Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6616702, 14 pages
https://doi.org/10.1155/2021/6616702

Research Article

Adaptive Optimization of Traffic Signal Timing via Deep Reinforcement Learning

Zibo Ma,[1,2] **Tongchao Cui**,[1,2] **Wenxing Deng**,[1,2] **Fengyao Jiang**,[1,2] **and Liguo Zhang**[1,2]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China*

Correspondence should be addressed to Zibo Ma; mazibo@emails.bjut.edu.cn

Received 26 October 2020; Revised 7 September 2021; Accepted 15 October 2021; Published 27 November 2021

Academic Editor: Jing Zhao

With rapid development of the urbanization, how to improve the traffic lights efficiency has become an urgent issue. The traditional traffic light control is a method that calculates a series of corresponding timing parameters by optimizing the cycle length. However, fixing sequence and duration of traffic lights is inefficient for dynamic traffic flow regulation. In order to solve the above problem, this study proposes a traffic light timing optimization scheme based on deep reinforcement learning (DRL). In this scheme, the traffic lights can output an appropriate phase according to the traffic flow state of each direction at the intersection and dynamically adjust the phase length. Specifically, we first adopt Proximal Policy Optimization (PPO) to improve the convergence speed of the model. Then, we elaborate the design of state, action, and reward, with the vehicle state defined by Discrete Traffic State Encoding (DTSE) method. Finally, we conduct experiments on real traffic data via the traffic simulation platform SUMO. The results show that, compared to the traditional timing control, the proposed scheme can effectively reduce the waiting time of vehicles and queue length in various traffic flow modes.

1. Introduction

The management of urban road intersections is mainly achieved by controlling traffic lights. However, ineffective control of traffic lights will bring numerous problems, such as long delays for passengers, large energy waste, and even traffic accidents [1, 2]. Early traffic lights control either deployed a fixed program without considering real-time traffic or considered very limited dimensions of traffic [3], such as timing control and induction control. Timing control generally uses the Webster timing method, which chooses the optimal cycle time by applying the minimum traffic delay and makes the green time ratio proportionally distributed by the maximum flow ratio of each phase. Induction control measures the traffic flow by presetting coils at the entrance of each lane and meets the traffic demand by adjusting the green time ratio of the cycle [4]. Besides, control methods such as fuzzy control [5], queuing theory-based method [6], and model-based method [7, 8] are also used in traffic lights control. Although the above schemes can optimize the traffic flow to some extent, the actual effects are not satisfactory enough due to the lack of adaptability, strong dependency of experiment, and other factors [9].

With the vigorous development of deep learning in artificial intelligence, the researches in the field of adaptive traffic lights control have become more and more deep [10]. Meanwhile, the coming of big data era makes the way to extract traffic features becoming more and more diverse, so that using methods based on DRL can take advantage of data and dig the connection between different kinds of data better. Therefore, many AI based control methods have emerged. Prashanth and Bhatnagar [11] proposed using the queue length and the current phase duration as the states and approximate the Q value with a linear function. Liu et al. [12] proposed a cooperative signal control system based on reinforcement learning. This scheme proposes cluster vehicles and uses a linear function to approximate the Q value, with the state input only considering vehicle queue information. Li et al. [13] took the length of the queue in each direction of the intersection as input and applied a stacked

autoencoder which estimates the Q function of the deep neural network. Most of the above schemes use the length of the vehicle queue as the input state. However, this single-dimensional input will miss some important traffic information, resulting in the agent's inability of fully perceiving environmental information and will affect the final decision-making effect.

In recent years, the data processing capabilities of computers have been significantly improved, and more and more new reinforcement learning methods have been proposed by scholars. Konda and Tsitsiklis [14] proposed the Actor-Critic method, in which the Actor network is introduced to select actions and the Critic network is introduced to judge the value of the action. Xu et al. [15] used the DRQN algorithm which collects the number of vehicles, average speed, and traffic light status at the intersection as the state. Experimental results show that, compared with the traditional timing control, the proposed scheme reduces the average vehicle delay and travel time. Wade and Saiedeh [16] proposed using an asynchronous Q-learning algorithm that takes vehicle queue density, queue length, and current traffic light phase as inputs. The results show that the average vehicle delay is reduced by 10% under the condition of constant traffic flow. However, reinforcement learning is a rapidly developing field, and more and more new methods are proposed, for example, DDPG, A2C, A3C, and PPO. These new methods have better learning efficiency and convergence [17, 18].

Using reinforcement learning which regards the traffic light as an agent to explore reasonable behaviours through interacting with the environment has been favoured by more and more scholars. Alegre et al. [19] applied the learning method of Q-learning and used the difference in the accumulated waiting time of vehicles before and after the execution of the action as a reward function to update the decision parameters. Ge et al. [20] proposed a cooperative deep Q network (QT-CDQN) with Q value transmission. In QT-CDQN, the intersection in the area is modelled as an agent reinforcement learning system, and the change in the average queue length of vehicles is used as the reward. Liang et al. [21] used the DQN method to control the traffic lights phase which quantifies complex traffic scenes as simple states by dividing the entire intersection into small grids. The definition of the reward in this method is the cumulative waiting time difference between two cycles. In order to realize the model, convolutional neural network is introduced to map the state to the reward. On the whole, the main focus of the control schemes mentioned above is to maximize the throughput of intersections without consideration of safety factors.

This study has done the following three main works. Firstly, regarding the problem of limited input dimensions, we choose the vehicle state and the road state as inputs in order to increase the dimension of the state space and improve the decision-making performance of the signal light controller. The vehicle state is gathered by traffic cameras. Vehicle distribution images are obtained by shooting intersection roads; then a computer is used to build a vehicle spatial information matrix. The elements in the matrix reflect vehicle state, including speed, position, and direction. Secondly, with regard to the reinforcement learning algorithm, we select the PPO algorithm based on policy gradient to train the traffic light control policy. Thirdly, with regard to the issue that the reward equation only takes the traffic flow at the intersection into account, we formulate a maximum tolerable green time and design a reward equation that includes the green time. By detecting the green time and calculating the difference with the maximum tolerable green time, this equation will output a negative reward value which prevents the action from happening again when the actual green time is excessively long.

The following contents of this article are arranged as follows. The methodology part of the second chapter first briefly describes the process and components of reinforcement learning (RL). Then, the modelling process of turning traffic light control into RL is illustrated in detail, and the definitions of each element in the learning model are clarified. Finally, the composition of the traffic light decision-making network and the parameter update process of the entire DRL system are introduced. In the third chapter, the experiment introduces the construction of the simulation environment of SUMO [22] traffic simulation software for traffic light control, including road model, traffic light configuration, and vehicle attributes in simulation. Then, the experiment proves the effectiveness of the proposed scheme in this study and compares it with traditional timing control, which reveals the advantages of this method. Finally, the fourth chapter serves as conclusion which summarizes the content and core work of the full research and puts forward the outlook for the unresolved problems and the parts that can be optimized.

2. Methodology

2.1. System Framework. The ideal traffic light control should response dynamically to traffic flow and can adjust the output signal phase in real time [23]. This study proposes using the RL method to learn from the traffic flow in all directions of the intersection and then optimizes the phase time and sequence. The RL framework of this study is shown in Figure 1, which is mainly composed of two parts, namely, the agent and the environment. The environment part is simulated by traffic simulation software SUMO. The agent is built by a neural network and has the ability of perceiving the environment and output actions.

The workflow of RL is that the agent is based on the current environment state s_t; after taking an action a_t, the environment gets the next state s_{t+1}. At the same time, the environmental reward r_{t+1} obtained by taking the action a_t is fed back to the agent, so that the agent can adjust and improve the strategy according to the feedback reward while exploring [24]. After the above process is repeated many times, the agent finally finds the optimal strategy for the environment by adjusting its strategy continuously.

