Urban Traffic Route Guidance Method With High Adaptive Learning Ability Under Diverse Traffic Scenarios

Chuanhui Tang, Wenbin Hu, Simon Hu, and Marc E. J. Stettler

***Abstract*— With the rapid development of urbanization, the problem of urban traffic congestion has become increasingly prominent. Dynamic route guidance promises to improve the capacity of urban traffic management and mitigate traffic congestion in big cities. In the design of simulation-based experiments for most dynamic route guidance methods, the simulation data is generally estimated from a specific traffic scenario in the real-world. However, highly dynamic traffic in the city implies that traffic scenarios in real systems are diverse. Therefore, if a route guidance method cannot adjust its strategy according to the spatial and temporal characteristics of different traffic scenarios, then it cannot guarantee optimal results under all traffic scenarios. Thus, ideal dynamic route guidance methods should have a highly adaptive learning ability under diverse traffic scenarios so as to have extensive improvement capabilities for different traffic scenarios. In this study, an A* trajectory rejection method based on multi-agent reinforcement learning (A^*R^2) is proposed; the method integrates both system and user perspectives to mitigate traffic congestion and reduce travel time (TT) and travel distance (TD). First, owing to its adaptive learning ability, the A^*R^2 can comprehensively analyze the traffic conditions for different traffic scenarios and intelligently evaluate the road congestion index from a system perspective. Then, the A^*R^2 determines the routes for all vehicles from user perspective according to the road network congestion index. An extensive set of simulation experiments reveal that, under various traffic scenarios, the A^*R^2 can rely on its adaptive learning ability to achieve better traffic efficiency. Moreover, even in cases where many drivers are not fully compliant with the route guidance, the traffic efficiency can still be improved significantly by A^*R^2.**

***Index Terms*— Urban transportation network, traffic congestion, dynamic route guidance, reinforcement learning, A* algorithm.**

I. Introduction

MODERN urban transportation networks are highly complex systems with interdependency. With the fast development of modern society, urban traffic congestion has become a daily problem in many cities owing to the increase in traffic demands. Route guidance in a dynamic environment, which is a hot research topic in the field of intelligent transportation system (ITS), promises to improve the capacity of urban traffic management and mitigate traffic congestion in big cities. One consensus in many route guidance methods is to avoid traffic congestion as much as possible [1]–[4]. Traditionally, using traffic flow guidance has been suggested as a means of improving traffic efficiency to a certain degree; hence, many big cities have been vastly installing variable message signs (VMS) or using the radio to broadcast real-time traffic flow information. However, the information in these systems is open to all drivers, and congestion can easily be transferred to different parts of the network.

The route guidance of urban traffic is essentially the shortest path problem in graph theory. The A^* algorithm is the basic method for solving this problem, and a considerable amount of research has been done toward such an end [5]–[8]. In [6], an improved A^* algorithm is proposed to improve the reliability of route guidance in which penalties are added to roads with a high probability of congestion. In our variant of the A^* algorithm, however, the penalties added are based on the past travel trajectory of each vehicle. The A^* algorithm is widely used because it is easy to modify and is highly efficient in searching for optimal solutions. For instance, in [7] a bidirectional time-varying optimal path algorithm based on an improved heuristic function and a backward search technology was proposed; it significantly increased the path solving speed. In addition to these deterministic algorithms, there are also some heuristic algorithms [9]–[11] that can provide different best or suboptimal routes according to certain traffic scenarios. Many methods try to estimate the prevailing conditions or predicted conditions of a network [12]. These methods can be divided into two categories: short-term prediction [13], [14] and long-term prediction [15], [16]. Some methods use only current data [17], others take into account only historical data [18], while [19] integrate historical and current data. Traffic prediction can be very beneficial to drivers, but it is possible that a best match will not be found in the historical data database owing to changes in traffic patterns. Unlike most prediction methods, our method predicts the road congestion index rather than the road travel time (TT).

Route guidance methods have been studied from the system's perspective and/or the user's perspective. User-oriented route guidance methods that focus on a user's preferences have long been studied [20]–[22]. The shortest TT or travel distance (TD) is the primary consideration for such systems.

Manuscript received November 6, 2018; revised June 21, 2019 and October 17, 2019; accepted February 24, 2020. Date of publication April 6, 2020; date of current version May 3, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61572369 and Grant 61711530238. The Associate Editor for this article was H. Jula. *(Corresponding authors: Wenbin Hu; Simon Hu.)*

Chuanhui Tang and Wenbin Hu are with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: tangchuanhui@whu.edu.cn; hwb@whu.edu.cn).

Simon Hu is with the College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China, and also with the Civil and Environmental Engineering Department, Imperial College London, London SW7 2BU, U.K. (e-mail: simonhu@zju.edu.cn).

Marc E. J. Stettler is with the Civil and Environmental Engineering Department, Imperial College London, London SW7 2BU, U.K. (e-mail: m.stettler@imperial.ac.uk).

Digital Object Identifier 10.1109/TITS.2020.2978227

This may lead to traffic congestion because the optimal path for the user is the path that only maximizes the user's preferences. Route guidance methods for finding the system-optimal traffic distribution have also been studied [23]–[25]. It is well-known that a system-oriented guidance system sometimes seems unfair because certain users' preferences are sacrificed for the sake of the system's interests. Thus, in a system-oriented guidance system, when the unfairness reaches a certain level, the user may not follow the route guidance, thereby violating the original intention of the system optimization. By restraining users' selfish behavior or avoiding congested roads, a compromise can be achieved between individual and global benefits [1], [3], [26].

A multi-agent system (MAS) has been widely used in route guidance [27]–[29], because it has the characteristics of self-adaptation and cooperation, and can exchange information effectively and in real-time based on VANET [30]–[32]. For instance, Cao *et al.* [2] proposed a pheromone-based MAS, in which infrastructure agents collect pheromones left by vehicle agents on the road to predict the road congestion level in the near future. Currently, with the continuous development of machine learning, optimization algorithms based on reinforcement learning (RL) have been a research frontier of ITS in recent years [33]–[35]. Given the inherent nature of high dynamics and uncertainty of the urban traffic system, ITS is an excellent application scenario for multi-agent RL (MARL) algorithm [36]. Compared with its application in traffic signal optimization [37], [38], the application of RL in route guidance is still in its infancy. In [35], the author introduces the concept of node pressure to the traffic control system, and proposes a multi-agent static-route calculation method via RL. In [39], Sadek shows the feasibility of using RL to provide on-line dynamic route guidance for drivers by deploying a single self-learning agent on a simple network. In order to avoid congestion and reduce traffic delays, Arokhlo *et al.* [40] proposed a MARL method to calculate the minimum cost path from the origin to the destination. To the best of our knowledge, there are few studies on the application of MARL in route guidance [41], because the curse of dimensionality is the main challenge. In this study, we apply a new MARL model to dynamic route guidance, which not only effectively reduces the complexity of the algorithm, but also realizes cooperation between agents.

Traffic simulation is a powerful tool for obtaining actual effects that cannot be obtained from analytical methods. To determine the effectiveness of a route guidance method, a simulation-based experiment is usually conducted. However, the simulation data are generally estimated from a specific traffic scenario in the real-world, as it is difficult to estimate all traffic scenarios in the real-world. Therefore, if a route guidance method cannot adjust its strategy according to the spatio-temporal characteristics of different traffic scenarios, then it cannot guarantee optimal results under all traffic scenarios. Although most dynamic route guidance methods have a certain degree of dependence on simulation data owing to their lack of adaptive capabilities, there are still some methods that can be used to overcome this problem, such as the dynamic traffic assignment (DTA) [42]–[44] algorithm; however, the DTA is not a viable solution for real-time traffic guidance because of its high time complexity and the need to know the origin-destination (OD) set before iteration.

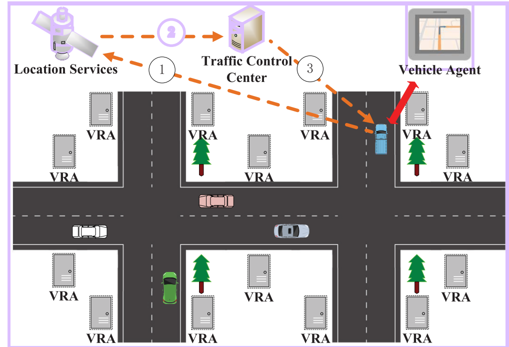


Fig. 1. Overview of the proposed method.

