TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/18 pp91–102
DOI: 10.26599/TST.2020.9010062
Volume 27, Number 1, February 2022

A Dynamic and Deadline-Oriented Road Pricing Mechanism for Urban Traffic Management

Jiahui Jin, Xiaoxuan Zhu, Biwei Wu, Jinghui Zhang*, and Yuxiang Wang

Abstract: Road pricing is an urban traffic management mechanism to reduce traffic congestion. Currently, most of the road pricing systems based on predefined charging tolls fail to consider the dynamics of urban traffic flows and travelers' demands on the arrival time. In this paper, we propose a method to dynamically adjust online road toll based on traffic conditions and travelers' demands to resolve the above-mentioned problems. The method, based on deep reinforcement learning, automatically allocates the optimal toll for each road during peak hours and guides vehicles to roads with lower toll charges. Moreover, it further considers travelers' demands to ensure that more vehicles arrive at their destinations before their estimated arrival time. Our method can increase the traffic volume effectively, as compared to the existing static mechanisms.

Key words: road pricing; traffic congestion alleviation; deep reinforcement learning

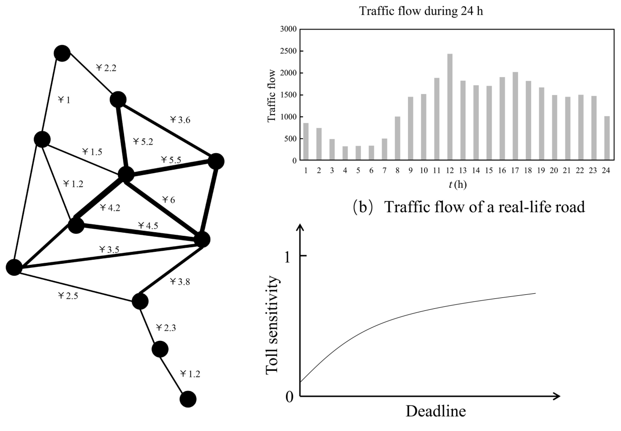
1 Introduction

Large cities nowadays face a challenging problem on urban road networks, which is traffic congestion[1, 2]. China's economic loss caused by traffic congestion accounts for 20% of the disposable income of the urban population, which is equivalent to 5%–8% of the annual gross domestic product loss, reaching 250 billion yuan per year (http://www.cnki.com.cn/Article/CJFDTotal-JSHJ2011S2031.htm). To reduce traffic congestion, immense attention has been given to road pricing mechanism in the urban management field, which aims to divert traffic flows by charging vehicles on busy roads[3]. In this way, vehicles wanting to reduce travel expenses are guided to noncongested and cheaper roads. This scheme is implemented using electronic toll collection and has been successfully applied in some countries or regions.

In implementing road pricing, determining a reasonable price for each road is mostly important[4]. Figure 1a depicts an abstract road network diagram, where vertices represent urban areas, edges represent roads, weights of edges represent road price, and thickness of edges represents traffic volume. On road network, two problems with road pricing arise: First, traffic conditions constantly change and are highly dynamic, especially during unexpected circumstances, e.g., traffic accidents. For example, Fig. 1b illustrates the one-day traffic volume of a real-life road condition in Nanjing, China (http://www.cnki.com.cn/Article/CJFDTotal-JSHJ2011S2031.htm), where traffic flows of the road change from time to time. Thus, dynamic road pricing based on the real-time traffic volume is a necessary action to carry out. Second, driving routes of vehicles are highly time-related. Some people, such as office workers and individuals who have a scheduled flight or train, may have strict time requirements. They must arrive at their destination before the exact time, so they may not mind the traveling expense, while other travelers who do not have time requirements are more willing to choose routes with lower toll charges. Figure 1c depicts how travelers' deadline affects toll sensitivity. When travelers' deadline is fast

• Jiahui Jin, Xiaoxuan Zhu, Biwei Wu, and Jinghui Zhang are with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China. E-mail: {jjin, xxuanzhu, beilwu, jhzhang}@seu.edu.cn.
• Yuxiang Wang is with the Department of Computer Science and Engineering, Hangzhou Dianzi University, Hangzhou 310018, China. E-mail: lsswyx@hdu.edu.cn.
∗ To whom correspondence should be addressed.

Manuscript received: 2020-12-01; accepted: 2020-12-16



(a) Road network of Nanjing

(b) Traffic flow of a real-life road

(c) Toll sensitivity vs. deadline

Fig. 1 An example of road pricing.

approaching, they have lower sensitivity to tolls. On the other hand, when travelers' deadline is far, they tend to have higher sensitivity. Therefore, a dynamic road pricing mechanism is proposed that can set real-time road price based on the current traffic volume on the roads and it considers travelers' time requirements to arrive at the destination at the exact time.

Static and dynamic pricing mechanisms are the existing road pricing mechanisms[5]. Static pricing, which sets fixed tolls on roads, is easy to implement but may not match the traffic dynamics. As for dynamic pricing, in some early works, dynamic pricing is assigned in different tolls to a road on different time slots[6]; however, this may not adapt well to the dynamic traffic environment. Recent machine learning approaches adjust road fees in real time dynamically. An example is DyETC[7] that uses a policy function based on β distribution to solve this problem. However, DyETC has some issues: First, tolls are limited to a range $[0, a_{\max}]$ (e.g., in Singapore, the maximum toll is 6 Singapore dollars), and the β distribution is ineffective when it comes to dealing with bounded toll interval. Hence, it cannot adapt properly to the traffic environment. Second, it does not consider the travelers' time requirement, and thus, cannot be well adapted with the complex and dynamic environment.

In this paper, a novel dynamic road pricing mechanism aiming to optimize traffic congestion and meet travelers' time requirements is proposed. Travelers' demands refer to the requirement of individuals moving in different ways for various purposes. We propose travelers' deadlines to model travelers' demands. The deadlines prompt vehicles to arrive at their destination before a specific time. Vehicles' choice of routes will also be affected by deadlines. We also propose a deep reinforcement learning based algorithm to solve this problem.

In summary, our contributions are as follows:

- Deadlines are used as travelers' demands to adapt to real-life traffic environment. Each deadline of vehicles is allocated randomly to adapt to the changing time requirements in reality. To reduce the scale of state space and speed up calculation, deadlines are assigned at the minute level, and the travelers having the same deadline are regarded as traffic flow to reduce the complexity of the problem.
- To adapt to real-time changing traffic conditions and meet the travelers' time requirements, we propose dynamic and deadline-oriented (DADO), a deep reinforcement learning based algorithm, which uses deep neural networks to simulate the policy network and critic network in the policy gradient algorithm, thereby better representing policy and value functions. In addition, we adopt an asynchronous method to accelerate training speed.
- As the experimental results show, the method we proposed increases the amount of vehicles arriving at the destination before the estimated deadline as compared to other existing mechanisms.