The RL model can be defined by three important elements $<S, A, R>$, where S is the environment state space, A is the agent action space, and R is the reward equation. For this intelligent traffic light control system, the environmental

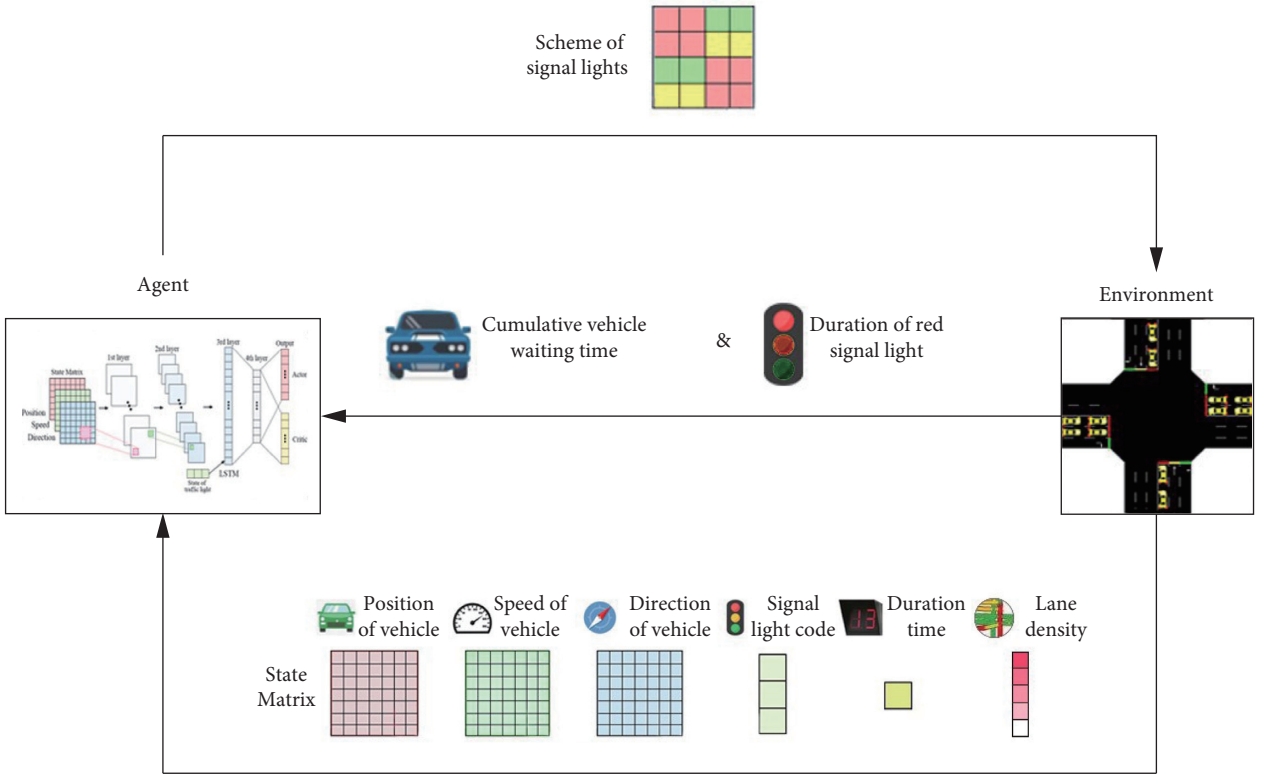


Figure 1: Intelligent traffic light system framework based on RL.

state space needs to reflect various pieces of information such as the traffic flow information of the intersection and road information, aiming at avoiding the output falling into the local optimum. Based upon past experience, the action space should be the sequence number of all signal phases, and the reward equation should be able to feedback a reasonable score to each action [25]. Besides defining the above basic elements of RL, a deep neural network is designed to define the coupling relationship between states and actions.

2.2. Reinforcement Learning Model

2.2.1. State Representation. If the intelligent traffic light control system can select a reasonable phase after perceiving the environment states, then the agent must be able to accurately perceive the environment. Therefore, the selected state variables must be able to describe the key characteristics of the intersection traffic flow in detail. These key features mainly include vehicle distribution state and current road information. With regard to the acquisition of vehicle distribution state, the traditional method uses sensor coils which can acquire vehicle position to build a spatial matrix. This method divides the lanes into a grid. When the vehicle is in the grid, the position is set to 1; otherwise, the position is set to 0. The grid information of all lanes is counted and summarized into a spatial matrix. However, the above method only considers the location distribution of the vehicle and does not consider the dynamic information of the vehicle. Therefore, it will affect the decision accuracy of the agent and reduce the training efficiency and convergence speed of the control model.

In view of the shortcomings of the above method, this study proposes using traffic cameras at an intersection for obtaining traffic flow information. Then, vehicles' positions, speed, and directions are obtained and then summarized into the virtualized road grid by analysing actual traffic flow images. Relative to the traditional method, the cameras used in space matrix method not only can obtain the location distribution and speed information of the vehicle, but can also obtain the drivers' driving intention by observing the vehicles' lane occupancy or turn signal conditions. The specific scheme is shown in Figure 2, where the vehicle matrix element information in Figure 2, respectively, represents direction, position, and speed. Specifically, the direction element represents the direction in which the vehicle passes through the intersection, and "1" indicates left turn, "2" indicates straight ahead, and "3" indicates right turn. The position element represents the order of the vehicle in all vehicles in current lane, and the integer number means the number of the vehicles.

Using the traffic cameras to extract the vehicle information, the occupancy rate ρ of each lane can be calculated. Specifically, we define the length of the vehicle as l_{car}, the total length of the current lane as l_{lane}, and the number of vehicles acquired by the camera as n; then the lane occupancy rate ρ can be defined as the following formula:

$$\rho = \frac{\sum_{i=1}^{n} l_{\text{car}i}}{l_{\text{lane}}}. \tag{1}$$

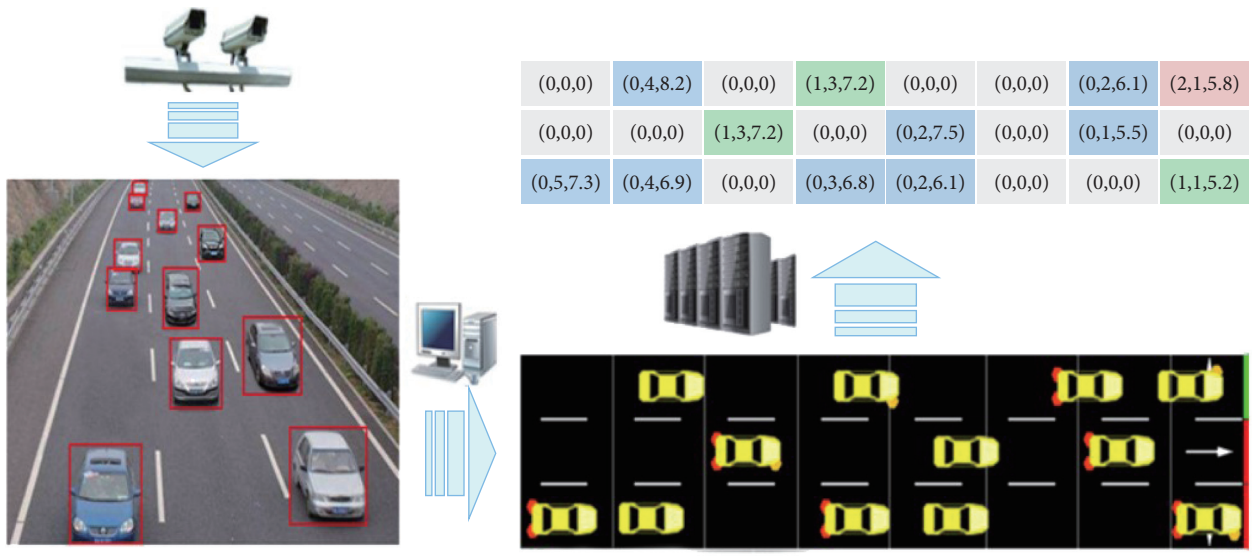


Figure 2: Vehicle information distribution matrix.

In addition to extracting traffic flow information at the intersection, status information of the traffic light should also be fully considered. For example, extending the green phase will improve traffic conditions when the traffic flow in a single direction is too large. However, this may cause the queue length of lanes in other directions to be too long, resulting in more serious traffic problems. Therefore, it is necessary to introduce the ratio of the current green time to the maximum green time as one of the observation information. The purpose of this motivation is to balance the traffic flow in all directions and avoid traffic jams caused by an excessively long green time in a single direction. We assume that the ratio is τ, and the value range of $\tau \in (0, 1)$.

In order to perceive the state information of the intersection in multiple dimensions, the current phase of the traffic light can be considered as one of the input states. Under normal circumstances, a standard intersection has twelve modes of vehicle movement, including going straight (east-west, west-east, south-north, and north-south), turning left (east-south, west-north, north-east, and south-west), and turning right (east-north, west-south, north-west, south-east). The setting of the signal phase at the intersection needs to be considered according to the size of the traffic flow in each direction. However, when the traffic flow in all directions of the intersection is very large, too many traffic flow conflicts will occur within the same phase. At this time, more phases must be set up in order to reasonably allocate the green time for the traffic flow in all directions to improve traffic safety and efficiency. The phase setting is mainly divided into the following types [26]:

(1) Two phases: When the traffic flow in each direction of the intersection is not distinguished by priority and there are few left-turning vehicles, we can only set the phase for the straight direction.

(2) Three phases: When a dedicated left-turn lane is set on a main road at the intersection and the traffic flow on other branch roads is low, a separate left-turn signal phase can be added to the main road.

(3) Four phases: When the traffic flow on the main road at the intersection is large and four separate left-turn lanes are set, the traffic lights can be set to a simple four-phase.