In this study, we propose an A* trajectory rejection method based on multi-agent RL (A*R^2) to effectively mitigate traffic congestion and to reduce the TT and TD under diverse traffic scenarios. To achieve this goal, the A*R^2 is designed to integrate both system and user perspectives, i.e., to mitigate the traffic congestion of the system and reduce the TT and TD for the users. The A*R^2 method consists of two parts: the modular Q-learning (MQ) algorithm and the A* trajectory rejection (A*R) algorithm. First, based on the MQ algorithm, which is an improved MARL, the A*R^2 can comprehensively analyze the traffic conditions and intelligently evaluate the road congestion index from a network level. Then, based on the A*R algorithm, the A*R^2 determines the routes for all the vehicles in the road network from a user perspective according to the congestion index of road network. Because RL has lower requirements for system prior knowledge, it is good at learning the new traffic scenario and has strong adaptive ability for different traffic scenarios. Thus, irrespective of the traffic scenario, the A*R^2 can learn the best strategy to mitigate traffic congestion from a system perspective and increase traffic efficiency from user perspective adaptively.

The major contributions of this study are as follows: (1) An MARL algorithm used for avoiding system-level traffic congestion takes into account the traffic conditions of upstream and downstream roads that have a direct impact on road condition; (2) a variant of the A* algorithm used for reducing the TT and TD from the user perspective avoids choosing the road that has been used by imposing penalties on it; and (3) a dynamic route guidance method with strong adaptive learning ability that can adjust its strategy according to the spatio-temporal characteristics of different traffic scenarios.

The remainder of the paper is organized as follows: Section II presents the system architecture of A*R^2 method. Section III describes our proposed method. Section IV presents the simulation of the A*R^2 method inner city transportation network and provides experimental results. Section V concludes the paper.

II. System Architecture

This section presents the system architecture of the A*R^2 method. Fig. 1 illustrates the three key components of the system: location services (LS), a traffic control center (TCC), and vehicle agents (VAs).

The LS are used to obtain the real-time location of vehicles, which is used to compute the traffic condition on the network.

The TCC aims to improve traffic efficiency from the system perspective and is dedicated to mitigating traffic congestion on the network. In this system architecture, the TCC generates a virtual road agent (VRA) for every single road in the network. Each VRA first obtains information about the road it controls from the LS; it then applies the MQ algorithm to calculate the travel cost of each road in the network. There are various criteria to define the travel cost of roads in road networks, such as the TT of roads. In our study, we define the travel cost of a road as the congestion index, which is determined by the action taken by the corresponding VRA (detailed in Section III), and its value is equal to the free flow TT multiplied by a real number greater than one. The congestion index is set by the VRA to achieve traffic equilibrium from the network level based on real-time traffic conditions.

The VAs elaborate on how to improve traffic efficiency from the user perspective with the objective to minimize the TT and TD. Each VA deployed on a vehicle applies the A*R algorithm, providing the vehicle with capabilities to find the user-optimal route according to the congestion index of the road network. Obviously, each VA can work independently, so the working mode of VAs is decentralized.

The work of the TCC and VAs can be seen as being independent of each other, so the proposed system is a hierarchical architecture in which the TCC centrally calculates the congestion index of the road network from the system perspective, while all VAs determine user-optimal routes in a decentralized mode.

The overview of the proposed method is as shown in Fig. 1. First, the LS are used to determine the number of vehicles on each road in the road network. Then, the data, which represents the current traffic condition, is transmitted to the TCC. The TCC will set a congestion index for all roads to manage traffic. Finally, the congestion index of each road will be delivered to the VAs of all vehicles to provide decision-making support for each vehicle.

III. Proposed Method

This section describes the methods we proposed in this study. The A*R^2 method consists of two parts, the MQ algorithm and the A*R algorithm. In the next few subsections, we will introduce the MQ and A*R algorithm in detail.

A. MQ Algorithm

The detailed process of computing the network congestion index is shown in Algorithm 1. The process is repeated until all vehicles have reached their destinations. It should be noted that the time interval in this paper is represented by σ, and its optimal setting is discussed in the experiment part in Section IV.

The goal of the MQ algorithm is to obtain an accurate estimation of road travel cost according to the spatial and temporal characteristics of different traffic scenarios. With the agent acting as a process controller, the environment of the transportation system can be modeled as a Markov decision process (MDP) to describe the interaction between the transportation system environment and the agent. The optimal strategy for an MDP can be solved by RL. Single-agent RL, such as the Q-learning algorithm, is the focus of most studies. The optimal strategy for the agent in an arbitrary state s_t is to select action a_t for the largest $Q^t(s_t, a_t)$; then, by any Q^0 value function, if the agent selects the action at each iteration according to the following formula, then after an infinite number of iterations, the sequence Q^t will converge to the optimal value.

Algorithm 1 Compute the Network Congestion Index

Input : The number of vehicles on each road;
Output: The congestion index of the network.

```
for time t =1, 2, 3 ... do
    Get the number of vehicles on each road in the road network
    for each VRA do
        Use the MQ algorithm to calculate the congestion index of the corresponding road
    end for
    Output the congestion index of the network
end for
```

$$Q^t(s_t, a_t) = (1-\alpha)Q^{t-1}(s_t, a_t) + \alpha\left(r^t + \gamma \max_{a_{t+1}\in A} Q^{t-1}(s_{t+1}, a_{t+1})\right) \quad (1)$$

where $\alpha \in (0,1)$ is the learning rate and $\gamma \in (0,1)$ is the discount factor.

However, in the Q-learning algorithm, the single agent greedily maximizes its own utility without taking into account the impact of decisions made by other agents. To overcome this problem, the system can be modeled as an MARL model. The simplest MARL is a combination of all the single agents, in which each VRA can cooperate with all the other VRAs to optimize the traffic network state by analyzing the possible impacts of other VRAs and making better decisions. However, the problem of dimensionality is the major challenge associated with the application of MARL to a route guidance problem. In road network G, each VRA needs to maintain a table whose size is $|S_1| \times \ldots \times |S_M| \times |A_1| \times \ldots \times |A_M|$, where M is the number of road sections, S_i represents the state spaces for road i, and A_i represents the action spaces. Thus, with an increase in agents, MARL encounters the dimensionality problem that is due to the exponential increases in both joint state and joint action spaces. The problem of dimensionality not only results in a large use of memory space, but also leads to slow convergence of Q functions.

Therefore, we designed a new coordination mechanism to avoid these problems. First, we designed an RL mechanism between two agents, which we termed 2-agent RL. Then, based on 2-agent RL, we proposed MARL, i.e., the MQ algorithm, in which the VRA of each road can form multiple incidences of 2-agent RL with neighboring VRAs, and these 2-agents together form a module.

In the following subsection, we first present the definitions and notations of the MQ algorithm. Then, we will introduce the mechanism of a 2-agent RL, because it is the basis of the MQ algorithm. Finally, we will present an implementation of the MQ algorithm.

1) Definitions and Notations: Given an urban road network $G = (E,V)$, which consists of a set of road sections (directed edge, i.e., two different directions of a real road will be regarded as two different road sections) $E = \{E_1,E_2,\ldots,E_M\}$ (M is the number of road sections) and a set of intersections $V = \{V_1,V_2,\ldots,V_N\}$ (N is the number of intersections). The agent VRA on road i is denoted as VRA_i where $i \in E$. Before describing our model in detail, we first introduce the definitions of state, action, and reward in our algorithm, because they are fundamental concepts in designing RL systems.

State: Number of Vehicles. On each road, the number of vehicles is a key factor determining whether the road is congested. Therefore, in order to considerably simplify the design and reduce the space complexity of MARL, the VRA's state is designed to be represented by the total number of vehicles on the corresponding road.

Action: Congestion index of the road. The action of VRA is designed to reduce congestion from the system's perspective and will effectively manage the traffic flow of each road in the next time period. Therefore, the VRA's action is designed to represent the variable congestion index of the road. The congestion index is set by the VRA to achieve traffic equilibrium from the network level; it can be used as the travel cost of the road. The road congestion index is negatively correlated with the expectation for road traffic in the next time period, that is, the larger the value, the less desirable it is for more vehicles to use the road. In this paper, any action in the action space is the free flow TT multiplied by a real number greater than one and less than 5.

Reward: Reduction in the distance of the corresponding road between the number of vehicles and optimal flow. Where 'distance' is defined as |Number of Vehicles − Optimal Flow|. In this paper, the difference between the distance of two successive decision points is defined as the reward for the corresponding VRA. After executing the selected action, the reward can be positive or negative. The reward is positive only when the distance between the number of vehicles and the optimal flow is reduced. Otherwise, it is negative, which implies that the action results in a low road utilization or congestion.