This paper is organized as follows. Section 1 points out the relevant background of road pricing, introduces the issues existing in the current road pricing mechanisms, and puts forward our solutions. Section 2 outlines the existing research work on road pricing mechanisms. Section 3 introduces the modeling process for the road pricing mechanism considering the deadline. Section 4 develops the DADO algorithm in detail. Section 5 offers details of experiments conducted to compare the DADO algorithm with the existing road pricing mechanisms to verify the effectiveness and efficiency of the DADO. Section 6 provides a conclusion to the paper.

2 Related Work

In recent years, road pricing mechanism received immense attention in the urban traffic management field. The research on road pricing mechanism has gone through three stages[8].

The first stage of road pricing mechanism was the original static pricing. Researchers based road price on historical traffic flow data and experience. Joksimovic et al. formulated the optimal toll assignment problem as a bi-level optimization problem[9]. Comparing the effects of fixed tolls and time-varying tolls assigned on the road network, they adopted a simple heuristic search algorithm to determine the optimal toll pattern[8]. Lin et al. proposed an algorithm[10] using a heuristic combining dual variable approximation technique and a method of successive average to determine the toll charges on roads. Zhou et al. assumed that the traffic demand on road networks was known and the vehicles on the road are homogeneous[11]. Except for adopting a two-level iteration method to solve the problem, approximate subgradient projection method for outer-level iteration and partial linearization method for inner-level iteration were proposed.

The second stage was initial dynamic road pricing. Researchers started assigning fine-grained, time-varying tolls; however, a lot of strict assumptions were made about the model. Sharon et al. introduced an efficient tolling scheme and proved the effectiveness of the secheme using a traffic secheme[13]. Based on previous work, Sharon et al. proposed a new dynamic charging scheme, $\varDelta$-tolling. It was the first time a dynamic road pricing model was realized. The toll on each road was distributed proportionally based on the difference between the current driving time of the vehicle and the free driving time without congestion. Since $\varDelta$-tolling was based on the observed traffic flow, $\varDelta$-tolling could adapt to real-time traffic conditions. The calculated locally toll could adapt to large networks. However, they assumed that the parameters on each link are the same[12]. Bui et al. designed a novel mechanism known as user-centric dynamic pricing (UCDP)[14]. This mechanism introduced a fairness constrained shortest path problem with a special structure, thus maximizing social welfare and guaranteeing fairness through polynomial time computation of path allocation.

With machine learning development, various research projects on road pricing mechanism based on reinforcement learning methods have been proposed. Mirzaei et al. defined $\varDelta$-tolling in more details. They considered that the parameters on each link were different. Although it would introduce more parameters and increase training difficulties, they had verified the effectiveness of varying parameters based on reinforcement learning[15]. PG-β adopted reinforcement learning to implement traffic charging for city roads[7] and adopted the policy gradient algorithm. It aims to maximize the number of vehicles that arrives at their destination. PG-β defined a DyETC model, abstracted city links into road network diagrams, adopted the bureau of public roads (BPR) model to explain vehicles' driving time, and then expressed DyETC problem as a Markov decision process problem. Based on DyETC model , PG-β defined several elements in reinforcement learning. Compared with some traditional heuristic algorithms, the trained PG-β alleviates traffic congestion by allocating the road tolls. Although PG-β was one of the few successful examples of reinforcement learning, it had limitations since different time requirements of travelers were not taken into account. Soylemezgiller et al. proposed a radically different road pricing scheme[16], wherein the road price is adjusted dynamically based on the instantaneous traffic densities of each road. In addition, machine learning algorithm is adopted to learn the past usage statistics of the road in order to predict a possible congestion. The mechanism proposed in this paper homogenizes the traffic densities over the entire traffic network. DPG-β[17] proposed deep reinforcement learning to solve the problem of low computational efficiency. The DPG-β adopted a deep neural network to replace the linear representation of PG-β and used temporal difference to replace the Monte Carlo to speed up the update of the target value. However, the research was only limited to travelers who have the same time requirements and did not distinguish among different travelers.

Except for adopting punishment mechanism to alleviate the traffic flow, some research projects use incentive mechanism to divert traffic flow on busy roads. Aung et al. proposed a new congestion pricing system based on reward and punishment policies in a smart city environment[18]. The vehicles were rewarded for voluntarily choosing to take an alternative path to alleviate traffic congestion. Aung et al. also designed a new virtual currency known as T-coin (traffic coin), which is used to reward the vehicles for their positive attitude[18].

3 Problem Formulation

This section outlines the urban traffic environment and builds models for areas, roads, vehicle driving time, vehicle travel costs, and traffic demand.

3.1 Dynamic road pricing problem

Urban environment: We define the dynamic road pricing problem based on the DyETC[7] model. The city is represented as a directed graph network $G = (O, E, U)$. O represents the set of origin-destination pairs, E is the set of roads, and U is the set of urban areas. An OD (origin-destination) pair requires a pair of origin and destination, as well as the transportation needs and all the paths from the origin to the destination in the entire road network. We define the OD pair as a tuple $\langle u_k, u_j, q^t_{k,j}, P_{k,j}\rangle$, where area u_k is origin, u_j is destination, $q^t_{k,j}$ represents the OD demand from u_k to u_j at time step t, and $P_{k,j}$ represents all the paths from u_k to u_j, which do not contain a cycle. $q^t_{k,j}$ is a random variable following a Gaussian distribution function.

Deadlines: Travelers' demands refer to the requirements of individuals moving in different ways for various purposes. Deadlines are used to model the travelers' demands to adapt to real-life traffic environment. The deadlines mean that vehicles must arrive at their destination before an exact time. Many vehicles are on the road during peak hours, and intuitively, each traveler has a different deadline. If each traveler's deadline is processed separately, this problem will result in large data volume and high dimension, making the data processing and training tasks difficult. Therefore, treating travelers with the same deadline as traffic flow is considered.

The government published road tolls during the rush hour of the day; the length of decision-making time is H (minute). We divide H into some integer time intervals. The length of each interval is τ minutes. Thus, continuous time H is discretized to some time periods $t = 0, 1, \ldots, H$. Deadline $d = 0, \ldots, H$ (minute) is assigned to vehicles randomly, and vehicles with the same deadline are treated as traffic flow. For example, if $d = 1$, the traveler's deadline is the first minute in the long decision-making time. Travelers having the same deadline are regarded as a flow to process, which in turn, can reduce the data volume, improve the accuracy of neural network model, reduce the training difficulty, and increase the speed of calculation.

Driving time: The driving time of vehicles whose deadlines are d on road e at time step t is $T^t_e = T^0_e[1 + M(s^t_e/C_e)^N]$, and T^0_e is the free driving time on road e (no congestion), which depends on the length of the road e. M and N are constants in the BPR model; their function is to quantify the impact of the traffic congestion to the vehicles' driving time. s^t_e denotes the amount of vehicles on road e; the more vehicles on the road, the longer the driving time. C_e indicates the capacity of road e.