(4) Eight phases: This is used for an intersection that detects traffic flow in all directions and optimizes the conventional green phase. When the traffic flow in a certain direction of the intersection is too large, the traffic light can adjust the green phase according to the situation detected by the sensor.

When the current signal phase is acquired, the current phase can be encoded and entered in the form of one-hot encoding. Since the number of signal phases needs to be formulated according to the actual traffic flow at the intersection, the number of phases can be defined as n, and the coding method is shown in Table 1. At the same time, the code is defined as σ, which represents the current phase state.

Combining the requirements mentioned above, the environmental status information can be classified into two dimensions, including vehicle status which is defined as $\vec{s}_v$ < direction, position, speed > and the road status which is defined as $\vec{s}_r$ < the ratio between green time of the current phase and maximum green time, current signal phase, and lane occupancy >. Therefore, the input state of the environment can be written in vector form:

$$\vec{S} = \langle \vec{s}_v, \vec{s}_r \rangle = \langle \text{Direction, Speed, Position}, \tau, \sigma, \rho \rangle. \quad (2)$$

2.2.2. Action Representation. The traffic light needs to choose the appropriate phase output according to the traffic flow situation at the intersection, so as to relieve the traffic pressure and improve the traffic efficiency. Therefore, the flexibility of the action space will have a great impact on the decision-making effect of the traffic light. The design of the

Table 1: Green light phase coding table.

Traffic flow direction	One-hot coding	Action code
NS-straight	$[10\ldots00]_n$	a_0
NE-left; SW-left	$[01\ldots00]_n$	a_1
...	...	...
EW-straight	$[00\ldots10]_n$	a_{n-1}
ES-left; WN-left	$[00\ldots01]_n$	a_n

action space in this study mainly considers two factors: firstly, the agent can jump to any green phase based on traffic flow information. Secondly, the duration of the green phase can be dynamically adjusted according to the length of the queued vehicles. However, in order to avoid changing phases frequently which will lead to drivers' slow response or a long single green phase that may cause a long queue length in other directions, the duration φ of the green time needs to satisfy the condition of $\varphi \in (T_{\min\,\text{greentime}}, T_{\max\,\text{greentime}})$. In addition, since the right-hand driving policy is implemented in the area where this study is located, the right turn does not conflict with traffic flow in other directions, and in most cases vehicles can turn right at any time at an intersection. So, the right turn signal is set to an evergreen state. Regarding the traffic flow in other directions, the signal phase mode can be divided into n phases: north-south straight, north-south left, east-west straight, east-west left, etc. Therefore, the set of n phases can form the action space of this design, as shown in Table 1. At the same time, the action space can be expressed as a collection:

$$A = \{a_0, a_1, \ldots, a_{n-1}, a_n\}. \tag{3}$$

The traffic light can choose any action in the action space according to the traffic state of the intersection. For example, when there are many vehicles in the east-west straight direction, the action a_{n-1} will be performed. And the current phase can be coded as $A = [0, 0, \ldots, 1, 0]$. It is worth noting that if the next action is different from the current action, we need to insert a yellow phase before jumping to the next action for the sake of avoiding hidden traffic hazards. The yellow time is given by the following formula [26]:

$$T = t + \frac{s_{85}}{2a + (19.6 \times G)}, \tag{4}$$

where t is the driver's maneuver time, s_{85} is 85% of the speed limit of the intersection, a is the average deceleration, and G is the slope of the entrance lane of the intersection. We choose $T = 3$ s as the yellow time.

2.2.3. Reward Representation. In the process of RL, the reward value of each action can reflect the current state's preference for the action. From the perspective of the entire process, the reward value can provide direction for the agent's strategy update, and the lack of a fully considered reward equation often leads to the slow convergence of the control model. For the formulation of rewards, Liang et al. [21] proposed using the cumulative vehicle's waiting time difference at the intersection before and after the traffic light action as the reward equation. Liao et al. [27] put forward the corresponding penalty items when setting the reward equation in order to avoid the excessively long green time which will cause traffic loss in all directions at the intersection. Combining the above viewpoints, the definition of reward equation in this study will be measured from two dimensions.

Firstly, we consider the change in the cumulative waiting time of vehicles between consecutive actions at the intersection. For example, when the traffic light outputs an action a_t, it will get a reward r_{t1}. The reward obtained in this process can be defined as

$$r_{t1} = W_t - W_{t+1}. \tag{5}$$

Among them, W_t and W_{t+1}, respectively, represent the accumulated waiting time of all vehicles at the intersection before and after the action a_t. The meaning of W_t is presented in the following formula:

$$W_t = \sum_{\varepsilon=0}^{N} w_{s,e}. \tag{6}$$

In the formula, ε is the number of vehicles queuing at the intersection, N is the total number of vehicles queuing, and $w_{s,e}$ is the vehicle delay, which means cumulative total waiting time of the vehicle from the stop moment to the departure moment. Combining formulas (5) and (6), it can be concluded that the greater the change in the accumulated waiting time before and after the action is performed, the greater the reward value.

Secondly, in order to balance the traffic flow in all directions at the intersection and achieve the goal of safe driving, when defining the reward equation, a penalty term is formulated in order to avoid long green time. The penalty term is shown in the following formula :

$$r_{t2} = -\alpha \cdot \max\left\{\left(T_t - T_{\max\,\text{greentime}}\right), 0\right\}. \tag{7}$$

In the formula, T_t represents the duration of the corresponding green time at step t. The predefined maximum green time is $T_{\max\,\text{greentime}}$, and α is the coefficient, which is used to control the weight of punish term in reward function. When multiple green phases occur in succession and exceed the set value, a penalty will be given to avoid traffic flow unbalance in all directions at the intersection.

Based on the formulas, the final reward equation is shown in the following equation :

$$R_t = r_{t1} + r_{t2} = \left(W_t - W_{t+1}\right) - \max\left\{\left(T_t - \alpha T_{\max\,\text{greentime}}\right), 0\right\}. \tag{8}$$

2.3. Agent Deep Decision Network. The traffic light decision network model is shown in Figure 3. The input state of the system designed in this study is a matrix containing vehicle direction, position, and speed information. Since the detection length of a single lane is divided into 8 grids, and there are 8 detection lanes at the intersection, thus the input data dimension is $8 \times 8 \times 3$. According to the characteristics of the input data, the convolutional layer and the fully

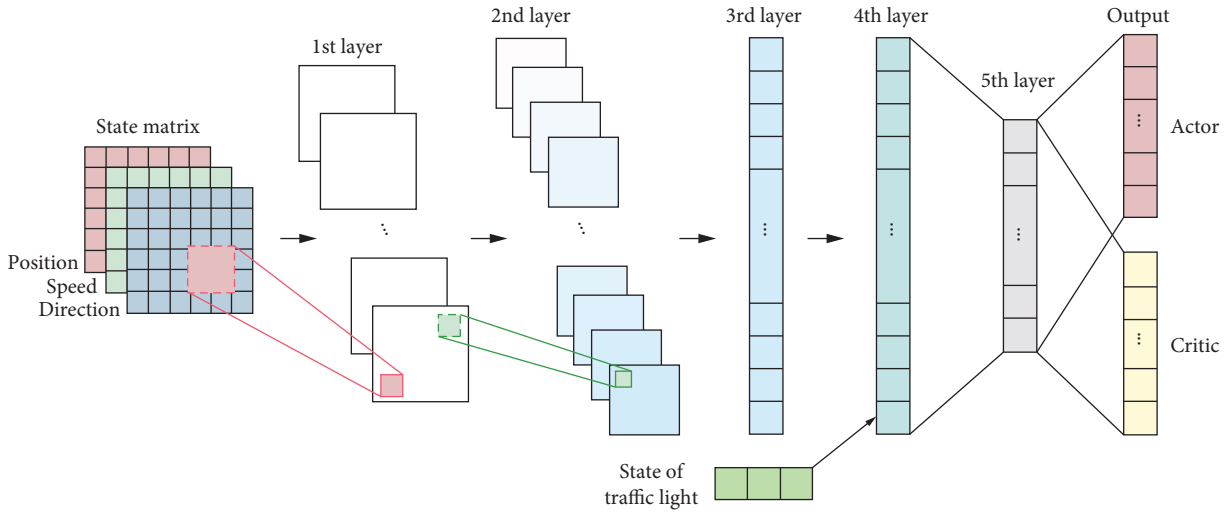


Figure 3: PPO decision network model.

connected layer of the decision network model are constructed as follows. The first convolutional layer contains 4 filters. The size of each filter is 1×1, and each movement step is 1×1. The second convolutional layer has 8 filters. The size of each filter is 1×1, and each movement step is 1×1. The pooling of both convolutional layers adopts the maximum pooling method, the size of the convolution kernel is 2×2, and the moving step size is 1×1. The third layer is a fully connected layer which converts the output of the convolutional layer into a vector form, the fourth and fifth layers are, respectively, 64 and 32 fully connected layers, and the activation functions in the network all use ReLU. The Actor of the output layer is composed of two fully connected layers, which outputs μ and σ, respectively. Critic is composed of a fully connected layer and outputs the value v which is one of the important parameters for updating the network decision.