OF_i (i.e., optimal flow of road i): The optimal traffic flow of road i. On a road, the low level of traffic leads to underutilization of the road (waste of road resources). However, a high volume of traffic will result in congestion, which will reduce traffic efficiency. We define optimal flow (OF) as the optimal traffic flow of the road, where 'optimal' implies that the vehicle number will neither result in congestion nor underutilization of the road. The optimal traffic flow depends on certain physical attributes of the road. These attributes are listed in TABLE I.

According to the attributes described in TABLE I, OF_i is calculated as follows:

$$OF_i = lan_i \times len_i \times sth_i \times rto_i \times grd_i \times sfy_i \times F_{OF}\% \tag{2}$$

TABLE I

PHYSICAL ATTRIBUTES OF THE ROAD

Parameter	Description
lan	Lanes of the road section, $0<lan$.
len	Length of the road section, $0<len$.
sth	Smoothness of the road section, the larger the value, the more flat the road is, $0<sth\leq 1$.
rto	The ratio of the speed limit of the road section to the max speed limit of the road network, $0<rto\leq 1$.
grd	Gradient of the road section, the steeper the road, the smaller the value, $0<grd\leq 1$.
sfy	Road safety, the larger the value, the smaller the possibility of accident happening, $0<sfy\leq 1$.

TABLE II

NOTATION OF MQ ALGORITHM

Notation	Definition
G	Urban road network.
E	Set of road sections, $E=\{E_1,E_2,\ldots,E_M\}$.
V	Set of intersections, $V=\{V_1,V_2,\ldots,V_N\}$.
σ	The time interval until the next calculation of the congestion index.
i,j	Road section $i,j \in E$.
VRA_i	VRA of road section i.
S_i	State space of VRA_i, $s_i, s_i' \in S_i$.
S	Joint state space of VRA_i and VRA_j, $S = S_i \times S_j$, $s, s' \in S$.
A_i	Action space of VRA_i, $a_i, a_i' \in A_i$.
A	Joint action space of VRA_i and VRA_j, $A = A_i \times A_j$, $a, a' \in A$.
OF_i	The optimal traffic flow of road i.
F_{OF}	Optimal flow coefficient of road network.
$d_j^{a_j}(s)$	An event of VRA_j, $d_j^{a_j}(s)$=(s, a_j).
$D_j(s)$	An event set of VRA_j, $D_j(s)$=$\left\{\cup_{a_j' \in A_j}\left(s, a_j'\right)\right\}$.
NR_i	Neighboring roads of road i, furthermore, the roads that are close to road i and have a direct impact on road i.
NR_i^k	The kth neighboring road of road i.
$\Omega_{j,a_j}^{t,s}$	The number of occurrences of event $d_j^{a_j}(s)$ in Γ_j up to time t.
$\Omega_j^{t,s}$	The number of occurrences of events of $D_j(s)$ in Γ_j up to time t.
Γ_j	A set of sequences that contain a joint state and an action, Γ_j=$\left\{\left(s^0, a_j^0\right), \left(s^1, a_j^1\right), \cdots\right\}$.

where F_{OF} is the optimal flow coefficient of the transportation network, which adjusts the calculation of optimal flow. If the number of vehicles in the entire transportation network is very small, then the F_{OF} can be set to a smaller value. If the number of vehicles is very large, then F_{OF} should be set to a larger value. The setting of F_{OF} has a large influence on the effect of the algorithm, which is studied in Section IV.

TABLE II presents the notation system to be used in the MQ algorithm.

2) 2-Agent RL: There are many modules in our model, and each module is centered on a VRA. Each module contains multiple incidences of 2-agent RL, which are formed between the center VRA and each neighboring VRA. We will now introduce the definitions of 2-agent RL.

Definition 1: 2-agent RL in this paper can be represented by the tuple $\{S, A, P, R\}$, where

- $S = S_i \times S_j$ is the discrete joint state space of VRA_i and VRA_j, where S_i and S_j are referred the local state spaces of VRA_i and VRA_j, respectively. Local here implies that the components belong to just one VRA.

- $A = A_i \times A_j$ is the discrete joint action space of VRA_i and VRA_j, where A_i and A_j are referred to as the action spaces of VRA_i and VRA_j, respectively.
- $P = P\left(A_i|S, A_j\right)$ is the transition probability function. $p\left(a_i|s, a_j\right)$ is the probability that VRA_i takes the action a_i when VRA_j takes the action a_j at the joint state s, where $a_i \in A_i$, $a_j \in A_j$, and $s \in S$. Obviously, the transition probability map is subject to the constraint $\sum_{a_i' \in A_i} p\left(a_i'|s, a_j\right) = 1$.
- $R = R(S, A, S)$ is the reward function. $r\left(s, a, s'\right)$ is the reward when VRA_i and VRA_j take the joint action $a \in A$ at the joint stage $s \in S$.

Definition 2: A 2-agent RL is said to be reward independent if there exist r_i and r_j such that

$$r_{ij}\left(s, a, s'\right) = r_i\left(s_i, s_i'\right) + r_j\left(s_j, s_j'\right) \tag{3}$$

where $s = \left(s_i, s_j\right)$, $a = \left(a_i, a_j\right)$, and $s' = \left(s_i', s_j'\right)$, respectively. In other words, the overall reward of VRA_i and VRA_j is composed of two independent reward functions, each of which depends only on the local state and next local state of the VRAs.

Definition 3: A 2-agent RL is said to be transition independent if the transition probability function p is subject to

$$p\left(a_j|s, a_i\right) = p\left(a_j|s\right) \tag{4}$$

$$p\left(a_i|s, a_j\right) = p\left(a_i|s\right) \tag{5}$$

where $s \in S$. In other words, the new action of each VRA depends only on its previous joint stage.

Obviously, if a 2-agent RL is both transition independent and reward independent, then it can be solved as two separate MDPs.

Definition 4: An event $d_j^{a_j}(s) = (s, a_j)$ is a triplet that includes a joint state and an action. An event set $D_j(s) = \left\{\cup_{a_j' \in A_j}\left(s, a_j'\right)\right\}$ is a set of events at joint state s.

Definition 5: A record $\Gamma_j = \left\{\left(s^0, a_j^0\right), \left(s^1, a_j^1\right), \ldots\right\}$ is a sequence that contains a joint state and an action, which implies that VRA_j takes the action at the joint state. Let symbol $\Gamma_j(t)$ represent the tth record $\left(s^t, a_j^t\right)$, and it subject that s^{t+1} is outcome state of s^t.

Therefore, the transition probability at time t can be derived by

$$p^t\left(a_j|s, a_i\right) = p^t\left(a_j|s\right) = \frac{\Omega_{j,a_j}^{t,s}}{\Omega_j^{t,s}} \tag{6}$$

where $\Omega_{j,a_j}^{t,s}$ represents the number of occurrences of event $d_j^{a_j}(s)$ in record Γ_j up to time t and $\Omega_j^{t,s}$ represents the number of occurrences of events of $D_j(s)$ in record Γ_j up to time t.

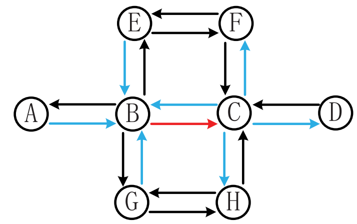


Fig. 2. Neighbor roads of $B \to C$.

Therefore, $\Omega_{j,a_j}^{t,s}$ and $\Omega_j^{t,s}$ can be derived by

$$\Omega_{j,a_j}^{t,s} = \sum_{k=0}^{t-1} \boldsymbol{I}_{\Gamma_j(k)}\left(d_j^{a_j}(s)\right) \tag{7}$$

$$\Omega_j^{t,s} = \sum_{k=0}^{t-1} \boldsymbol{I}_{D_j(s)}\left(\Gamma_j(k)\right) \tag{8}$$

where $\boldsymbol{I}$ is an indicator function.

3) Implementation of the MQ Algorithm: To reduce congestion from the system's perspective, the MQ algorithm is designed to apply RL to calculate the congestion index of each road of the network based on collected real-time traffic condition information. The principle of the MQ algorithm is that, although the vehicle may not choose the user-optimal route, it can meet the transportation network traffic expectations, thus mitigating congestion and reducing the overall TT delay.

Instead of considering equilibrium as necessary, the goal of the MQ algorithm is to achieve an optimal payoff in the presence of other VRAs. To overcome the complexity barrier, for any VRA_i, only some VRAs that have the most significant impact on the VRA_i should be considered. We think the neighboring roads have the most significant impact on road i, and its definition is as follows.

