the statistics in Table 6 that among the four methods, our method has the shortest average travel time. Compared to the iterative method, our method reduces the average travel time by 2.31%; compared to the incremental method, our method reduces the average travel time by 5.55%; and compared to the fastest method, our method reduces the average travel time by 22.07%.

To confirm this claim, we have also performed a two-sample paired t-test with the average travel time of our proposed method and the average travel time of each of the other three methods. The null hypothesis is that the average travel times of the two are equal. The P-values of these t-tests for our proposed method with the iterative method, the incremental method and the fastest method for Traffic Scenario 1 are 8.40776E-17, 1.34279E-29, and 3.84097E-35, respectively; and the P-values of these t tests for our proposed method with the iterative method, the incremental method, and the fastest method for Traffic Scenario 2 are 0.00739, 2.72914E-06, and 5.36163E-22, respectively. Since all P-values are less than 0.05, the null hypothesis is rejected. This indicates that the average travel time of our method is significantly shorter than that of any of the benchmark methods.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a coordinated multi-vehicle dynamic routing framework that considers the dynamics of the road network and fairness of all vehicles. This proposed framework uses a route generator to periodically and dynamically generate a new route plan for each of the vehicles on the road network, with the aim of minimizing the average travel time of all vehicles. Fairness is considered by assigning to all vehicles that are currently in the same or similar location and have the same destination location a route with the same or similar travel time.

This proposed coordinated multi-vehicle dynamic routing framework has been implemented and evaluated by comparing with three popular noncoordinated multi-vehicle routing frameworks by simulation. The simulation results have shown that the proposed framework significantly outperforms the three noncoordinated multi-vehicle routing frameworks.

The objective of this research is to prove that a coordinated multi-vehicle dynamic routing framework is better than noncoordinated multi-vehicle dynamic routing frameworks. The work presented in this paper is our preliminary research result. One of our future work is to parallelize the hill-climbing algorithm to reduce its computation time, as this multi-vehicle dynamic routing problem is a real-time optimization problem. Thus, it is necessary to make the implementation of the algorithm very efficient.

REFERENCES

[1] Aust. Traffic. (2023). Accessed: Sep. 5, 2023. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/australia-country-traffic

[2] T. S. J. Darwish and K. Abu Bakar, "Fog based intelligent transportation big data analytics in the Internet of Vehicles environment: Motivations, architecture, challenges, and critical issues," *IEEE Access*, vol. 6, pp. 15679–15701, 2018.

[3] A. Kirimtat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren, "Future trends and current state of smart city concepts: A survey," *IEEE Access*, vol. 8, pp. 86448–86467, 2020, doi: 10.1109/ACCESS.2020.2992441.

[4] M. Talebkhah, A. Sali, M. Marjani, M. Gordan, S. J. Hashim, and F. Z. Rokhani, "IoT and big data applications in smart cities: Recent advances, challenges, and critical issues," *IEEE Access*, vol. 9, pp. 55465–55484, 2021.

[5] H. Yu, H. Lee, and H. Jeon, "What is 5G? emerging 5G mobile services and network requirements," *Sustainability*, vol. 9, no. 10, p. 1848, 2017, doi: 10.3390/su9101848.

[6] M. Vallati, "Centralised versus decentralised traffic optimisation of urban road networks: A simulation study," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Eng. (ICITE)*, Sep. 2020, pp. 319–325.

[7] E. Angelelli, I. Arsik, V. Morandi, M. Savelsbergh, and M. G. Speranza, "Proactive route guidance to avoid congestion," *Transp. Res. Part B, Methodol.*, vol. 94, pp. 1–21, Dec. 2016, doi: 10.1016/j.trb.2016.08.015.

[8] E. Angelelli, V. Morandi, and M. G. Speranza, "Congestion avoiding heuristic path generation for the proactive route guidance," *Comput. Operations Res.*, vol. 99, pp. 234–248, Nov. 2018.

[9] E. Angelelli, V. Morandi, and M. G. Speranza, "Minimizing the total travel time with limited unfairness in traffic networks," *Comput. Operations Res.*, vol. 123, Nov. 2020, Art. no. 105016.