The PPO algorithm is improved based on Trust region policy optimization (TRPO) [28]. At the same time, using the importance sampling for advantage estimation solves the problem of the sampling method of having high variance and low data efficiency [29]. The loss function of the PPO algorithm is presented as follows:

$$L_{\text{actor\_}t}(\theta) = E_t\left[\min\left(\frac{\pi_{\theta\_\text{new}}(a_t|s_t)}{\pi_{\theta\_\text{old}}(a_t|s_t)}A_t, \text{clip}\left(\frac{\pi_{\theta\_\text{new}}(a_t|s_t)}{\pi_{\theta\_\text{old}}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon\right)A_t\right)\right], \tag{9}$$

$$L_{\text{critic\_}t}(\theta) = E_t\left[(A_t)^2\right]. \tag{10}$$

In (9) and (10), A_t is the advantage function which represents the advantage of performing the current action over other actions in a certain state. π_{θ_new} represents the strategy of the Actor-new network, θ_new represents the strategy parameter which will change during every update, and π_{θ_old} represents the strategy of the Actor-old network. The network parameters are only updated periodically. The off-policy method which uses the Actor-new network to interact with the environment obtains the experience parameter θ_new and then uses the weight of the Actor-new network to update the Actor-old network. In order to prevent the probability distribution of the output of the two Actor networks from being too different to avoid a sudden change in strategy, the clip method is used to tailor the distribution difference between π_{θ_old} and π_{θ_new}, with ε being the coefficient of clip which is generally 0.2.

$$A_t = \delta_t + \gamma\delta_{t+1} + \cdots + \gamma^{T-t+1}\delta_{T-1}, \tag{11}$$

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t). \tag{12}$$

The meaning of δ_t in (11) is shown in (12), where $V(s_t)$ is the value description of state s_t which is output by the Critic network, γ is the discount coefficient, and r_t is the reward value obtained by taking actions in state s_t.

The training process of PPO model is shown in Figure 4. At each time step t, the acquired observation information s_t is input into the network by the agent, and action a_t is output according to μ and σ of the Actor-new network. At the same

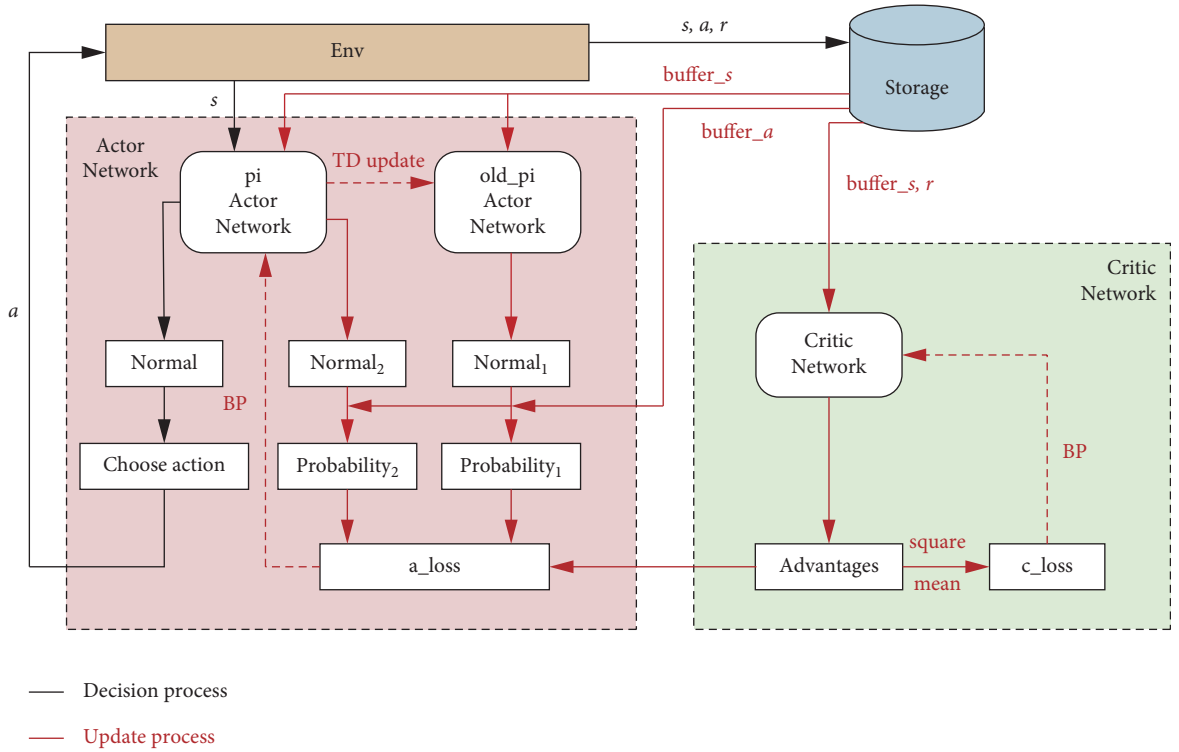


Figure 4: PPO algorithm decision network update process.

time, the agent obtains the new environment state s_{t+1}, after several iterations until a certain amount of state s, action a, and reward value r are stored; the last state s of the above stage is input into the Critic network to obtain the value of the state and calculate the discount reward for each action. Then, the combination of all states s is input into the Critic network to obtain the value of all states, and the loss function $L_{\text{critic}\_t}(\theta)$ is constructed by the discount reward and value judgment. The parameters of the Critic network are updated by using backpropagation.

For the update of the Actor network, it is necessary to input all the stored states s into the Actor-old and Actor-new networks (the structure of the two networks is identical) to obtain the normal distributions Normal1 and Normal2, respectively. At this time, all stored actions are input into the normal distributions Normal1 and Normal2 and the prob1 and prob2 corresponding to each action are obtained; then the importance weight is calculated by dividing prob2 by prob1. Finally, the loss function $L_{\text{actor}\_t}(\theta)$ is formed for performing backpropagation to update the Actor-new network parameters. After the above steps loop a certain number of times, the Actor-new network weight is then used to update the Actor-old network.

3. Experiment

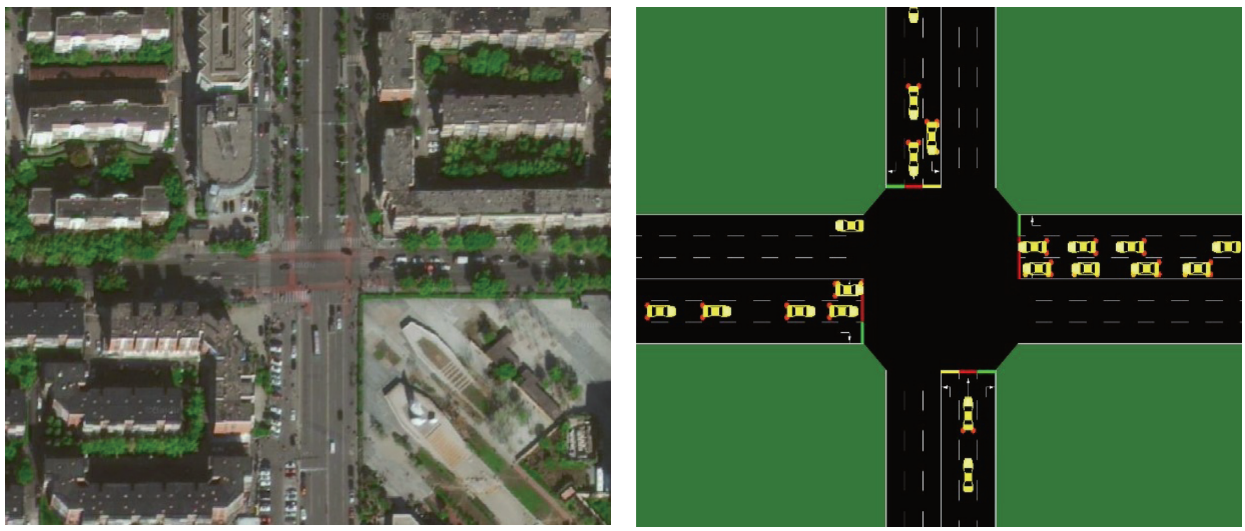
3.1. Simulation Environment

3.1.1. Intersection Environment. In order to verify the effectiveness and accuracy of the above scheme, the generation of virtual traffic scenes will be realized by using the urban traffic simulation software SUMO which is an open source, micro, multimodal traffic simulation software. Compared with other simulation software programs such as Aimsun and Vissim, SUMO executes faster. Not only can it perform large-scale traffic flow management, but it also can be linked with other applications such as PyCharm. Most importantly, SUMO's own API interface Traci (traffic control interface) can extract simulation environment data online and can use agent commands for real-time simulation in order to realize the interactive process of RL.