NR_i (i.e., neighboring roads of road i): Roads that are close to road i and have a direct impact on road i. Taking the road $B \to C$ in Fig. 2 as an example, $\mathrm{NR}_{B\to C}$ is $\{E \to B$, $A \to B$, $G \to B$, $C \to H$, $C \to D$, $C \to F$, $C \to B\}$. The traffic condition of road $B \to C$ is directly affected by any of the neighboring roads. Although $B \to A$, $B \to E$, $B \to G$, $F \to C$, $H \to C$, and $D \to C$ are also close to $B \to C$, their traffic condition cannot directly affect the traffic condition of $B \to C$, so they are non-neighboring roads. In order to better describe, we use NR_i^k to represent the kth neighboring road of road i.

For any VRA_i, $\mathrm{VRA}_{\mathrm{NR}_i}$ can form a module with VRA_i. This module consists of multiple independent 2-agent RL. Taking the road $B \to C$ in Fig. 2 as an example, the VRA of road $B \to C$ can form a module with the neighboring roads' VRA. In this module, the VRA of road $B \to C$ and each VRA in $\mathrm{NR}_{B\to C}$ forms independent 2-agent RL in turn.

To simplify the presentation, for any road i, the kth neighboring road (NR_i^k) is represented by symbol j. At time t, not only the states s_i^t and s_j^t of VRA_i and VRA_j can be observed, but also the information prior to time t can be recorded, such as the best action selection a_i^{t-1}, a_j^{t-1} and the reward value r_i^{t-1} of VRA$_i$ and VRA$_j$ at time $t-1$.

Therefore, the best response br_i^t at joint state $\left[s_t^t, s_j^t\right]$ can be derived by

$$br_i^t = \max_{a_i \in A_i} \sum_{a_j \in A_j} Q_{i,j}^{t-1}\left(\left[s_i^t, s_j^t\right], \left[a_i, a_j\right]\right) \times p_{i,j}^{t-1}\left(a_j \middle| \left[s_i^t, s_j^t\right]\right) \tag{9}$$

$Q_{i,j}^t$ can be updated by

$$\begin{aligned} Q_{i,j}^t&\left(\left[s_i^{t-1}, s_j^{t-1}\right], \left[a_i^{t-1}, a_j^{t-1}\right]\right) \\ &= \alpha \times \left(r_i^{t-1} + \gamma\, br_i^t\right) + (1-\alpha) \\ &\quad \times Q_{i,j}^{t-1}\left(\left[s_i^{t-1}, s_j^{t-1}\right], \left[a_i^{t-1}, a_j^{t-1}\right]\right) \end{aligned} \tag{10}$$

where $\alpha \in (0,1)$ is the learning rate and $\gamma \in (0,1)$ is the discount factor.

The challenge is coordinating these multiple independent 2-agent RL to obtain the best action a_i^t. The following formula will demonstrate the procedure.

$$\begin{aligned} a_i^t = \arg\max_{a_i \in A_i} \sum_{j \in \mathrm{NR}_i} \sum_{a_j \in A_j} Q_{i,j}^t&\left(\left[s_i^t, s_j^t\right], \left[a_i, a_j\right]\right) \\ &\times p_{i,j}^t\left(a_j \middle| \left[s_i^t, s_j^t\right]\right) \end{aligned} \tag{11}$$

The best action a_i^t is the congestion index of road i at time t. The congestion index of each road will then be delivered to the VAs of all the vehicles for determining the optimal routes.

B. A*R *Algorithm*

The detailed process of finding the user-optimal route for each vehicle is shown in Algorithm 2, where F_t is referred to as all the vehicles in the network at any time t. The process is repeated until all the vehicles have reached their destinations.

Algorithm 2 Find the User-Optimal Route for Each Vehicle

Input : F_t, the set of vehicles at any time t in the network; The congestion index of the network;

Output: The user-optimal routes of all the vehicles.

```
for VA of f ∈ F_t do
    if f will leave the current road then
        Apply the A*R algorithm to compute the user-optimal
        route of vehicle f
    end if
end for
Output the user-optimal routes of all the vehicles
```

In the field of transportation, there are many studies on the efficiency and application of the shortest path search problem. Compared to other shortest path algorithms (such as the Dijkstra [45] and Floyd [46] algorithm), the advantage of the A* is that it combines a breadth-first search algorithm with a greedy strategy, which significantly improves the efficiency of the search. Based on the A* algorithm, the heuristic function is further improved, and the A*R algorithm is proposed. In this subsection, we first introduce the network representation used in the A*R. Then, we introduce the implementation of the A*R. Finally, we will analyze the advantage of the A*R algorithm.

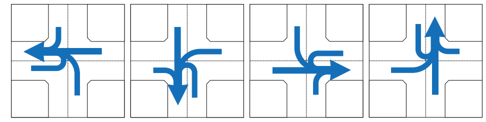


Fig. 3. Sixteen different travel trajectories.

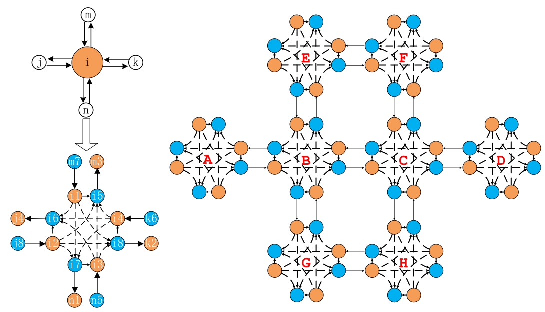


Fig. 4. Network representation.

1) Network Representation: The urban transportation network is complex and interdependent, with numerous intersections. For a vehicle traversing the network, the delay on the intersection is a key factor affecting the TT of the vehicle. In the most widely used network representation, each intersection is represented by only one node, the direction of travel between intersections is represented by edges, and the weight of the edges represents a certain cost, as shown in Fig. 2. However, it is difficult to represent the 16 different vehicle travel trajectories, as shown in Fig. 3, within the intersection. Therefore, we adopted a better network representation to avoid these problems.

To represent the travel cost within the intersection, we can expand the intersection node one-to-many according to the actual construction of the intersection; then, we can add a virtual edge to represent the sixteen movements in the intersection, whereby the weight of the virtual edge represents the travel cost of the corresponding movement. As shown on the left of Fig. 4, node (intersection) i of the original network representation is expanded to 8 nodes $(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8)$ in the new network representation, and 16 virtual edges are added to represent the corresponding movements within the intersection i. Therefore, after this conversion, the new network representation of the original transportation network in Fig. 2 is shown on the right part of Fig. 4.

We use notation Θ to represent all the nodes in the new network representation. To indicate the relationship between the node in the new network representation and the node in the original network representation, for any node $x \in \Theta$, $O(x)$ represents the original network's node whose extension contains the x node. For example, on the right part of Fig. 4, $O(A_1) = O(A_2) = \cdots = O(A_8) = A$, and $O(B_1) = O(B_2) = \cdots = O(B_8) = B$.

In the following subsection, a 'road' refers to either a virtual edge or a real edge.

2) Implementation of the A*R *Algorithm:* In the A* algorithm, a heuristic function is used to reduce a large number of invalid searches. Given a current location S and destination D, for any node x, the estimation function of node x is obtained by

$$f(x) = g(x) + h(x) \tag{12}$$

where $g(x)$ is the distance function that represents the smallest TT from the current location S to x, and $h(x)$ is a heuristic estimate of the TT between x and destination D. In each step, the node selected for expanding is the node with the smallest $f(x)$ value, which implies that the node shows the most promise to reach the destination faster.

The A* algorithm is the basic method for solving the shortest path problem, but we cannot directly use it in high dynamic traffic, because the TCC's system-oriented traffic flow induction strategy may have a conflict with the interests of individual vehicles. For example, if the TCC guides the vehicle to change its route, which contains a large number of road sections that the vehicle has traveled along, then the driver needs to consider whether it is worth taking this route. The A* algorithm does not take into account the effects of the road sections that a vehicle has already traveled along. Therefore, the A* algorithm should be further improved.