Travel cost: The travel cost through path $p \in P_{k,j}$ from areas u_k to u_j includes road price cost and time cost. As mentioned before, each traveler has different deadlines. If the current time is still long before the traveler's deadline, the traveler considers the road price and the time cost in choosing a route, but when the current time is very close to the traveler's deadline, the traveler's time cost increases abruptly, leading the traveler to choose the nearest route. Specifically, the travel cost is defined as follows:

$$c^t_{k,j,p,d} = \begin{cases} \sum_{e\in p}(a^t_e + e^{D-x}), & x > D, d \neq 0; \\ (D-x)^2, & x < D, d \neq 0; \\ \sum_{e\in p}(a^t_e + \omega T^t_e), & d = 0 \end{cases} \quad (1)$$

We denote $\left(d - \left(T^t_e + (t-1)\,\tau\right)\right)$ as x, which indicates the time interval between the current time and the traveler's deadline. Furthermore, d is the traveler's deadline, and $T^t_{e,d}$ is the driving time on road e during period t. t represents the current time period. Additionally, $(t-1)\tau$ indicates the vehicles' total driving time from the beginning of the decision-making process. Those vehicles with d equal to zero have no time constraint; hence, they choose route based on the road tolls and the driving time on the road. a^t_e is the road price on the road e. w is the value of time. When $d \neq 0$, the choice will be very different. D is a time threshold, and it is a constant. When $x > D$, it indicates that the remaining time is very abundant, which leads the traveler to choose route based on road toll and the remaining time. With the decrease of x, the cost will increase slowly. When $x < D$, the current time is very close to the deadlines. With the increase of x, the time cost of travelers increases dramatically.

Our traffic flow model is based on the widely adopted stochastic user equilibrium model[7, 19–21], which is $x^t_{k,j,p,d} = \dfrac{\exp\{-w' c^t_{k,j,p,d}\}}{\sum_{p'\in P_{k,j}} \exp\{-w' c^t_{k,j,p',d}\}}$, measuring the proportion of the traffic flow demand via path p from u_k to u_j. $-w'$ is a constant measuring vehicles' sensitivity to traveling cost. When the sensitivity is higher, the cost is greater, making the traffic demand of this route smaller. When the sensitivity becomes smaller, the traffic cost becomes smaller, making the traffic demand of this route greater.

3.2 Reinforcement learning model

The problem we defined is a long time decision problem. We formulate the problem as a discrete-time Markov decision process (MDP), wherein the scale and dimension of state and action are high. Due to the huge scale of MDP, a reinforcement learning based model is used to determine dynamic road pricing. Different from the existing mechanisms, we further consider the traveling deadline to model the travelers' time requirements. The reinforcement learning model has four elements: environment, state, action, and reward function. Figure 2 depicts the reinforcement learning framework of our problem.

State: The state in traffic environment is $s_{e,j,d}^{t}$, which means that the amount of vehicles traveling on road e going to their destination u_j must get there before d. $\boldsymbol{s}_e^t = \langle s_{e,j,d}^{t} \rangle$ is the state vector of road e at time step t, and $\boldsymbol{s}^t = \langle \boldsymbol{s}_e^t \rangle$ is the state matrix of road network G.

Action: The government should publish a reasonable road price at each time step t, so the action is defined as $\boldsymbol{a}^t = \left\langle a_e^t \right\rangle, e \in E$. We assume that all roads have toll facility.

Traffic demand: Traffic conditions and road price change with time; therefore, once a traveler reaches the end of a certain road, there is an incentive to change his path according to the current traffic conditions and tolls at the intersection. The readjusted route does not depend on decisions made by the traveler in the past but only depends on the travelers' specified destination and deadline. Therefore, for vehicles whose deadline falls in d arriving at the end point u_j of the road e, we think it is the new starting point u_k of these vehicles, and the destination of these vehicles is still u_j; their deadlines are still unchanged. In order to distinguish these vehicles from vehicles that really start with u_k, we define as follows:

During time period t, there are two types of total OD demands, namely, primary OD demand and secondary OD demand. The primary OD requirements $q_{k,j,d}^{t}$ from u_k to u_j are the amount of vehicles that hope to start from the u_k at time period t, while the secondary OD demand $q_{k,j,d}^{-t}$ is the amount of vehicles from the adjacent road of u_k to the destination u_j during the time period $t-1$. Vehicles generating these two types of OD demands must arrive at their destination before d. The secondary OD demand pair from u_k to u_j is expressed as $q_{k,j,d}^{-t} = \sum_{e'+=u_k} s_{e',j,d,\text{out}}^{t}$. $s_{e',j,d,\text{out}}^{t}$ is the vehicles leaving the road. Supposing they are proportional to the average speed of vehicles during the time period t, it is formulated as $\dfrac{s_{e,j,d}^{t} \cdot \tau}{T_e^0[1+M(s_e^t/C_e)^N]}$. e^+ represents the terminal point of road e (correspondingly, e^- is the starting point of road e).

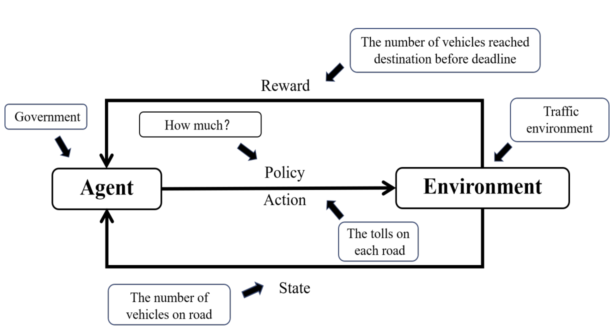


Fig. 2 Reinforcement learning interaction diagram.

State transition: After the traffic equilibrium is formed, the environment can be switched to the next state. The amount of vehicles on road e during time step $t+1$ depends on three parts, expressed as

$$s_{e,j,d}^{t+1} = s_{e,j,d}^{t} - \frac{s_{e,j,d}^{t} \cdot \tau}{T_e^0[1+M(s_e^t/C_e)^N]} + \sum_{u_k=e^- \cap e \in p \in P_{k,j}} \left(q_{k,j,d}^{t} + \sum_{e'+=u_k} s_{e',j,d,\text{out}}^{t} \right) \cdot x_{k,j,p,d}^{t} \tag{2}$$

where $s_{e,j,d}^{t}$ is the amount of vehicles still on road e at last time step t, and this batch of vehicles' deadlines is d. $\dfrac{s_{e,j,d}^{t} \cdot \tau}{T_e^0[1+M(s_{e,d}^t/C_e)N]}$ is the amount of vehicles leaving the road. $\sum_{u_k=e^- \cap e \in p \in P_{k,j}} (q_{k,j,d}^{t} + \sum_{e'+=u_k} s_{e',j,d,\text{out}}^{t}) \cdot x_{k,j,p,d}^{t}$ is the vehicles entering the road. As mentioned earlier, there are two kinds of traffic demands on a road. One is the primary demand, that is, the vehicles taking u_k as the origin and u_j as the destination, and the other is the secondary demand, that is, the traffic entering road e from neighboring roads.