This study uses the intersection of Hongyan East Road and Xinwang South Road in Chaoyang District, Beijing, as the traffic simulation scene, as shown in Figure 5(a). This intersection has 3 lanes in each direction: left turn lane, through lane and right turn lane. The intersection area includes the road within 150 meters in each direction, and the maximum allowable speed in all lanes is 14 m/s (50.4 km/h). We use SUMO to virtualize the real intersection scene, as shown in Figure 5(b).

3.1.2. Traffic Flow Generation. In order to simulate the real traffic situation as much as possible, we use the traffic flow data from the intersection of Hongyan East Road and Xinwang South Road within one day (from 4 : 00 to 24 : 00) in the experiment, as shown in Figure 6 [30, 31]. At the same time, through consulting the data, it is concluded that the traffic light at the intersection adopts the four-phase timing control scheme and the time length and phase sequence of each phase are shown in Table 2.

Three traffic modes are developed for SUMO experiment by classifying the real traffic flow data. The traffic flow of each mode is described as follows:



(a) (b)

Figure 5: Schematic diagram of intersection model. (a) Intersection scene in reality. (b) Experimental intersection scene.

(1) Heavy traffic mode P1: In this mode, the traffic flow in all directions of the intersection is in the peak period, and the traffic flow in the straight direction is greater than that in the left turn direction. We choose the traffic flow data at 14 : 00 to configure traffic flow of mode P1.

(2) Primary and minor traffic mode P2: In this mode, the north-south direction is the main road, and the east-west direction is the secondary road. At the same time, the traffic demand in the north-south direction is greater than that in the east-west direction. We choose the traffic flow data at 19 : 00 to configure traffic flow of mode P2.

(3) Tidal traffic mode P3: In this mode, the traffic demand from south and east is higher than their respective opposite directions. We choose the traffic flow data at 9 : 00 to configure traffic flow of mode P3.

3.1.3. Hyperparameters of Agent Decision Model. The hardware platform of this experiment is provided by a notebook equipped with Intel core i7-6700k CPU, Samsung 16 GB RAM, and Nvidia GeForce GTX970 GPU. The software platform uses an open source Linux system and installs common modules such as SUMO, Gym, TensorFlow, and DRL algorithm libraries. The settings of the simulation environment are shown in Table 3.

3.2. Simulation Results and Discussion. The experiment will be divided into two parts. In the first part, we apply the signal timing control scheme (FST) to three traffic modes: P1, P2, and P3 and count the waiting time of all vehicles in each mode. The timing control scheme is set as Table 2. The simulation time is set as 20000 seconds, which is used to observe the change of waiting time of all vehicles at the intersection in a long period.

As shown in Figures 7–9, when the timing control scheme is adopted, the waiting time of all vehicles fluctuates within a certain range regardless of whether the current traffic flow mode of the intersection is P1, P2, or P3. This phenomenon will bring a lot of urban problems.

Therefore, in the second part of the experiment, we try to use DRL scheme to effectively alleviate the above problem. The training process contains 200 episodes, and each episode has 2500 steps. At the beginning of each episode, SUMO randomly generates vehicles according to the configured traffic flow parameters in each direction, and the traffic flow parameters is given according to the actual traffic data. Then, PPO algorithm is used to optimize the policy of the agent in the rest steps of episode. This part of the experiment evaluates the performance of traffic lights from two aspects: (1) convergence of performance indicators under P1–P3 traffic flow modes when DRL scheme is adopted, i.e., changes of overall waiting time and average queue length of vehicles at intersection; (2) performance indicators of DRL scheme and timing control scheme under P1–P3 traffic flow modes are compared.

As shown in Figures 10–12, the convergence of performance indicators of DRL scheme under three traffic modes is shown. It can be seen from the figure that the actions generated by the agent at the beginning of the exploration environment may be unreasonable, so the intersection will be congested. At this time, the increase of vehicle queue length leads to the long waiting time of all vehicles. However, as the agent is constantly updating the decision parameters, the queue length and waiting time can rapidly decrease and remain stable. It is not difficult to see that the DRL scheme is more satisfactory than the timing control scheme.

Figures 13–15 present the performance indicators comparison between DRL scheme and timing control scheme under P1–P3 traffic modes. We repeat the experiment of three traffic modes corresponding to the two schemes for 200 times and draw the box diagram through the statistical performance indicators. In these figures, the

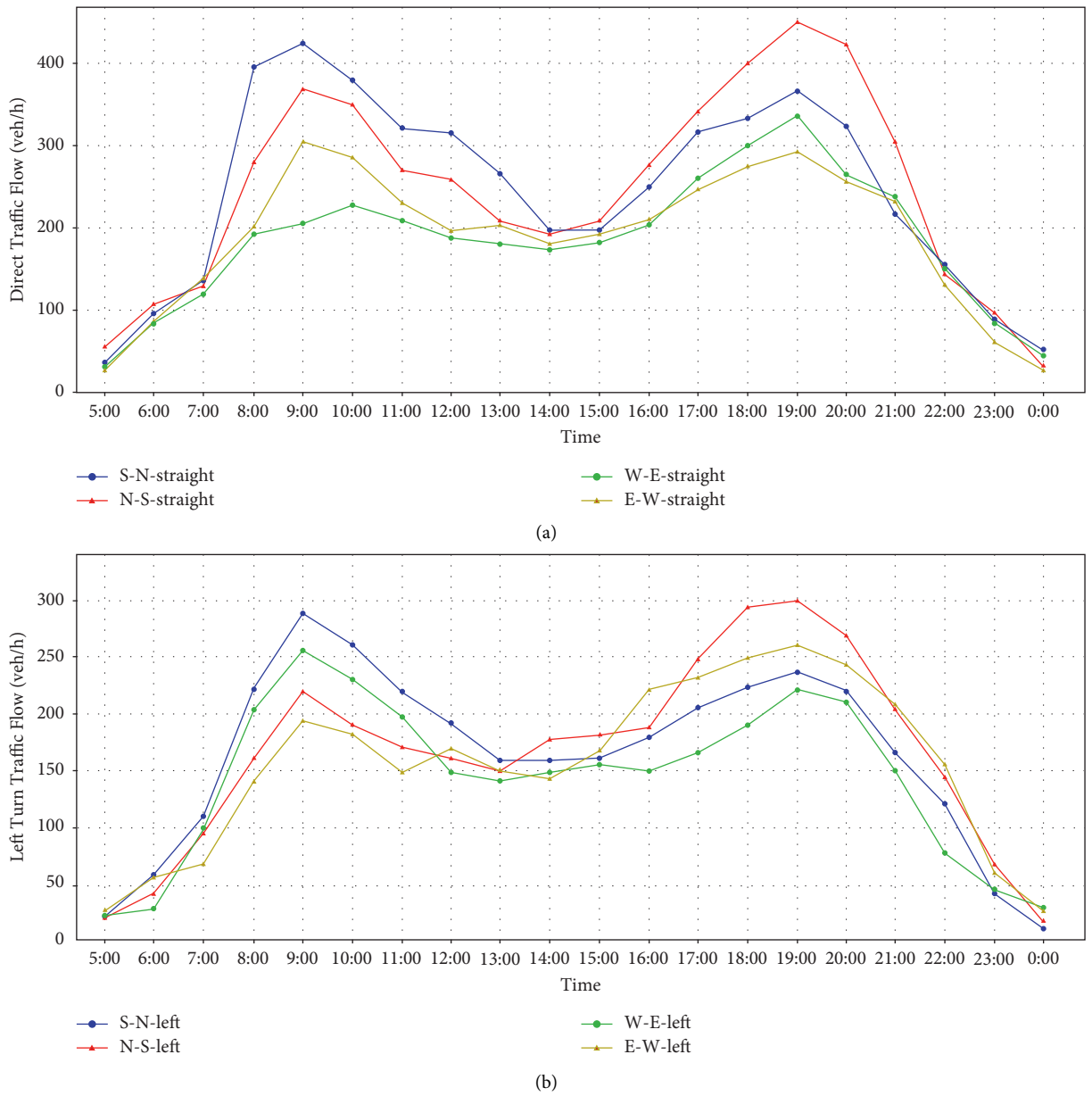


Figure 6: The traffic demands in real intersection. (a) Traffic flow in all straight directions. (b) Traffic flow in all left turn directions.

Table 2: Traffic light four-phase control scheme.

Phase sequence	Traffic flow direction	Duration (s)
0	NS-straight	32
1	t_{switch}	3
2	NE-left; SW-left	16
3	t_{switch}	3
4	EW-straight	32
5	t_{switch}	3
6	ES-left; WN-left	16
7	t_{switch}	3

dotted line in the middle of the box is the median, the lower line is the lower quartile (the first quartile), and the upper line is the upper quartile (the third quartile). Obviously, compared with the timing control, the average queue length and the total waiting time of DRL scheme in each traffic mode present better results.