We think that re-selecting the roads that have already been traveled along is a waste of road resources. The principle of the A*R algorithm is to add penalties to this behavior. In the A*R algorithm, the notation $\mathcal{P}$ represents all the roads that the vehicle has traveled along, and the notation $\mathcal{L}(x)$ represents all the roads of the shortest route from the current location to the x node. We use notation $\mathcal{R}(x)$ to denote the set of previously traveled roads contained in $\mathcal{L}(x)$; then, $\mathcal{R}(x)$ is calculated as follows:

$$\mathcal{R}(x) = \mathcal{P} \cap \mathcal{L}(x) \tag{13}$$

We use notation ψ to represent the total time the vehicle spent on the road, and ψ_x^y to represent the total time the vehicle spent on road $y \rightarrow x$. Then, we define Ψ_x^y as follows:

$$\Psi_x^y = \sum_{\substack{\exists s \in \Theta, \\ O(s)=O(y)}} \sum_{\substack{\exists t \in \Theta, \\ O(t)=O(x)}} \left(\psi_s^t + \psi_t^s\right) \tag{14}$$

Then, the function $e(x)$ is defined as the total penalty for calculating the optimal path from the starting node to the x node, which can be obtained by

$$e(x) = F_{\mathrm{A*R}} \times \sum_{g \rightarrow h \in \mathcal{R}(x)} \Psi_h^g \tag{15}$$

where $F_{\mathrm{A*R}}$ is the penalty coefficient of the A*R algorithm. The value of $F_{\mathrm{A*R}}$ has a large influence on the effect of the algorithm. If $F_{\mathrm{A*R}}$ is too small, the A*R algorithm will degenerate into the A* algorithm, and the setting of the $F_{\mathrm{A*R}}$ value will be discussed in Section IV.

If the x node is extended by the y node through the road $y \rightarrow x$, then the recursive formula of the $\mathcal{L}(x)$ can be obtained by $\mathcal{L}(x) = \mathcal{L}(y) \cup \{y \rightarrow x\}$. Then, the recursive formula of the $\mathcal{R}(x)$ can be obtained from the following equation:

$$\begin{aligned} \mathcal{R}(x) &= \mathcal{P} \cap (\mathcal{L}(y) \cup \{y \rightarrow x\}) \\ &= (\mathcal{P} \cap \mathcal{L}(y)) \cup (\mathcal{P} \cap \{y \rightarrow x\}) \\ &= \mathcal{R}(y) \cup (\mathcal{P} \cap \{y \rightarrow x\}) \end{aligned} \tag{16}$$

The recursive formula of the $e(x)$ functions can be obtained from the following equation:

$$\begin{aligned} e(x) &= F_{A*R} \times \sum_{g \rightarrow h \in \mathcal{R}(y) \cup (\mathcal{P} \cap \{y \rightarrow x\})} \Psi_h^g \\ &= e(y) + F_{A*R} \times \Psi_x^y \times \boldsymbol{I}_{\mathcal{P}}(y \rightarrow x) \end{aligned} \tag{17}$$

where $\boldsymbol{I}$ is an indicator function.

Finally, the estimation function $f_2(x)$ of the A^*R algorithm is redefined as

$$f_2(x) = g_2(x) + h_2(x) \tag{18}$$

where $g_2(x)$ and $h_2(x)$ are referred to as the distance function and the heuristic estimate of the A*R algorithm, respectively. They can be derived by

$$g_2(x) = g(x) + \tau \times e(x) \tag{19}$$

$$h_2(x) = h(x) + (1 - \tau) \times e(x) \tag{20}$$

where $\tau \in (0, 1)$; in this paper, $\tau = 0.5$.

Thus, the recursive formula of the $g_2(x)$ can be obtained as follows:

$$\begin{aligned} g_2(x) &= g(y) + l_x^y + \tau \times (e(y) + F_{A*R} \times \Psi_x^y \times \boldsymbol{I}_{\mathcal{P}}(y \rightarrow x)) \\ &= g_2(y) + l_x^y + \tau \times F_{\mathrm{A*R}} \times \Psi_x^y \times \boldsymbol{I}_{\mathcal{P}}(y \rightarrow x) \end{aligned} \tag{21}$$

where l_x^y represents the TT of road $y \rightarrow x$.

3) Advantage of the A*R *Algorithm:* Fig. 5 shows how the A*R algorithm works. At time t-1, vehicle f_1 departs from the B_6 node, and its destination is the D_8 node. According to the congestion index of all the roads at the time t-1, the optimal route calculated by vehicle f_1 by the A*R algorithm is $path_1(B_6 \rightarrow A_4 \rightarrow A_7 \rightarrow D_1 \rightarrow D_8)$.

Suppose that vehicle f_1 arrives at the A_4 node at time t; thus, $\mathcal{P}^{f_1} = \{B_6 \rightarrow A_4\}$, where $\mathcal{P}^{f_1}$ represent the $\mathcal{P}$ of vehicle f_1. At the same time, vehicle f_2 departs from the A_4 node at time t, and the destination is also the D_8 node.

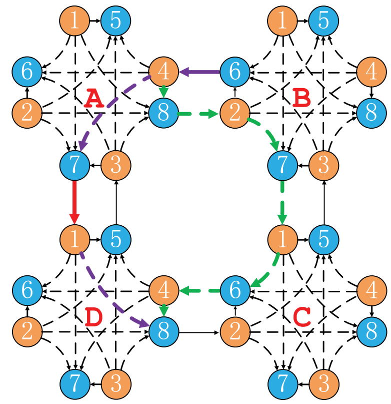


Fig. 5. Principle of A*R.

In addition, for some reason, road $A_7 \rightarrow D_1$ is congested at time t, so the congestion index of $A_7 \rightarrow D_1$ is very large. At time t, there are two paths from the current position (A_4) to the D_8 node, i.e., $path_2(A_4 \rightarrow A_7 \rightarrow D_1 \rightarrow D_8)$ and $path_3(A_4 \rightarrow A_8 \rightarrow B_2 \rightarrow B_7 \rightarrow C_1 \rightarrow C_6 \rightarrow D_4 \rightarrow D_8)$.

For vehicle f_2, $\mathcal{P}^{f_2} = \{\}$, so which path to choose just depends on which path is shorter, that is, the total congestion index of the path is the smallest. Because the congestion index of $A_7 \rightarrow D_1$ is very large, the vehicle f_2 will select the $path_3$ as long as the congestion index of the $path_2$ is sufficiently large.

For vehicle f_1, $\mathcal{P}^{f_1} = \{B_6 \rightarrow A_4\}$ and $O(B_6) = O(B_2) = B$ and $O(A_4) = O(A_8) = A$; therefore, $\Psi_{B_2}^{A_8} = \Psi_{A_8}^{B_2} = \Psi_{A_4}^{B_6} = \psi_{A_4}^{B_6} > 0$. If vehicle f_1 chooses $path_3$, because $path_3$ contains road $A_8 \rightarrow B_2$, vehicle f_1 must suffer not only the total congestion index of $path_3$ but also the penalty of road $A_8 \rightarrow B_2$. Therefore, vehicle f_1 will select $path_2$ as long as the penalty of road $A_8 \rightarrow B_2$ is sufficiently large.

As we all know, many route guidance methods may lead to inevitable congestion by providing the same alternative for multiple vehicles while congestion occurs. From this small example, we see that the A*R algorithm avoids choosing the roads that have been traveled along by imposing penalties on them. Therefore, even if the vehicles have the same current location and destination, as long as the roads they have traveled along are different, the A*R algorithm can provide different routes for these vehicles.

IV. Experiments

In this section, we investigate the performance of the A*R^2 method under various traffic scenarios, and the results are compared with those of other established methods. The specific research questions we are trying to answer in this section are as follows:

- How do parameters (time interval σ, penalty coefficient $F_{\mathrm{A*R}}$, optimal flow coefficient F_{OF}) influence the performance of the proposed method under various traffic scenarios?
- What is the performance from the user's perspective under various scenarios (i.e., minimize TT and TD)?
- What is the performance from the system's perspective under various scenarios (i.e., mitigating congestion)?
- Is the system still robust at low compliance rates (i.e., many drivers are not fully compliant with the route guidance)?

The following subsections will focus on answering the above questions. We start with an assumption of a 100% compliance rate, i.e., all vehicles follow the routes provided by the route guidance method completely. However, the assumption will be relaxed later to study the robustness of the system at low compliance rates.

A. Experiment Setup

The VANET Simulator is used to model the movement of vehicles in the transportation network. The VANET Simulator is an open-source simulator for microsimulation that supports easy adaptation and extension of features. Each vehicle in VANET is simulated individually and makes decisions on its own. Hence, VANET is appropriate for use in our study.

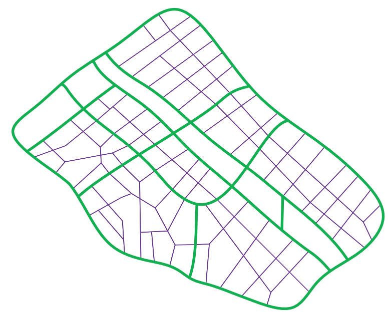


Fig. 6. Yizhuang economic and technological development zone, Beijing, China.

TABLE III
Attribute Values of the Two Road Types

	Main road	Secondary road
lan	3	2
sth	0.9	0.7
$speed\ limit$	$60km/h$	$50km/h$
grd	1	0.9
sfy	0.9	0.7

1) The Transportation Network: First, we introduce the transportation network we used in this study.