Reward function: The effect of the reinforcement learning model depends on the setting of reward function. With an effort to get a policy aiming to guide vehicles in order to alleviate traffic congestion, we designed several different reward functions, based on which the road pricing mechanism on alleviating traffic congestion is compared.

The effect is measured by the number of vehicles arriving at their destination before deadline. The reward function is defined as follows:

$$R^t(\boldsymbol{s}^t) = \sum_{e \in E} \sum_{u_k=e^+} \frac{s_{e,j,d}^{t} \cdot \tau}{T_e^0[1+M(s_e^t/C_e)^N]}, d \geqslant t \times \tau \tag{3}$$

This reward function measures the number of vehicles arriving at their destination before deadline. The agent will maximize the expected rewards.

$$R^t(\boldsymbol{s}^t) = -\sum_{e \in E} \sum_{u_k=e^+} \frac{s_{e,j,d}^{t} \cdot \tau}{T_e^0[1+M(s_e^t/C_e)^N]}, \quad d < t \times \tau \tag{4}$$

The goal of above reward function is to minimize the number of travelers who fail to arrive at their destination on time with the final strategy.

$$R^t(\boldsymbol{s}^t) = -\sum_{e \in E} \sum_{u_k = e+} \frac{s^t_{e,j,d} \cdot \tau}{T^0_e[1 + M(s^t_e/C_e)^N]} \times (t \times \tau - d), \quad d < t \times \tau \tag{5}$$

This function aims to minimize the time of arrival at the destination beyond travelers' deadlines.

The state value function $v^t(\boldsymbol{s}^t)$ represents the expected total reward obtained by selecting actions according to the strategy π from t to $t + H$:

$$v^t(\boldsymbol{s}^t) = \sum_{t'=t}^{t+H} \gamma^{t'-t} R^{t'}(\boldsymbol{s}^{t'}) \tag{6}$$

where γ is a discount factor.

Policy: At time step t, the policy $\pi^t(\boldsymbol{a}^t|\boldsymbol{s}^t)$ is a conditional probability function, used to determine the probability of selecting an action in a certain state. The optimal policy obtained by training the agent maximizes the value function.

4 DADO Algorithm

Due to the huge scale of the problem, great challenges are faced in finding the optimal policy function. Traditional machine learning methods require a comprehensive understanding of the environment. However, in our defined problem, traffic environment is complicated and constantly changing; it cannot be expressed by mathematical formulas. In addition, the characteristics of samples used by traditional machine learning usually need to be designed by human experts. As known, features have a crucial influence on the generalization of the model; however, traffic environment features cannot be expressed as the environment changes with time. Generally, with the increase of state and action space, the complexity of the reinforcement learning increases exponentially. Traditional reinforcement learning has a two-dimensional Q table presenting the value of state-action pair, but this problem has multidimensional and continuous state space and action space[22].

Hence, function approximators are generally adopted. There are many kinds of approximators such as linear function approximation or nonlinear function approximation. Deep neural networks as nonlinear function approximation have also been used for large reinforcement learning tasks. Deep neural networks have the ability of automatic feature extraction; thus, the use of deep learning is an advantage to represent the agent's observation as an abstract representation in learning an optimal control policy. When faced with the high-dimensional state space and bounded and continuous[23] action space, deep learning will have convergence problems. The policy gradient methods have advantages when dealing with this situation[24, 25]. The policy gradient uses gradient descent methods in finding the optimal policy. Policy gradient does not estimate the value of state-action functions; it learns the policy directly. Based on the above factors, we present our solution algorithm, which is the DADO policy gradient algorithm.

DADO is different from the Monte Carlo based policy gradient algorithm that observes the whole process of an episode and calculates the accumulative reward until the end of the episode. It uses the average reward of all episodes to estimate the reward of the current policy. The total reward from an episode is a random variable, and the Monte Carlo based algorithm uses the sum of the reward, leading to the large variance of the actual cumulative reward obtained by the Monte Carlo algorithm.

DADO adopts temporal difference learning, which has lower variance compared to the Monte Carlo algorithm. More specifically, DADO adopts a structure named "actor-critic". Actor is the policy function aiming to approximate the optimal policy and choose the optimal action, while critic is the value function that evaluates the actor's choice and guides the next choice. We use the advantage function that has a small variance compared with accumulative reward as the evaluation indicator of critic. The formula of advantage function is as follows:

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s) \tag{7}$$

where $V_\pi(s)$ is the sum of the value of all possible actions taken in this state. $Q_\pi(s, a)$ is the action value function corresponding to the action in this state. $Q_\pi(s, a) - V_\pi(s)$ means the advantage of action value function over the current state value function.

The actor aims to maximize the sum of discounted rewards $J(\theta, s_t)$. $J(\theta, s_t)$ is represented as

$$E[\log \pi_\theta(a|s)(Q^\pi(s, a) - V^\pi_s)] \tag{8}$$

where $\log \pi_\theta(a|s)$ is the logarithm of the probability of taking action given the state s. $Q^\pi(s, a)$ is the action value when performing the action a given the state s. V^π_s denotes the state value, and it is the output of the critic. The critic learns the state value, and we use Eq. (9) as loss function. The critic learns to minimize the difference between real action value and estimated value:

$$\text{loss}_{\text{critic}} = (Q^\pi(s, a) - V^\pi_s)^2 \tag{9}$$

The state value of the traditional MDP will not change with time, but in the problem we studied, the amount of vehicles arriving at the destination depends on the specific time step and OD demand of the future time step. In addition, the action also depends on an exact time period. Hence, we updated its value function $v^t(\boldsymbol{s}^t)$ and policy function $\pi^t(a|s,\theta^t)$ for each time period t:

$$\nabla_{\theta^t} v_\pi(\boldsymbol{s}) = Q^t(\boldsymbol{s}^t,\boldsymbol{a}^t)\,\nabla_{\theta^t}\log\pi^t(\boldsymbol{a}^t|\boldsymbol{s}^t,\boldsymbol{\theta}^t) \qquad (10)$$

where $Q^t(\boldsymbol{s}^t,\boldsymbol{a}^t)$ is the action value of executing action under a given state.

The actor training network of DADO uses a deep neural network with three full connection layers to approximate the policy function $\pi^t(\boldsymbol{a}^t|\boldsymbol{s}^t)$. The critic network consists of two full connection layers. Connected parameters are trained separately and updated with the stochastic gradient descent method.

The data network required is independent and distributed. An asynchronous method is adopted that does not produce data at the same time to break the correlation between data and make the convergence easier. Each worker directly takes parameters from the global network and interacts with the environment to output the action. The gradient of each worker is used to update the parameters of the global network. Figure 3 depicts the architecture of the proposed DADO.

At each time step, the local agent extracts state representation from traffic dynamics and puts the state to the actor, and the actor performs an action based on the input state and policy. After that, the critic makes an evaluation of the action and updates the local parameters. After the update, the local agent will push the parameters to the global agent and pull the latest parameter from the global agent.