Specifically, when the traffic mode is P2, the total waiting time and average queue length of vehicles using the

Table 3: Simulation environment hyperparameters.

Parameter	Meaning	Value
γ	Discount factor	0.99
l	Learning rate	0.001
ε	Clip range	0.2
T	Every episode simulation time	5000 s
$n_$steps	The number of steps for update	128
ent_coef	Entropy coefficient for the loss calculator	0.01
$vf_$coef	Value function coefficient for the loss function	0.5

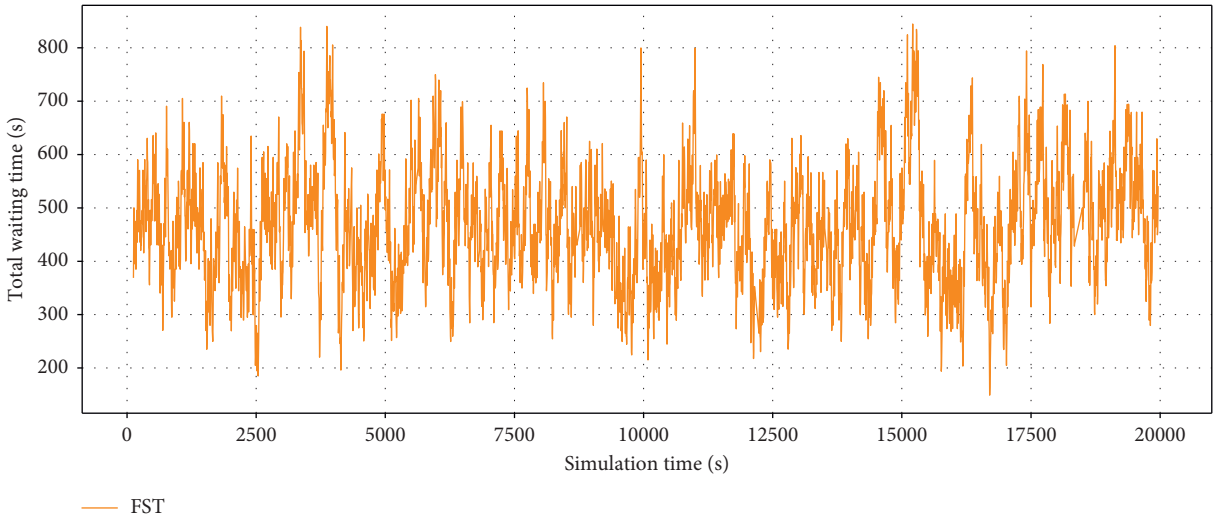


Figure 7: Performance indicator of FST control traffic lights in P1 mode.

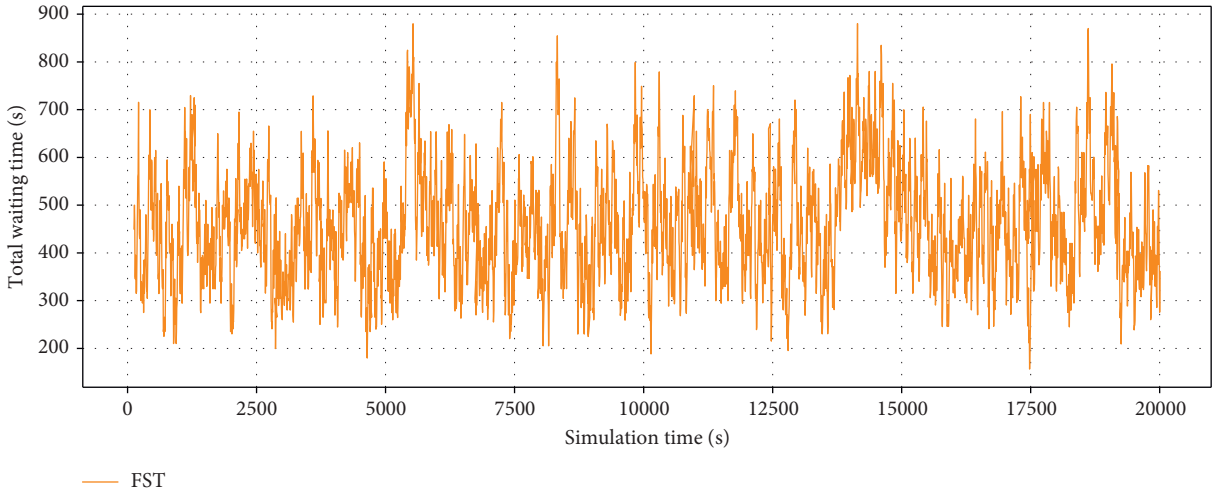


Figure 8: Performance indicator of FST control traffic lights in P2 mode.

DRL scheme are reduced the most compared with the timing control scheme, which is about 80.4% and 50%, respectively. When the traffic mode is P3, the total waiting time and average queue length of the vehicles in the DRL scheme are the least reduced compared with the timing control scheme, which is about 76.3% and 33.4%, respectively. Combining the analysis of the above results, it can be seen that the performance of the DRL scheme is significantly improved under the three traffic modes; that is, the total waiting time of vehicles is reduced from 76.3% to 80.4%, and the average queue length is reduced from 33.4% to 50%.

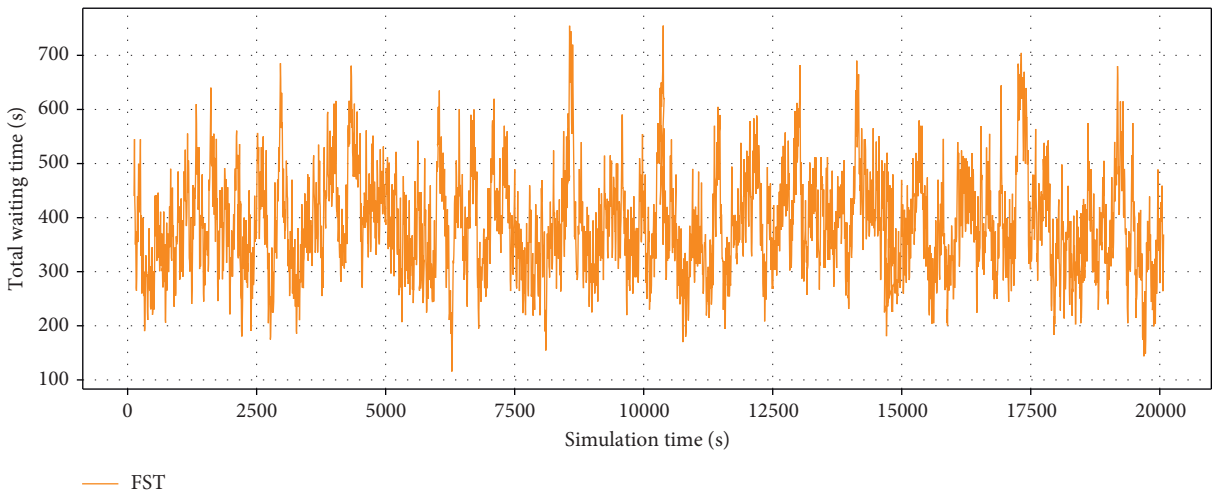


Figure 9: Performance indicator of FST control traffic lights in P3 mode.

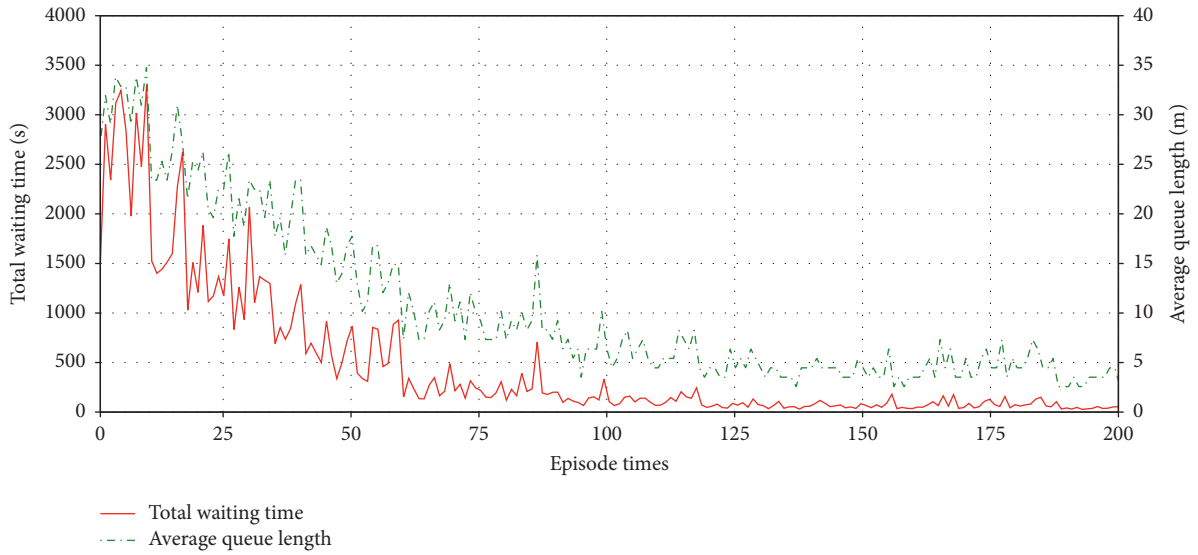


Figure 10: Performance indicators of DRL control traffic lights in P1 mode.