The real-world transportation network of the Yizhuang economic and technological development zone of Beijing, China, is used as our experiment network (shown in Fig. 6). The dataset provides the topology of the road network and information about the lanes, lengths, gradients, and speed limits. To simplify our experiment, all roads are set to be bidirectional, and all roads are divided into two types: the main road and secondary road (the thicker lines in Fig. 6 represents the main roads, and the thinner lines represents the secondary roads); furthermore, the same types of roads have the same attribute values except for the length. The associated attribute (defined in TABLE I) values of the two road types are shown in TABLE III.

2) Methods for Comparison: The A*R^2 method is compared with four different route guidance methods: the shortest path (SP) algorithm, dynamic shortest path (DSP) algorithm, dynamic random k shortest path (DRkSP) algorithm, and DTA algorithm.

The SP algorithm will simply calculate the shortest TT route from the origin to destination for each vehicle based on the speed limit. It does not take into account the traffic information and does not provide the re-routing function during the trip.

The DSP is a classical dynamic route guidance method that computes the shortest path based on real-time traffic information about the road network. The advantage of DSP is its simplicity. The service periodically collects real-time traffic condition and then assigns to each vehicle a new current shortest time path. The drawback of DSP is that it only considers the current traffic condition when performing rerouting and does not take into account the road sections

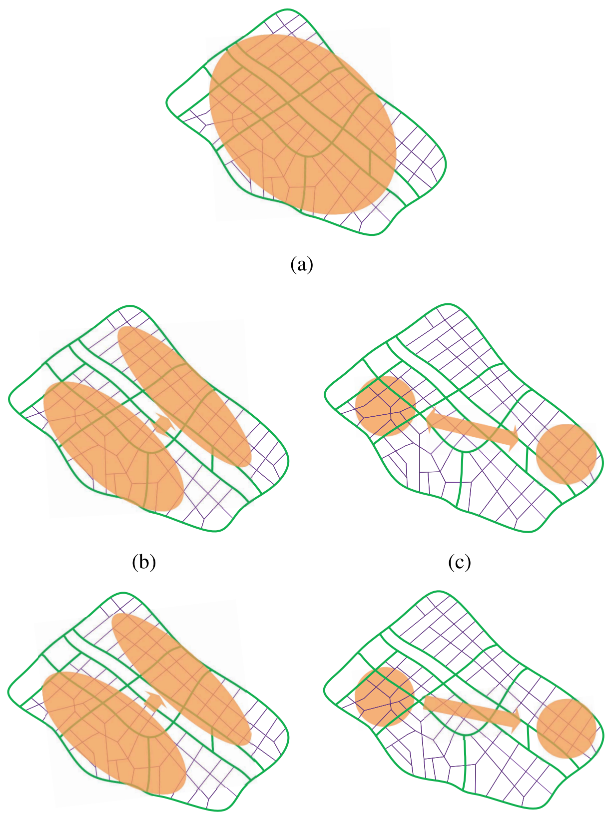


Fig. 7. Traffic scenarios.

on which the car has already traveled. Thus, it is possible to generate new congestion or make detours.

The DRkSP is another classical dynamic route guidance method based on the real-time TT on the road network. It assigns each vehicle to one of the k paths randomly to balance the traffic. In our experiment, to prevent situations in which there is a huge difference between the 'best' path and 'worst' path, we set the limit for the difference between the length of the best path and the worst path as 20%. Thus, the DRkSP method achieves a compromise between the system's perspective and the user's perspective to a certain extent. A fair comparison can therefore be made between it and our A*R^2 method.

In the field of traffic assignment and optimization, DTA models are widely used. The current research on the DTA algorithm can be broadly divided into two categories: simulation-based methods and analytical methods. In this study, simulation-based DTA deployed in [47] was used to achieve user equilibrium through an iterative simulation.

The results of these methods (except those of the SP algorithm) were averaged over 10 simulation runs to take into account the stochastic nature of the simulation.

3) Traffic Scenarios: In order to study whether the A*R^2 method has adaptive ability for different traffic scenarios, we designed several traffic scenarios with different demand patterns.

As shown in Fig. 7, we have five traffic scenarios with different demand patterns. All the information in the OD

TABLE IV

KEY PARAMETERS OF A*R^2

Parameter	Description
σ	time interval; by default σ=20s.
F_{A^*R}	penalty coefficient of A*R algorithm; by default $F_{\mathrm{A^*R}}$=2.
F_{OF}	optimal flow coefficient.

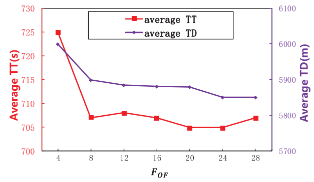
set (departure time, origin, destination, etc. of each vehicle) is randomly generated according to the different demand patterns, and all traffic is evenly generated during the first 5 minutes. In traffic scenario (a), the origin and destination of all vehicles are evenly distributed throughout the road network. In traffic scenario (b), if the origin of a vehicle is on the left part of the road network, its destination will be on the right part of the road network. Conversely, if the origin is on the right part of the road network, its destination will be on the left part of the road network. In traffic scenario (c), if the origin of a vehicle is on the upper left part of the road network, its destination will be on the lower right part of the road network. Conversely, if the origin is on the lower right part of the road network, its destination will be on the upper left part of the road network. In traffic scenario (d), the origin of a vehicle will only be on the left part of the road network, and its destination will only be on the right part of the road network. In traffic scenario (e), the origin of a vehicle will only be on the upper left part of the road network, and its destination will only be on the lower right part of the road network.

B. Optimal Values of Parameters

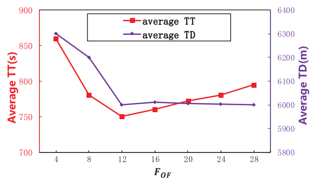
TABLE IV lists the key parameters involved in the A*R^2 method, i.e., the time interval σ, penalty coefficient $F_{\mathrm{A^*R}}$, and optimal flow coefficient F_{OF}. These three parameters have a significant impact on the performance of the A*R^2 method. We conducted several experiments to determine the optimal setting of the parameters. For simplicity, the experimental processes of σ and $F_{\mathrm{A^*R}}$ are not described in detail. In the following experiments, we set σ and $F_{\mathrm{A^*R}}$ as 20s and 2, respectively. This is because when σ and $F_{\mathrm{A^*R}}$ are set to these two values, good results can be achieved under various traffic scenarios.

In traffic scenario (b), we design a variety of OD sets, then set different values for F_{OF}, and calculates the average TT and average TD of all the vehicles. As shown in Fig. 8, the results of the average TT and average TD of 2000, 4000, and 6000 vehicles are given.

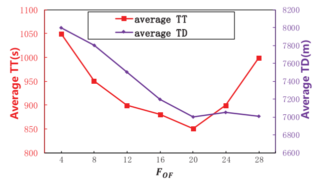
For a small number of vehicles, the average TT and TD first decrease with an increase in F_{OF}; then, they tend to remain constant. This is because when the value of F_{OF} is too small, the transportation network is easily considered to be congested, and the A*R^2 method will guide the vehicle toward routes that are not optimal, resulting in an increase in the TT and TD. When the F_{OF} value is F^*_{OF}, because the number of vehicles is relatively small, most roads cannot meet the congestion conditions prescribed by the A*R^2 method; therefore, most vehicles will still choose the optimal route. When $F_{OF} > F^*_{OF}$, most of roads are hard considered to



(a)



(b)



(c)

Fig. 8. Average TTs and TDs for different F_{OF}. (a) veh=2000. (b) veh=4000. (c) veh=6000.

be congested; therefore, the average TT and TD tend remain constant. When the number of vehicles is 2000, $F^*_{OF} \geq 8$.

In the case of a moderate number of vehicles, the average TT of the vehicle decreases rapidly before an increase in F_{OF} to F^*_{OF}; then, it increases slowly with an increase in F_{OF}. This is because when F_{OF} reaches a certain value, it improves the criteria for judging road congestion. However, when $F_{OF} > F^*_{OF}$, because of the relatively large traffic flow on some roads, congestion actually occurs in the transportation network. When the number of vehicles is 4000, $F^*_{OF} = 12$.

In the case of a large number of vehicles, the traffic flow of the entire transportation network is very large, so F^*_{OF} must be very large, and the average TT has a higher value before F_{OF} increases to F^*_{OF}, for a small F_{OF} will lead to a greater probability of road congestion. When $F_{OF} > F^*_{OF}$, it is easy to generate actual congestion, and the average TT increases rapidly. When the number of vehicles is 6000, $F^*_{OF} = 20$.