Algorithm 1 illustrates the processing of a single actor. In Algorithm 1, we assume the global shared network

Algorithm 1 DADO algorithm for each actor learner

```
// Assume global shared parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ and global shared counter $T = 0$
// Assume thread-specific parameter vectors $\theta'$ and $\boldsymbol{\vartheta}'$
Initialize global $\boldsymbol{\theta}^t \leftarrow \boldsymbol{\theta}_0, \boldsymbol{\vartheta}^t \leftarrow \boldsymbol{\vartheta}_0, \forall t = 0,1,\ldots,H-1$;
Initialize step counter $t \leftarrow 1$
repeat
    Reset gradients: $\mathrm{d}\theta_t \leftarrow 0$ and $\mathrm{d}\vartheta_t \leftarrow 0$
    Synchronize thread-specific parameters $\theta'_t = \theta_t$ and $\vartheta'_t = \vartheta_t$
    $t_{\text{start}} = t$
    Get state $s_t$
    repeat
        Choose action $a^t$ with the highest probability according to policy $\pi(a_t|s_t;\theta')$;
        Receive reward $r_t$ and new state $s_{t+1}$
        $t \leftarrow t+1$
        $T \leftarrow T+1$
    until terminal $s_t$ or $t - t_{\text{start}} == t_{\max}$
    $R = \begin{cases} 0, & \text{for terminal } s_t; \\ V(s_t,\vartheta'), & \text{for non-terminal //Bootstrap from last state} \end{cases}$
    for $i \in \{t-1,\ldots,t_{\text{start}}\}$ do
        $R \leftarrow r_i + \gamma R$
        Accumulate gradients $\theta'_t$ :
        $\mathrm{d}\theta_t \leftarrow \mathrm{d}\theta_t + \nabla_{\theta'_t}\log\pi\left(a_i|s_i;\theta'_t\right)\left(R - V\left(s_i;\vartheta'_t\right)\right)$
        Accumulate gradients $\vartheta'_t$ :
        $\mathrm{d}\vartheta_t \leftarrow \mathrm{d}\vartheta_t + \partial\left(R - V\left(s_i;\vartheta'_t\right)\right)^2/\partial\vartheta'_t$
    end for
    Perform asynchronous update of $\theta_t$ using $\mathrm{d}\theta_t$ and of $\vartheta_t$ using $\mathrm{d}\vartheta_t$
until $T > T_{\max}$
```

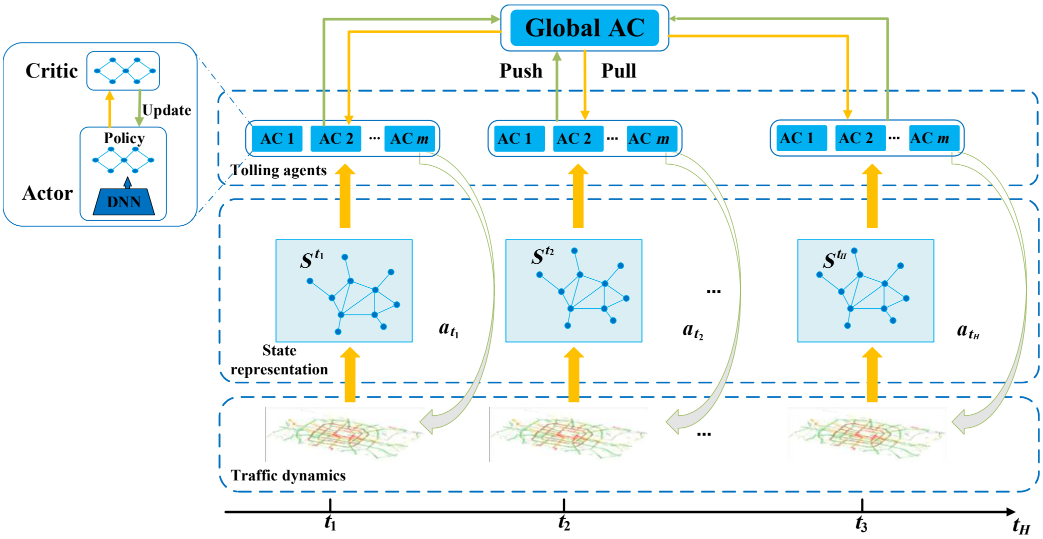


Fig. 3 Architecture of the proposed DADO.

parameters are vectors $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$. The thread-specific parameter vectors are $\boldsymbol{\theta}'$ and $\boldsymbol{\vartheta}'$. The input requires the decision-making horizon H and the global shared counter T. We need to initialize the step counter t to perform an advanced update. Later, the algorithm starts the loop. In the loop, the agent resets the gradients, pulls the global parameter assignments to the local network, and starts a new episode. Post that, the agent interacts with the environment, and the action is selected based on the policy function $\pi\ (a_t|s_t;\theta')$. The agent will receive immediate rewards which are computed using Eqs. (3)–(5), and the environment will transform to the next state. If an episode has ended or the number of steps calculated in advance has been reached (Line 15), the algorithm then begins to calculate the total discount reward and updates the parameters at each time period with stochastic gradient descent method. R means the sum of rewards from time t_{start} to t obtained from the interactions, representing the "real" value based on the policy. $V(s_i;\vartheta')$ is the "estimated" sum of rewards approximated by the critic. After updating the local parameters, the local parameters will be pushed to the global network to update the global net. The whole process iterates until the number of episodes reaches a predefined global counter $T_{\max}$.

5 Evaluation

The effectiveness of our approach is evaluated using simulations performed on a PC with Intel CPU (I7 87000K), 16 GB memory, and NVIDA GPU (2080Ti). Simulation data and parameters are as follows.

Simulation data: The simulation data are generated based on the "2019 Main Population Data of Nanjing Regions" published by the Nanjing Municipal Bureau of Statistics, containing the population of Qinhuai, Jiangning, Jianye, and Qixia districts. A road network connecting these districts (see Fig. 4) is used in our evaluation.

We use the amount of vehicles per person in different urban areas to estimate the real origin-destination demands of Nanjing. The regional population is 3.5415 million, and the total number of vehicles is 2.58 million; hence, the amount of vehicles per person has is $258/354 = 0.73$.

At the initial time period t, the amount of vehicles on the road that falls in each d $(d = 0, 1, \dots, H)$ is randomly generated by random function within the range of $0.5 - 0.7$ of road capacity. The primary OD demand is generated by a function of time, where the demand at time period $t = 0$ is the lowest, reaches the peak in the middle of the decision time H, and then gradually tends to a lower level. This specialty is realized via a Gaussian distribution function. The peak traffic demand for each origin-destination pair is randomly generated within $[8, 12]$ vehicles per minute. We set the OD demand at $t = 0$ (usually starting at peak time) to 60% of the peak OD demand.