[10] E. Angelelli, V. Morandi, M. Savelsbergh, and M. G. Speranza, "System optimal routing of traffic flows with user constraints using linear programming," *Eur. J. Oper. Res.*, vol. 293, no. 3, pp. 863–879, Sep. 2021.

[11] M. C. Ho, J. M.-Y. Lim, C. Y. Chong, K. K. Chua, and A. K. L. Siah, "Collaborative vehicle rerouting system with dynamic vehicle selection," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14546–14555, Dec. 2023.

[12] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[13] R. Bellman, "On a routing problem," *Quart. Appl. Math.*, vol. 16, no. 1, pp. 87–90, 1958.

[14] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SCS-4, no. 2, pp. 100–107, Jul. 1968.

[15] K. Kim, M. Kwon, J. Park, and Y. Eun, "Dynamic vehicular route guidance using traffic prediction information," *Mobile Inf. Syst.*, vol. 2016, pp. 1–11, Jan. 2016.

[16] J. Pan, I. S. Popa, and C. Borcea, "DIVERT: A distributed vehicular traffic re-routing system for congestion avoidance," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 58–72, Jan. 2017.

[17] A. Zavin, A. Sharif, A. Ibnat, W. M. Abdullah, and M. N. Islam, "Towards developing an intelligent system to suggest optimal path based on historic and real-time traffic data," in *Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2017, pp. 1–6.

[18] J. Pan, I. S. Popa, K. Zeitouni, and C. Borcea, "Proactive vehicular traffic rerouting for lower travel time," *IEEE Trans. Veh. Technol.*, vol. 62, no. 8, pp. 3551–3568, Oct. 2013.

[19] Z. Liang and Y. Wakahara, "Real-time urban traffic amount prediction models for dynamic route guidance systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, pp. 1–3, Dec. 2014.

[20] M. Rezaei, H. Noori, M. Mohammadkhani Razlighi, and M. Nickray, "ReFOCUS+: Multi-layers real-time intelligent route guidance system with congestion detection and avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 50–63, Jan. 2021.

[21] Y.-T. Tseng and H.-W. Ferng, "An improved traffic rerouting strategy using real-time traffic information and decisive weights," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9741–9751, Oct. 2021.

[22] S. Aiko, P. Thaithatkul, and Y. Asakura, "Incorporating user preference into optimal vehicle routing problem of integrated sharing transport system," *Asian Transp. Stud.*, vol. 5, no. 1, pp. 98–116, 2018.

[23] M. Z. Arokhlo, A. Selamat, S. Z. M. Hashim, and M. H. Selamat, "Route guidance system based on self adaptive multiagent algorithm," in *Proc. Int. Conf. Comput. Collect. Intell.*, 2011, pp. 90–99.

[24] D. Cagara, B. Scheuermann, and A. L. C. Bazzan, "Traffic optimization on islands," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2015, pp. 175–182.

[25] M. Carey and D. Watling, "Dynamic traffic assignment approximating the kinematic wave model: System optimum, marginal costs, externalities and tolls," *Transp. Res. Part B, Methodol.*, vol. 46, no. 5, pp. 634–648, Jun. 2012, doi: 10.1016/j.trb.2012.01.008.

[26] Z. Qian, W. Shen, and H. M. Zhang, "System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost," *Transp. Res. Part B: Methodol.*, vol. 46, no. 7, pp. 874–893, Aug. 2012, doi: 10.1016/j.trb.2012.02.008.

[27] Y. Nie, "A cell-based merchant–nemhauser model for the system optimum dynamic traffic assignment problem," *Transp. Res. Part B, Methodol.*, vol. 45, no. 2, pp. 329–342, Feb. 2011, doi: 10.1016/j.trb.2010.07.001.

[28] N. Groot, B. De Schutter, and H. Hellendoorn, "Toward system-optimal routing in traffic networks: A reverse stackelberg game approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 29–40, Feb. 2015.

[29] B. Zhu, J. Li, Q. Yuan, J. Lu, and S. Yang, "An edge-assisted vehicle routing method based on game-theoretic multiagent learning," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2019, pp. 462–469.