C. TT and TD

To verify the performance of the A*R^2 method in reducing the TT, we compare the A*R^2 method with DSP, DR*k*SP, and DTA under different scenarios. In these five traffic scenarios, the size of OD is set to 5000, i.e., the number of vehicles is 5000. As shown in Fig. 9, the A*R^2 method can reduce the average TT of the vehicles to varying degrees regardless of the traffic scenarios because it can obtain better strategies through RL. The DSP method can dynamically update the driving path according to real-time traffic conditions. However, it ignores the impact of the road sections that the car has already traveled along and the future impact of the current rerouting. Therefore, it has a poor performance in some traffic scenarios. Under most traffic scenarios, the average TT of all the methods satisfied DSP>DR*k*SP>DTA>A*R^2. This shows that the A*R^2 method has a strong adaptive ability for different traffic scenarios.

In addition, we also studied the winners and losers in traffic scenario (b), that is, how much time each vehicle saved or wasted against the results of the SP method. We used negative numbers to represent the time wasted and positive numbers to indicate time saved. The results are shown in Fig. 10, which is a histogram, where the horizontal axis represents time segmentation, the vertical axis represents the number of corresponding vehicles, and the dashed vertical line indicates the separation between the time wasted and time saved.

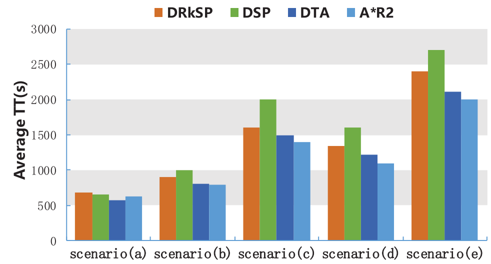


Fig. 9. Average TTs for different traffic scenarios.

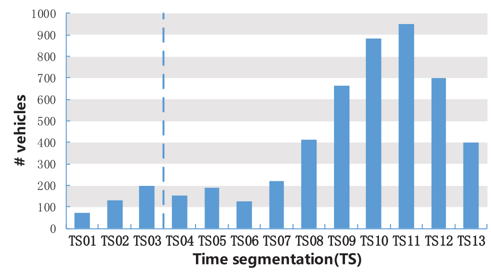


Fig. 10. Winners and losers compared against the results of the SP method. TS01 = [−300, −200), TS02 = [−200, −100), TSX = (−400 + X ∗ 100, −300 + X ∗ 100), . . . , TS13 = [900, 1000).

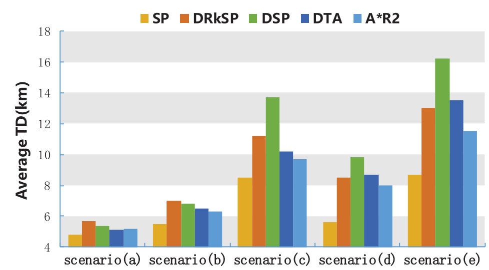


Fig. 11. Average TDs for different traffic densities.

The average TT and average TD are the key indicators for evaluating the route guidance method. Fig. 11 shows a comparison of the TDs of A*R^2 with those of SP, DSP, DR*k*SP, and DTA under different traffic scenarios. The SP algorithm plans the shortest path for all vehicles according to the maximum speed limit, and the path does not change; therefore, the average TD is the smallest. Under most traffic scenarios, the TD of DSP is the largest because only the current state of the transportation network is considered in path planning, regardless of the road sections that the vehicle has already traveled along, which causes the state of the transportation network to oscillate. The DR*k*SP avoids the shortcomings of the DSP algorithm by randomly selecting multiple routes. The A*R^2 method considers the penalty for the roads that the vehicle has traveled along, avoiding their re-selection, thus significantly reducing the average TD.

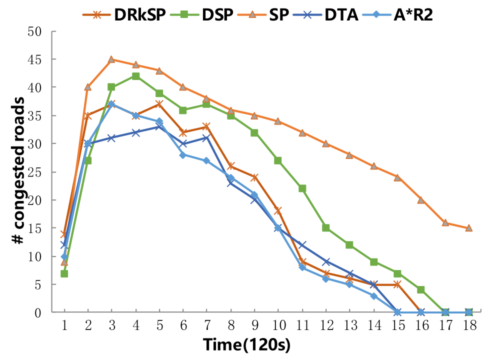


Fig. 12. The number of congested roads detected every 120s.

The experimental results reveal that the A*R^2 method has a strong adaptive ability for different traffic scenarios. Thus, the A*R^2 can increase travel efficiency from the user's perspective under all kinds of traffic scenarios.

D. Traffic Congestion

The objective of the A*R^2 method is to balance the traffic flow on all roads of the network to improve the utilization of road resources and reduce the number of congested roads.

As shown in Fig. 12, by detecting the number of congested roads in the network every 120 s, we have obtained the ability to alleviate the traffic network congestion in traffic scenario (b) with all methods; the size of the OD set is 5000. As shown in Fig. 12, the SP method cannot dynamically adjust the path selection, the number of congested roads is quite high, and the downward trend is very slow. The DSP and DTA algorithms have the ability of re-route the path; therefore, they perform better than the SP algorithm. The A*R^2 method has a higher congestion number than the DTA method in the early period, but because of its adaptive learning ability, its ability to mitigate traffic congestion in the middle and late period is greater than that of the DTA method. This shows that the A*R^2 method can increase traffic efficiency from a system's perspective.

E. Performance at Different Compliance Rates

In this study, we assume that all vehicles are fully compliant with the route guidance methods, but it is unrealistic, because not all vehicles in real life will install route guidance systems; furthermore, some users will not accept routes that do not meet their preferences. The compliance rate is a key factor that must be considered in the design of all dynamic route guidance methods. In this paper, the $x\%$ compliance rate indicates that, in each iteration, the probability that each vehicle participates in the new route calculation (perhaps the new route is the same as the original route) is $x\%$; therefore, the probability of continuing to travel along the original route is 1-$x\%$. In scenario (b), and the size of OD set is 5000, by changing the compliance rate, we measure the average vehicle's TT as an indicator of the performance of each route guidance method.

Fig. 13 indicates that, even under low compliance rates, all the methods except the SP method can significantly improve the average TT. This is because, as long as there are still vehicles that are compliant with the route guidance methods, there will be vehicles that not only use the better route, but also give way to other vehicles.

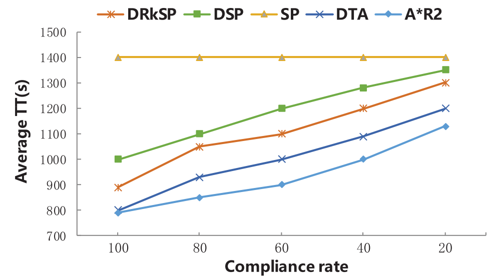


Fig. 13. Average TT varies with the compliance rate.

Regardless of the compliance rate, the A*R^2 method has the best performance among the tested methods. This is because the A*R^2 method can adaptively adjust the congestion index so that the compliant vehicles avoid competing with the non-compliant vehicles for road resources. The compliant vehicles are therefore able to use better routes and reduce the likelihood of congestion.

V. Conclusion

In this study, we have proposed an urban traffic route guidance method with a high adaptive learning ability and demonstrated its ability to reduce congestion, TT, and TD under diverse traffic scenarios (A*R^2). The A*R^2 method adopts the MQ algorithm to assess real-time traffic conditions from the system's perspective based on real-time traffic information. Then, the A*R algorithm is used to search for the optimal route from the user's perspective. The process is repeated periodically during each user's journey.

In future research, we will incorporate the effects of traffic signals on traffic conditions in the network and integrate them into the route guidance system to further improve the method's performance.

Acknowledgements

This work was supported in part by the Zhejiang University/University of Illinois at Urbana-Champaign Institute, and was led by Principal Supervisors Wenbin Hu and Simon Hu.

References

[1] E. Angelelli, I. Arsik, V. Morandi, M. Savelsbergh, and M. G. Speranza, "Proactive route guidance to avoid congestion," *Transp. Res. B, Methodol.*, vol. 94, pp. 1–21, Dec. 2016.
[2] 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.
[3] S. Wang, S. Djahel, Z. Zhang, and J. Mcmanis, "Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2888–2899, Oct. 2016.
[4] 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.
[5] B. Y. Chen, W. H. K. Lam, and Q. Li, "Efficient solution algorithm for finding spatially dependent reliable shortest path in road networks," *J. Adv. Transp.*, vol. 50, no. 7, pp. 1413–1431, Sep. 2016.