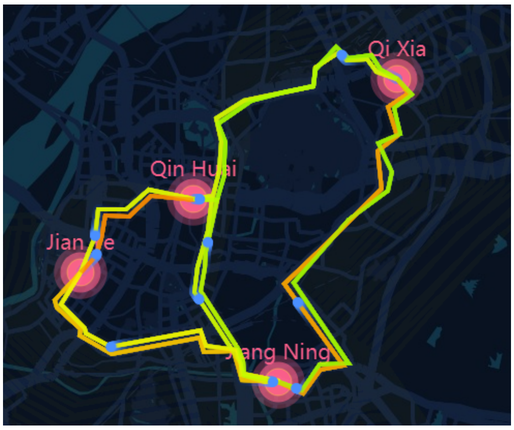


Fig. 4 Road network used by simulations.

Parameter setting: Table 1 showcases the parameter setting of the simulations. The actor and critic's learning rate are set as 10^{-10} and 10^{-7}, respectively, which decide the fineness of learning. Discount factor γ assigns different weights to different time periods. The amount of episodes of training is 1000, and there are 10 agents interacting with the environment at the same time. The number of urban areas U is set to 4, and roads are set to $|E| = 10$. Advanced update method is adopted; hence, the local agent will update parameters every five steps. Furthermore, the constants M and N in driving time definition are set to 0.15 and 4, respectively. We determine that the max toll on each road is 6 Yuan according to Singapore's road tolls. Decision time (usually the peak time) is set as 360 min, and each time period τ is 10 min. The value of time of travelers with no time requirements is set as 0.5.

Table 1 Parameter setting.

Parameter	Value	Parameter	Value
Learning rate of actor	10^{-10}	Period length τ	10 min
Learning rate of critic	10^{-7}	Update step	5
Discount factor γ	0.9	Number of local agents	10
Training episodes	1000	Number of areas	4
Max toll	6	Number of roads	10
Decision time H	360 min	N	4
M	0.15	Cost sensitivity w'	0.5
Value of time w	0.5		

5.1 Comparison with other mechanisms

We compare DADO based on three policies trained with existing road pricing mechanisms. The existing mechanisms are as follows:

(1) Fixed[26]. This road pricing mechanism allocated fixed fees on each road through assuming that fees are proportional to average traffic demand on the road at each time step. The traffic demand is estimated based on historical traffic data.

(2) *Δ*-tolling[13]. The tolls were allocated in proportion to the difference between the current driving time and the free driving time of vehicles on each road.

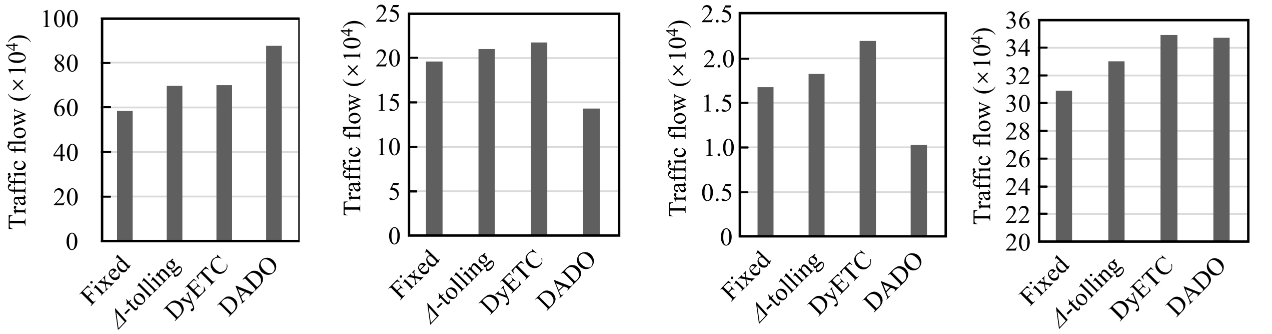
(3) DyETC[7]. The road pricing policy adopted beta distribution to balance the "exploitation" and "exploration" when choosing actions.

Results: The policy of DADO in Figs. 5a and 5d is trained based on Eq. (3). Figure 5b is trained based on Eq. (4), and Fig. 5c is trained based on Eq. (5). Figure 5 is the comparison result of different mechanisms under different measuring standards. Figures 5a–5d indicate that DADO performs better than the other three mechanisms under different measuring standards. Figure 5a shows that DyETC, which is also based on reinforcement learning, is not sensitive in time dimension and cannot handle the time requirements of vehicles, leading to the same effect of guiding the traffic flow as the heuristic *Δ*-tolling. Figures 5b and 5c show that when processing the time dimension, DADO performed very well. The number of vehicles that fails to arrive on time and the time beyond deadlines are at the minimum compared with others. Figure 5d is the total traffic flow without considering time dimension. It shows that when time dimension is not considered, our proposed DADO performs a little worse when compared with DyETC. In the real-life traffic environment, the agent would give more consideration about the environment, so the effect would be less than DyETC with no time consideration.

5.2 Effects of different traffic conditions

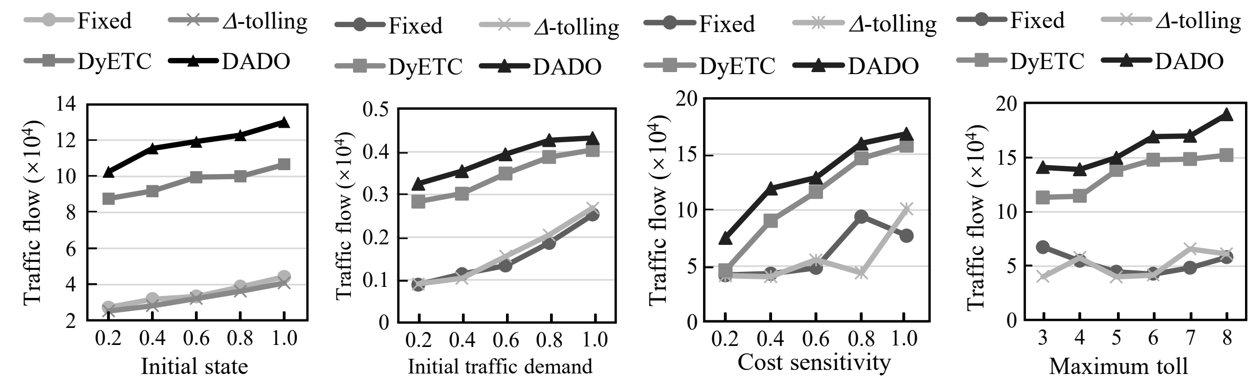
We compare DADO trained by Eq. (3) with the existing road pricing mechanisms under different traffic conditions. The results are shown in Fig. 6.

Results: The optimization objective is to maximize the number of vehicles arriving at destination before the deadline. Figure 6 depicts the amount of vehicles arriving at destination before the deadline for different



(a) Number of vehicles arriving on time (b) Number of vehicles failing to arrive on time (c) Time out of failing to arrive on time (d) Number of vehicles arriving destination

Fig. 5 Different mechanism of different measuring standards.