[30] R. Claes, T. Holvoet, and D. Weyns, "A decentralized approach for anticipatory vehicle routing using delegate multiagent systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 364–373, Jun. 2011.

[31] J. Lu, J. Li, Q. Yuan, and B. Chen, "A multi-vehicle cooperative routing method based on evolutionary game theory," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 987–994, doi: 10.1109/ITSC.2019.8917441.

[32] H. Guo, Z. Cao, M. Seshadri, J. Zhang, D. Niyato, and U. Fastenrath, "Routing multiple vehicles cooperatively: Minimizing road network breakdown probability," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 2, pp. 112–124, Apr. 2017.

[33] Z. Cao, S. Jiang, J. Zhang, and H. Guo, "A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1958–1973, Jul. 2017.

[34] Y. Shigehiro, T. Miyakawa, and T. Masuda, "Road traffic control based on genetic algorithm for reducing traffic congestion," *Electron. Commun. Jpn.*, vol. 95, no. 4, pp. 11–19, Mar. 2012.

[35] R. Liu, H. Liu, D. Kwak, Y. Xiang, C. Borcea, B. Nath, and L. Iftode, "Themis: A participatory navigation system for balanced traffic routing," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2014, pp. 159–166, doi: 10.1109/VNC.2014.7013335.

[36] Y. Sheffi, *Urban Transportation Networks*, vol. 6. Upper Saddle River, NJ, USA: Prentice-Hall, 1985.

[37] *Traffic Assignment Manual for Application With a Large, High Speed Computer*, U.S. Dept. Commerce, Washington, DC, USA, 1964.

[38] C. Gawron, "Simulation-based traffic assignment. Computing user equilibria in large street networks," Ph.D. dissertation, Dept. Math. Comput. Sci., Cologne Univ., Cologne, Germany, 1998.

[39] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simul.*, Barcelona, Spain, 2011, pp. 1–6.

[40] J. de Dios Ortuzar and L. G. Willumsen, *Modelling Transport*, 3rd ed. Warwick, NY, USA: JWS, 2011.

[41] L. Bieker, D. Krajzewicz, A. Morra, C. Michelacci, and F. Cartolano, "Traffic simulation for all: A real world traffic scenario from the city of Bologna," in *Proc. Model. Mobility Open Data, 2nd SUMO Conf.*, Berlin, Germany, May 2015, pp. 47–60.

[42] D. Krajzewicz, R. Blokpoel, P. Cataldi, L. Bieker, J. Ringel, J. Leguay, and Y. Lopez, *ITETRIS Deliverable 3.2–Traffic Modelling: ITS Algorithms*, document 93 S., Apr. 2010.

[43] Y. C. Chiu, J. Bottom, M. Mahut, A. Paz, and R. Balakrishna, "Dynamic traffic assignment: A primer," *Transp. Res. E-Circular e-c153*, Tech. Rep., 2011.

M. D. R. L. SILVA received the B.Sc. (Special) degree in computer science from the University of Sri Jayewardenepura, Nugegoda, Sri Lanka, and the M.Phil. degree in computer science from the University of Moratuwa, Moratuwa, Sri Lanka. She is currently pursuing the Ph.D. degree with the School of Computer Science, Queensland University of Technology, Brisbane, Australia. Her current research interests include evolutionary computing, intelligent transportation systems, and big data analysis.

M. TANG received the B.E. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, the M.E. degree in computer science from Chongqing University, Chongqing, China, and the Ph.D. degree in computer systems engineering from Edith Cowan University, Perth, Australia.

He is currently a Faculty Member with the School of Computer Science, Faculty of Science, Queensland University of Technology, Brisbane, QLD, Australia. He has published more than 100 papers in prestigious journals and international conference proceedings, including IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Industrial Electronics, IEEE Transactions on Systems, Man and Cybernetics–Part B, IEEE Transactions on Cybernetics, IEEE Transactions on Services Computing, IEEE Transactions on Cloud Computing, and IEEE Access. His current research interests include evolutionary computation and nature-inspired computation and their applications in optimization, planning, placement, and scheduling.