[6] I. Kaparias, M. G. H. Bell, K. Bogenberger, and Y. Chen, "Approach to time dependence and reliability in dynamic route guidance," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2039, no. 1, pp. 32–41, Jan. 2007.

[7] U. Demiryurek, F. Banaei-Kashani, C. Shahabi, and A. Ranganathan, "Online computation of fastest path in time-dependent spatial networks," in *Proc. Int. Symp. Spatial Temporal Databases*. Springer, 2011, pp. 92–111.

[8] B. Y. Chen, W. H. K. Lam, A. Sumalee, Q. Li, H. Shao, and Z. Fang, "Finding reliable shortest paths in road networks under uncertainty," *Netw. Spatial Econ.*, vol. 13, no. 2, pp. 123–148, May 2012.

[9] U. Atila, I. R. Karas, C. Gologlu, B. Yaman, and I. M. Orak, "Design of a route guidance system with shortest driving time based on genetic algorithm," in *Proc. 10th WSEAS Int. Conf. Appl. Comput. Appl. Comput. Sci. (ACACOS)*, Stevens Point, WI, USA: World Scientific and Engineering Academy and Society (WSEAS), 2011, pp. 61–66.

[10] Y. Gong, J. Zhang, O. Liu, R. Huang, H. S. Chung, and Y. Shi, "Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 254–267, Mar. 2012.

[11] I. C. Lin and S. Y. Chou, "Developing adaptive driving route guidance systems based on fuzzy neural network," in *Proc. IET 4th Int. Conf. Intell. Environ.*, 2008.

[12] R. Balakrishna, M. Ben-Akiva, J. Bottom, and S. Gao, "Information impacts on traveler behavior and network performance: State of knowledge and future directions," in *Proc. Adv. Dyn. Netw. Model. Complex Transp. Syst.* Springer, 2013, pp. 193–224.

[13] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

[14] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[15] I. L. Aurrecoechea, "Design and validation of novel methods for long-term road traffic forecasting," Ph.D. dissertation, Universidad del País Vasco-Euskal Herriko Unibertsitatea, Leioa, Spain, 2018.

[16] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1786–1796, Jun. 2016.

[17] Z. Liang and Y. Wakahara, "City traffic prediction based on real-time traffic information for intelligent transport systems," in *Proc. 13th Int. Conf. ITS Telecommun. (ITST)*, Nov. 2013, pp. 378–383.

[18] H. Markovič, B. D. Bašic, H. Gold, F. Dong, and K. Hirota, "GPS data-based non-parametric regression for predicting travel times in urban traffic networks," *PROMET-Traffic Transp.*, vol. 22, no. 1, pp. 1–13, Jan. 2012.

[19] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, Aug. 2014.

[20] H.-K. Chen and C.-F. Hsueh, "A model and an algorithm for the dynamic user-optimal route choice problem," *Transp. Res. B, Methodol.*, vol. 32, no. 3, pp. 219–234, Apr. 1998.

[21] D. P. Watling, T. K. Rasmussen, C. G. Prato, and O. A. Nielsen, "Stochastic user equilibrium with a bounded choice model," *Transp. Res. B, Methodol.*, vol. 114, pp. 254–280, Aug. 2018.

[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] S. Lafortune, R. Sengupta, D. E. Kaufman, and R. L. Smith, "Dynamic system-optimal traffic assignment using a state space model," *Transp. Res. B, Methodol.*, vol. 27, no. 6, pp. 451–472, Dec. 1993.

[24] 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.

[25] S. Samaranayake, W. Krichene, J. Reilly, M. L. D. Monache, P. Goatin, and A. Bayen, "Discrete-time system optimal dynamic traffic assignment (SO-DTA) with partial control for physical queuing networks," *Transp. Sci.*, vol. 52, no. 4, pp. 982–1001, Aug. 2018.

[26] L. Yan, W. Hu, and H. Wang, "Dynamic real-time algorithm for multi-intersection route selection in urban traffic networks," *J. Softw.*, vol. 27, no. 9, pp. 2199–2217, 2016.

[27] Z. Cao, H. Guo, and J. Zhang, "A multiagent-based approach for vehicle routing by considering both arriving on time and total travel time," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, pp. 1–21, Dec. 2017.

[28] Z. Cao, H. Guo, J. Zhang, and U. Fastenrath, "Multiagent-based route guidance for increasing the chance of arrival on time," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3814–3820.

[29] T. Yamashita, K. Izumi, K. Kurumatani, and H. Nakashima, "Smooth traffic flow with a cooperative car navigation system," in *Proc. 4th Int. Joint Conf. Auto. Agents Multiagent Syst. (AAMAS)*, 2005, pp. 478–485.

[30] R. Kim, H. Lim, and B. Krishnamachari, "Prefetching-based data dissemination in vehicular cloud systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 292–306, Jan. 2016.

[31] Y. Cao, Z. Sun, N. Wang, M. Riaz, H. Cruickshank, and X. Liu, "Geographic-based spray-and-relay (GSaR): An efficient routing scheme for DTNs," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1548–1564, Apr. 2015.

[32] F. Khodadadi, S. J. Mirabedini, and A. Harounabadi, "Improve traffic management in the vehicular ad hoc networks by combining ant colony algorithm and fuzzy system," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 4, pp. 44–53, 2016.

[33] A. Jahangiri and H. A. Rakha, "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2406–2417, Oct. 2015.

[34] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4592–4601, Jun. 2017.

[35] M. Seshadri, Z. Cao, H. Guo, J. Zhang, and U. Fastenrath, "Multiagent-based cooperative vehicle routing using node pressure and auctions," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–7.

[36] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[37] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman, "Simulation and optimization of traffic in a city," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 453–458.

[38] B. Zhao, C. Zhang, and L. Zhang, "Real-time traffic light scheduling algorithm based on genetic algorithm and machine learning," in *Proc. Int. Conf. Internet Vehicles*. Cham, Switzerland: Springer, 2015, pp. 385–398.

[39] A. Sadek and N. Basha, "Self-learning intelligent agents for dynamic traffic routing on transportation networks," in *Unifying Themes in Complex Systems*, A. Minai, D. Braha, and Y. Bar-Yam, Eds. Berlin, Germany: Springer, 2010.

[40] M. Z. Arokhlo, A. Selamat, S. Z. M. Hashim, and M. H. Selamat, "Route guidance system using multi-agent reinforcement learning," in *Proc. 7th Int. Conf. Inf. Technol. Asia*, Jul. 2011, pp. 1–5.

[41] M. Zolfpour-Arokhlo, A. Selamat, S. Z. M. Hashim, and H. Afkhami, "Modeling of route planning system based on q value-based dynamic programming with multi-agent reinforcement learning algorithms," *Eng. Appl. Artif. Intell.*, vol. 29, pp. 163–177, Mar. 2014.

[42] Y. Chiu, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks, *Dynamic Traffic Assignment: A Primer*. Washington, DC, USA: Transportation Research Board, 2011, p. 52.

[43] M. E. Ben-Akiva, S. Gao, Z. Wei, and Y. Wen, "A dynamic traffic assignment model for highly congested urban networks," *Transp. Res. C, Emerg. Technol.*, vol. 24, pp. 62–82, Oct. 2012.

[44] C. Gawron, "Simulation-based traffic assignment: Computing user equilibria in large street networks," Ph.D. dissertation, Dept. Math. Natural Sci., Univ. Cologne, Cologne, Germany, 1999.

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

[46] D. Wei, "An optimized floyd algorithm for the shortest path problem," *J. Netw.*, vol. 5, no. 12, p. 1496, Dec. 2010.

[47] 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.

Chuanhui Tang is currently a Graduate Student with the School of Computer Science, Wuhan University, Wuhan, China. His research interests include intelligent transport systems, optimization and scheduling, and uncertainty artificial intelligence.

Wenbin Hu is currently a Professor with the School of Computer Science, Wuhan University, Wuhan, China. His research interests include intelligent transport systems, data mining and big data, uncertainty artificial intelligence, and complex networks. He has authored or corresponding authored over 60 refereed journal articles and conference papers, such as the IEEE TKDE, the IEEE TMC, ACM TKDD, SigKDD, ICDM, AAAI, and IJCAI in these areas.

Simon Hu is currently an Assistant Professor with the ZJU-UIUC Institute, Zhejiang University, Hangzhou, China. He is also an Honorary Research Fellow with Imperial College London. His research interests lie in transport telematics, traffic simulation modeling, dynamic route guidance systems, traffic management, and data fusion.

Marc E. J. Stettler is currently a Senior Lecturer with the Centre for Transport Studies, Imperial College London. His main research interests are in intelligent transport systems, vehicle emissions measurement and modeling, and environmental impacts of freight transport.