(a) Initial state (b) Initial traffic demand (c) Cost sensitivity (d) Maximum toll

Fig. 6 Amount of vehicles arriving destinations before deadlines.

road pricing mechanisms under different traffic settings. The *y*-axis is the amount of vehicles arriving at their destination before deadline; the *x*-axis is for parameters to be evaluated, and its value represents the scaling ratio compared to the original setting. Figure 6a illustrates how the traffic flow increases with an increase of the initial amount of vehicles and DADO performs better in different scales of initial state. The effect of two heuristic algorithms on traffic flow is not good; we analyze the reason for it as that in the subsequent decision-making process, too many vehicles are imported into the roads and the heuristic algorithm cannot adapt well to the dynamically changing traffic environment. Figure 6b depicts initial traffic demand. With an increase in traffic demand, all the mechanisms are increased. DADO performs better than the other mechanisms. The two heuristic algorithms perform poorly than the others. We hold the opinion that fixed scheme and *Δ*-tolling scheme cannot adopt to large-scale traffic demand. Figure 6c depicts that as the cost sensitivity of vehicles increases, the traffic flow of road pricing mechanisms also increases. When the traveling cost of a busy road increases, the vehicles will divert to the less busy and cheaper road, which can reduce the traveling cost. Figure 6d depicts the better adjustment ability of DADO under different maximum tolls.

As can be seen in Figs. 6a–6d, we can observe that under different traffic parameter settings, DADO mechanism's performance is still optimal, and DyETC mechanism's effect is slightly worse than the DADO mechanism. However, these two road pricing mechanisms perform better than other mechanisms. It can be deduced from Fig. 6 that the performance of heuristic algorithm under different traffic conditions is not as good as that of the reinforcement learning algorithm. We analyze that heuristic algorithm cannot adapt to the dynamic traffic environment with time dimension, hence the poor performance.

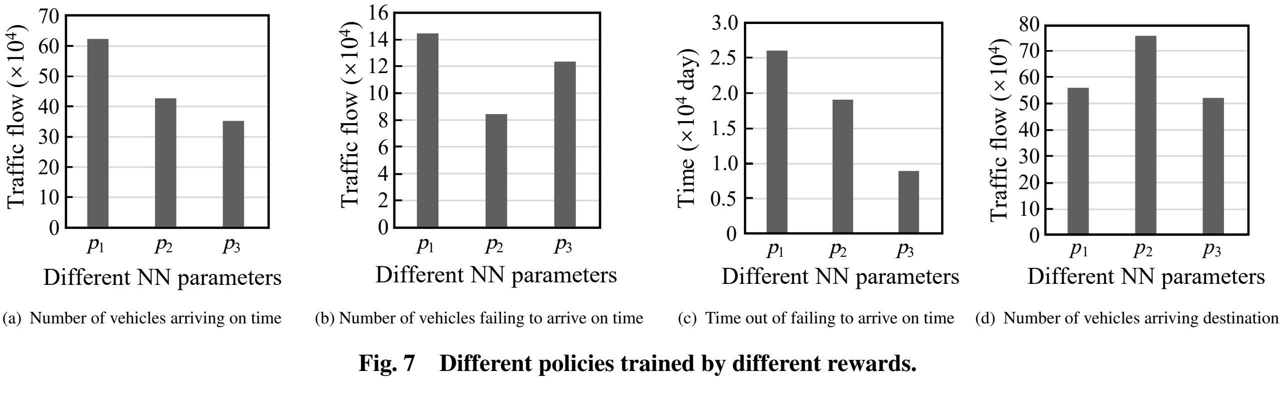
5.3 Effects of different policies

Setting reward functions plays an important role in policy learning. To verify the effectiveness, we trained three sets of policy (parameters) under three predefined reward functions.

Results: Figure 7 depicts the experimental result. p_1 is for parameters trained based on the reward function that maximizes the number of vehicles arriving on time; this policy will make more vehicles arrive at their destination on time when the agent adopts this policy. p_2 is for parameters trained based on the reward function that minimizes the number of vehicles failing to arrive on time, and this policy will minimize the number of vehicles arriving over time. p_3 is for parameters trained based on the reward function that minimizes the time of failure to arrive on time. As can be seen in Figs. 7a–7c, for different goals, each of the training policies trained by the specific reward function performs better than the others. For example, from Fig. 7a, *y*-axis is the number of vehicles arriving on time, and *x*-axis is different policies. p_1 performs better than the others, and Figs. 7a–7c prove the effectiveness of the reward function we designed. In Fig. 7d, *y*-axis is the number of vehicles arriving at the destination without considering whether the vehicles arrive on time. Because we want to know which policy is the most effective in reducing traffic congestion on a macro level, we can see that policy 2 is better on alleviating traffic congestion without time consideration.

6 Conclusion

In this paper, we design a reinforcement learning model and a neural network model based for dynamic road pricing. The DADO algorithm based on the policy gradient algorithm is proposed. In the study, the road pricing mechanisms trained by the DADO algorithm are compared with the other three existing road pricing



(a) Number of vehicles arriving on time (b) Number of vehicles failing to arrive on time (c) Time out of failing to arrive on time (d) Number of vehicles arriving destination

Fig. 7 Different policies trained by different rewards.

mechanisms to verify the effectiveness of DADO algorithm. Although this paper considers the difference in time value of different travelers, the varied money sensibility of different travelers is not taken into account. For example, travelers with high wages may be more concerned about time costs, while travelers with low wages are more concerned about money costs. This factor will be considered in our future work.

Acknowledgment

This work was supported by the National Key R&D Program of China (No. 2018AAA0101200), National Natural Science Foundation of China (Nos. 62072099, 61972085, 62072149, and 61872079), Public Welfare Research Program of Zhejiang (No. LGG19F020017), Jiangsu Provincial Key Laboratory of Network and Information Security (No. BM2003201), Key Laboratory of Computer Network and Information Integration of Ministry of Education of China (No. 93K-9), "Zhishan" Scholars Programs of Southeast University, and partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Fundamental Research Funds for the Central Universities. We also thank the Big Data Computing Center of Southeast University for providing the experiment environment and computing facility.

References

[1] S. Liu, W. T. Zhang, X. J. Wu, S. Feng, X. Pei, and D. Y. Yao, A simulation system and speed guidance algorithms for intersection traffic control using connected vehicle technology, *Tsinghua Science and Technology*, vol. 24, no. 2, pp. 160–170, 2019.

[2] N. Wang, G. Guo, B. Wang, and C. Wang, Traffic clustering algorithm of urban data brain based on a hybrid-augmented architecture of quantum annealing and brain-inspired cognitive computing, *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 813–825, 2020.

[3] J. J. Li, H. H. Jiao, J. Wang, Z. G. Liu, and J. Wu, Online real-time trajectory analysis based on adaptive time interval clustering algorithm, *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 131–142, 2020.