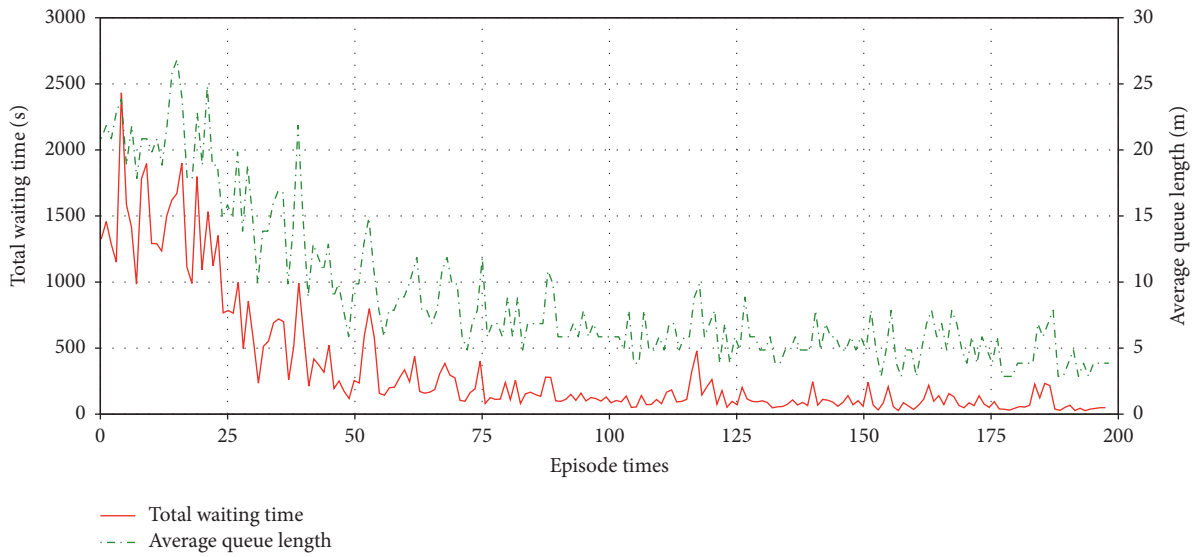


Figure 11: Performance indicators of DRL control traffic lights in P2 mode.

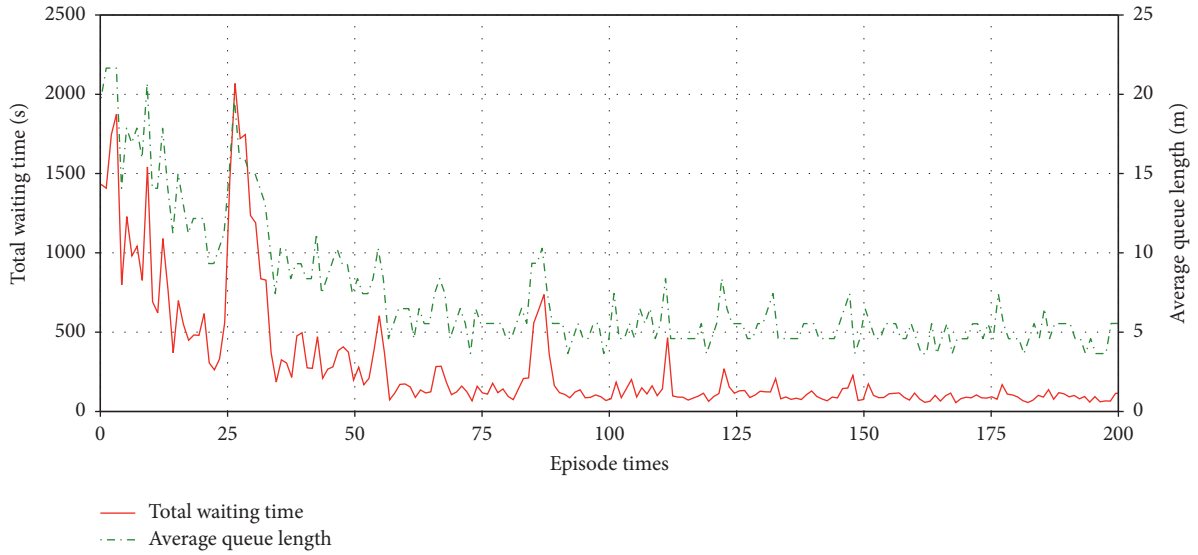


Figure 12: Performance indicators of DRL control traffic lights in P3 mode.

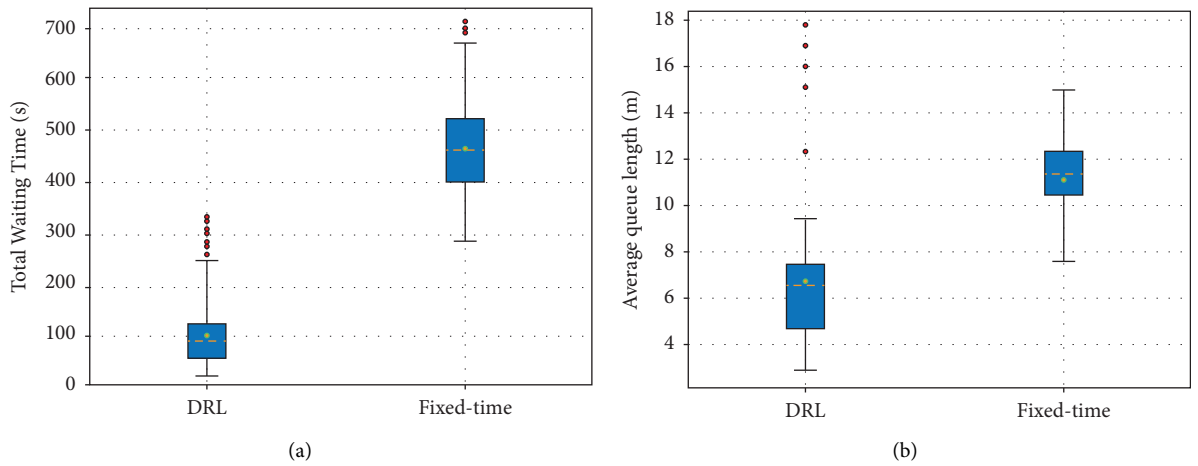


Figure 13: Distribution characteristics of performance indicators in P1 mode. (a) Total waiting time. (b) Average queue length.

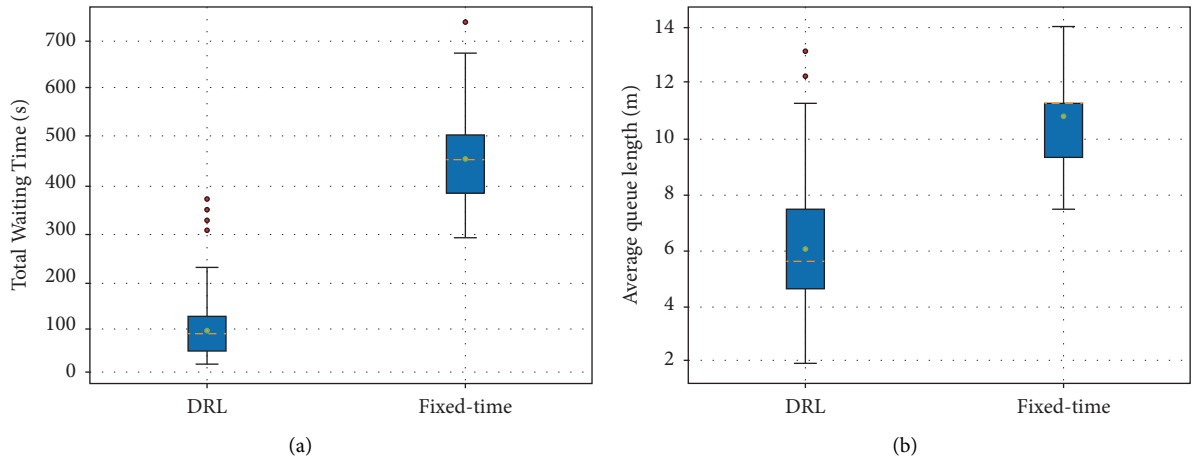


Figure 14: Distribution characteristics of performance indicators in P2 mode. (a) Total waiting time. (b) Average queue length.