[4] J. R. Gan, B. An, H. Z. Wang, X. M. Sun, and Z. Z. Shi, Optimal pricing for improving efficiency of taxi systems, in *Proc. 23rd Int. Joint Conf. Artificial Intelligence*, Beijing, China, 2013, pp. 2811–2818.

[5] M. J. Hausknecht and P. Stone, Deep reinforcement learning in parameterized action space, arXiv preprint arXiv:1511.04143, 2016.

[6] D. Joksimovic, M. C. J. Bliemer, P. H. L. Bovy, and Z. Verwater-Lukszo, Dynamic road pricing for optimizing network performance with heterogeneous users, in *Proc. Networking*, *Sensing and Control*, Tucson, AZ, USA, 2005, pp. 407–412.

[7] H. P. Chen, B. An, G. Sharon, J. P. Hanna, P. Stone, C. Y. Miao, and Y. C. Soh, DyETC: Dynamic electronic toll collection for traffic congestion alleviation, in *Proc. 32nd AAAI Conf. Artificial Intelligence*, New Orleans, Lousiana, USA, 2018, pp.757–765.

[8] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in *Proc. 12th Int. Conf. Neural Information Processing Systems*, Cambridge, MA, USA, 1999, pp. 1057–1063.

[9] D. Joksimovic, M. C. J. Bliemer, and P. H. L. Bovy, Optimal toll design problem in dynamic traffic networks with joint route and departure time choice, *Transportation Research Record*, vol. 1923, no. 1, pp. 61–72, 2005.

[10] D. Y. Lin, A. Unnikrishnan, and S. T. Waller, A dual variable approximation based heuristic for dynamic congestion pricing, *Networks and Spatial Economics*, vol. 11, no. 2, pp. 271–293, 2011.

[11] B. J. Zhou, M. Bliemer, H. Yang, and J. He, A trial-and-error congestion pricing scheme for networks with elastic demand and link capacity constraints, *Transportation Research Part B*: *Methodological*, vol. 72, pp. 77–92, 2015.

[12] G. Sharon, J. Hanna, T. Rambha, M. Albert, P. Stone, and S. D. Boyles, Delta-tolling: Adaptive tolling for optimizing traffic throughput, in *Proc. 9th Int. Workshop on Agents in Traffic and Transportation*, New York, NY, USA, 2016.

[13] G. Sharon, J. P. Hanna, T. Rambha, M. W. Levin, M. Albert, S. D. Boyles, and P. Stone, Real-time adaptive tolling scheme for optimized social welfare in traffic networks, in *Proc. 16th Conf. Autonomous Agents and MultiAgent Systems*, Richland, SC, USA, 2017, pp. 828–836.

[14] K. T. Bui, V. A. Huynh, and E. Frazzoli, Dynamic traffic congestion pricing mechanism with user-centric considerations, in *Proc. 2012 15th Int. IEEE Conf. Intelligent Transportation Systems*, Anchorage, AK, USA, 2012, pp. 147–154.

[15] H. Mirzaei, G. Sharon, S. D. Boyles, T. Givargis, and P. Stone, Link-based parameterized micro-tolling scheme for optimal traffic management, in *Proc. 17th Int. Conf. Autonomous Agents and MultiAgent Systems*, Richland, SC, USA, 2018, pp. 2013–2015.

[16] F. Soylemezgiller, M. Kuscu, and D. Kilinc, A traffic congestion avoidance algorithm with dynamic road pricing for smart cities, in *Proc. 2013 IEEE 24th Annual Int. Symp. Personal*, *Indoor*, *and Mobile Radio Communications*, London, UK, 2013, pp. 2571–2575.

[17] W. Qiu, H. P. Chen, and B. An, Dynamic electronic toll collection via multi-agent deep reinforcement learning with edge-based graph convolutional networks, in *Proc. 28th Int. Joint Conf. Artificial Intelligence*, Macao, China, 2019, pp. 4568–4574.

[18] N. Aung, W. D. Zhang, S. Dhelim, and Y. B. Ai, T-Coin: Dynamic traffic congestion pricing system for the internet of vehicles in smart cities, *Information*, vol. 11, no. 3, p. 149, 2020.

[19] H. K. Lo and W. Y. Szeto, A methodology for sustainable traveler information services, *Transportation Research Part B*: *Methodological*, vol. 36, no. 2, pp. 113–130, 2002.

[20] K. L. Hong, C. W. Yip, and K. H. Wan, Modeling transfer and non-linear fare structure in multi-modal network, *Transportation Research Part B: Methodological*, vol. 37, no. 2, pp. 149–170, 2003.

[21] H. J. Huang and Z. C. Li, A multiclass, multicriteria logit-based traffic equilibrium assignment model under ATIS, *Eur. J. Oper. Res.*, vol. 176, no. 3, pp. 1464–1477, 2007.

[22] K. Zhu and T. Zhang, Deep reinforcement learning based mobile robot navigation: A review, *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021.

[23] B. J. Zhou, M. Bliemer, H. Yang, and J. He, A trial-and-error congestion pricing scheme for networks with elastic demand and link capacity constraints, *Transportation Research Part B: Methodological*, vol. 72, pp. 77–92, 2015.

[24] O. Anschel, N. Baram, and N. Shimkin, Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning, in *Proc. 34th Int. Conf. Machine Learning*, Sydney, Australia, 2017, pp. 176–185.

[25] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, Continuous control with deep reinforcement learning, in *Proc. 4th Int. Conf. Learning Representations*, San Juan, Puerto Rico, 2016.

[26] P. A. Barter, A vehicle quota integrated with road usage pricing: A mechanism to complete the phase-out of high fixed vehicle taxes in Singapore, *Transport Policy*, vol. 12, no. 6, pp. 525–536, 2005.

Jiahui Jin is an associate professor at the School of Computer Science and Engineering, Southeast University, Nanjing, China. He received the PhD degree in computer science from Southeast University in 2015. He was a visiting PhD student at University of Massachusetts, Amherst, USA, during August 2012 to August 2014. His current research interests include large-scale data processing and urban computing.

Xiaoxuan Zhu received the BS degree from Hohai University, China in 2020. She is pursuing the master degree in electronic information at Southeast University, China. Her research interests include urban computing.

Biwei Wu received the BS degree in software engineering from Southeast University, Nanjing, China in 2019. Currently, he is pursuing the MS degree in computer science and engineering at Southeast University, Nanjing, China. His current research interests include game theory and urban computing.

Jinghui Zhang received the BS degree from Southeast University in 2005, and the PhD degree in computer science from Southeast University in 2014. He is an associate professor at School of Computer Science and Engineering, Southeast University, China. His current research interests include cloud computing, edge computing, and distributed machine learning.

Yuxiang Wang received the BE degree in software engineering from Tianjin University in 2014, and the PhD degree in computer science from Southeast University in 2015. He is currently an assistant professor at the Department of Computer Science and Engineering, Hangzhou Dianzi University, Hangzhou, China. His current research interests include knowledge graph query, approximate query processing, and query optimization.