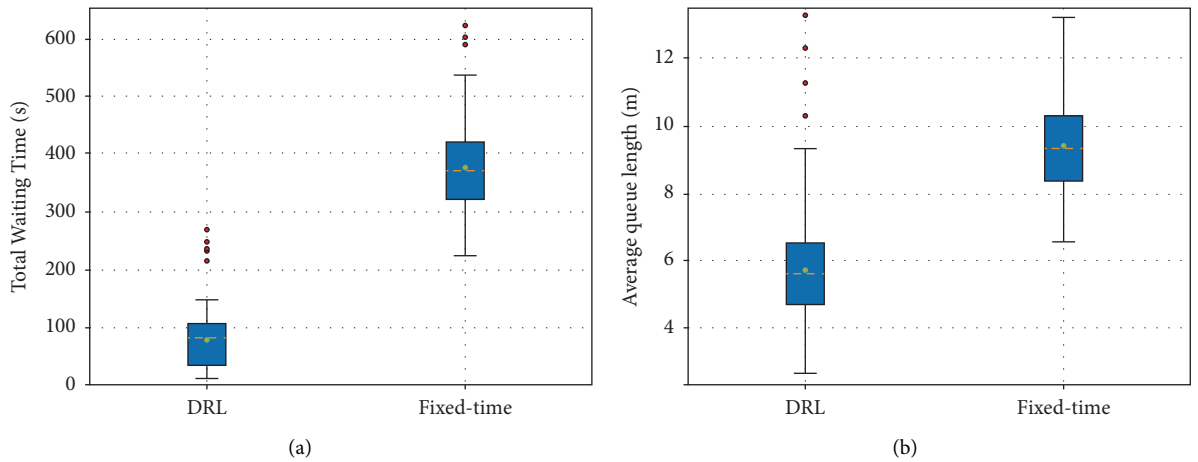


Figure 15: Distribution characteristics of performance indicators in P3 mode. (a) Total waiting time. (b) Average queue length.

4. Conclusion

This study proposes a scheme of using DRL technology to control the traffic light at an intersection. By using traffic cameras to collect vehicle distribution information at the intersection, a state space matrix with vehicle position, speed, and direction information is established as the input of the decision model after the information is obtained by image processing technology. In order to reduce the intense driving behaviour caused by the long queue of vehicles in one direction at the intersection, the reward value not only considers the cumulative waiting time difference between two actions, but also considers the impact of the green phase duration. To avoid danger, penalties are given when an excessively long green time emerges. The PPO algorithm based on the strategy gradient which has a better effect than the method based on value function will improve the strategy according to the approximate value of the strategy gradient every iteration. The experimental results show that, under different traffic flow modes, RL method is superior to the traditional timing control in terms of reducing vehicle waiting time and queue length.

It should be pointed out that the traffic light scheme designed for this study only uses the classic four-phase scheme and does not design multiple phase schemes for tidal traffic flow. Besides, the experimental scene is too single. In reality, intersections are not simply crossroads, but a mixed road network coupled with street roads and expressways. In addition, the traffic flow on the road network is a mixed traffic flow consisting of motor vehicles, pedestrians, and nonmotor vehicles. Therefore, in order to get closer to the real traffic scene, further studies can consider designing a mixed road network structure and traffic flow.

Data Availability

The traffic flow data and phase information of the traffic light used in this study are all from the data of Beijing Chaoyang District Traffic Police Detachment.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (NSFC) (Grant no. 61873007) and the Beijing Natural Science Foundation (Grant no. 1182001).

References

[1] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intelligent Transport Systems*, vol. 11, no. 7, pp. 417–423, 2017.

[2] X. Liang, T. Yan, J. Lee, and G. Wang, "A distributed intersection management protocol for safety, efficiency, and driver's comfort," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1924–1935, 2018.

[3] N. casas, "Deep deterministic policy gradient for urban traffic light control," vol. 2017, 2017, https://arxiv.org/abs/1703.09035.

[4] S. Göttlich, M. Herty, and U. Ziegler, "Modeling and optimizing traffic light settings in road networks," *Computers & Operations Research*, vol. 55, pp. 36–51, 2015.

[5] R. Hoyer and U. Jumar, "Fuzzy control of traffic lights," in *Proceedings of the 1994 IEEE 3rd International Fuzzy Systems Conference*, vol. 3, pp. 1526–1531, Orlando, FL, USA, June 1994.

[6] C. Yu, W. Sun, H. X. Liu, and X. Yang, "Managing connected and automated vehicles at isolated intersections: from reservation- to optimization-based methods," *Transportation Research Part B: Methodological*, vol. 122, pp. 416–435, 2019.

[7] J. Zhao, V. L. Knoop, and M. Wang, "Two-dimensional vehicular movement modelling at intersections based on optimal control," *Transportation Research Part B: Methodological*, vol. 138, pp. 1–22, 2020.

[8] Y. Bichiou and H. A. Rakha, "Developing an optimal intersection control system for automated connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, pp. 1908–1916, 2018.

[9] J. Pang, "Review of microcontroller based intelligent traffic light control," in *Proceedings of the 2015 12th International Conference & Expo on Emerging Technologies for a Smarter World (CEWIT)*, pp. 1–5, Melville, NY, USA, October 2015.

[10] F. Gao, X. S. Hu, S. E. Li, K. Li, and Q. Sun, "Distributed adaptive sliding mode control of vehicular platoon with uncertain interaction topology," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 99, p. 1, 2018.

[11] L. A. Prashanth and S. Bhatnagar, "Threshold tuning using stochastic optimization for graded signal control," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 3865–3880, 2012.

[12] W. Liu, G. Qin, Y. He, and F. Jiang, "Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X network's dynamic clustering," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8667–8681, 2017.

[13] L. Li, Y. Lv, and F. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[14] V. R. Konda and J. N. Tsitsiklis, "OnActor-critic algorithms," *SIAM Journal on Control and Optimization*, vol. 42, no. 4, 2000.

[15] M. Xu, J. P. Wu, L. Huang, R. Zhou, T. Wang, and D. Hu, "Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning," *Intelligent Transportation Systems*, vol. 24, no. 3, pp. 1–10, 2019.

[16] G. Wade and R. Saiedeh, "Asynchronous n-step Q-learning adaptive traffic signal control," *Intelligent Transportation Systems*, vol. 23, no. 7, pp. 319–331, 2019.

[17] S. M. A. Shabestary and B. Abdulhai, "Deep learning vs. Discrete reinforcement learning for adaptive traffic signal control," in *Proceedings of the International Conference on Intelligent Transportation Systems (ITSC)*, pp. 286–293, Maui, HI, USA, November 2018.

[18] J. Zeng, J. Hu, and Y. Zhang, "Adaptive traffic signal control with deep recurrent Q-learning," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1215–1220, Changshu, China, June 2018.

[19] L. N. Alegre, A. L. C. Bazzan, and B. C. D. Silva, *Quantifying the Impact of Non-stationarity in Reinforcement Learning-Based Traffic Signal Control*, PeerJ, Boston, Boston, MA, USA, 2020.

[20] B. P. Abbott, R. Abbott, T. D. Abbott et al., "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, no. 6, 2016.

[21] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

[22] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-Simulation of Urban MObility," *International Journal of Agile Systems and Management*, vol. 5, no. 3, pp. 128–138, 2012.

[23] S. E. Li, S. Xu, X. Huang, B. Cheng, and H. Peng, "Eco-departure of connected vehicles with V2X communication at signalized intersections," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 7, pp. 5439–5449, 2015.

[24] L. Z. Shu, J. Wu, and C. Wang, "Urban traffic signal control based on deep reinforement learning," *Computer Applications*, vol. 39, no. 5, pp. 255–259, 2019.

[25] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Design of reinforcement learning parameters for seamless application of adaptive traffic signal control," *Journal of Intelligent Transportation Systems*, vol. 18, no. 3, pp. 227–245, 2014.

[26] R. Li and L. Zhang, *Urban Traffic Signal Control*, Tsinghua University Press, Beijing, China.

[27] L. Liao, J. Liu, X. Wu et al., "Time Difference Penalized Traffic Signal Timing by LSTM Q-Network to Balance Safety and Capacity at Intersections," *IEEE Access*, vol. 8, no. 99, p. 1, 2017.

[28] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *Society for Industrial and Applied Mathematics*, vol. 42, no. 7, pp. 1143–1161, 2003.

[29] A. Guo, L. Song, and X. Chen, "Learning similar tasks based on PPO by transferring trajectory," in *Proceedings of the IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 126–131, Banff, Canada, May 2019.

[30] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: a Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019.

[31] Y.-T. Wu and C.-H. Ho, "The development of Taiwan arterial traffic-adaptive signal control system and its field test: a Taiwan experience," *Journal of Advanced Transportation*, vol. 43, no. 4, pp. 455–480, 2009.