IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2019, accepted March 22, 2019, date of publication March 26, 2019, date of current version April 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907618

Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control

HONGWEI GE[1,2], YUMEI SONG[1], CHUNGUO WU[3], JIANKANG REN[1], AND GUOZHEN TAN[1]

[1]College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[2]Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA
[3]Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun 130012, China

Corresponding author: Hongwei Ge (hwge@dlut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572104 and Grant 61103146, in part by the National Key R&D Program of China under Grant 2018YFB1600600, in part by the project of the Key Laboratory of Symbolic Computation and Knowledge Engineering in Jilin University under Grant 93K172017K03, and in part by the Fundamental Research Funds for Central Universities under Grant DUT17JC04.

ABSTRACT The problem of adaptive traffic signal control in the multi-intersection system has attracted the attention of researchers. Among the existing methods, reinforcement learning has shown to be effective. However, the complex intersection features, heterogeneous intersection structures, and dynamic coordination for multiple intersections pose challenges for reinforcement learning-based algorithms. This paper proposes a cooperative deep Q-network with Q-value transfer (QT-CDQN) for adaptive multi-intersection signal control. In QT-CDQN, a multi-intersection traffic network in a region is modeled as a multi-agent reinforcement learning system. Each agent searches the optimal strategy to control an intersection by a deep Q-network that takes the discrete state encoding of traffic information as the network inputs. To work cooperatively, the agent considers the influence of the latest actions of its adjacencies in the process of policy learning. Especially, the optimal Q-values of the neighbor agents at the latest time step are transferred to the loss function of the Q-network. Moreover, the strategy of the target network and the mechanism of experience replay are used to improve the stability of the algorithm. The advantages of QT-CDQN lie not only in the effectiveness and scalability for the multi-intersection system but also in the versatility to deal with the heterogeneous intersection structures. The experimental studies under different road structures show that the QT-CDQN is competitive in terms of average queue length, average speed, and average waiting time when compared with the state-of-the-art algorithms. Furthermore, the experiments of recurring congestion and occasional congestion validate the adaptability of the QT-CDQN to dynamic traffic environments.

INDEX TERMS Deep reinforcement learning, multi-intersection signal control, Q-learning, Q-value transfer, cooperative.

I. INTRODUCTION

Traffic congestion has become a major strategic problem facing the sustainable and harmonious development of cities. Dues to the limitation of urban space, to relieve traffic congestion by road expansion has become difficult. Traffic signal control is one of the most effective ways to improve the capacity of road intersections. The adaptive control of signal lights can optimize the traffic of regional road network, reduce congestion and carbon dioxide emissions [1]. The adaptive control strategy regards the transportation system as an uncertain system that realizes dynamic optimization and adjustment of signal timing to address the stochastic characteristics of traffic network through the feedback of measured state variables, such as traffic flow, delay time and queue length [2].

In recent years, a variety of machine learning methods are used for the control of urban traffic signal, such as fuzzy logic [3]–[6], neural networks [7]–[9], evolutionary algorithms [10]–[13] and dynamic programming [14]. Yang *et al.* [5] developed two adaptive two-stage fuzzy controllers for traffic signals under different traffic density at an isolated intersection. Bi *et al.* [3] proposed a type-2 fuzzy logic controller for coordinated arterial traffic signal control with the objective

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

of minimizing vehicular average delay. Fuzzy logic signal controllers generally establish a set of rules based on expert knowledge, from which appropriate actions of the traffic signal are selected based on inputs. However, the establishment of rules excessively depends on expert knowledge. Moreover, it is difficult to generate a set of effective rules when the number of phases increases at multiple intersections. Shen and Kong [8] investigated the combination of artificial neural network and fuzzy theory for a road network traffic intelligent coordination control with bus priority, although the performance of neural network and fuzzy logic is sensitive to the initialization and training process. Genetic algorithm was adopted in [10] for the optimization of traffic flow within an urban traffic light intersection. However, such an algorithm is not suitable for online problems such as intersection signal control, since it requires expensive computation cost to converge to the optimal solution. Besides, dynamic programming is widely used in traffic signal control. In [14], an action-dependent heuristic dynamic programming was proposed for traffic signal control at two intersections. With the expansion of the problem scale, dynamic programming requires effective mechanisms to address excessive computation cost and the intractability of computing the transition probability for the operating environment.

Many studies use the framework of reinforcement learning (RL) to find the optimal control strategies [15], [16]. Reinforcement learning learns the optimal strategies by perceiving the state of environment and receiving uncertain information from environment. The goal is to find the optimal strategy to maximize the discounted cumulative reward via continuous interaction with its environment. Traffic signal control is actually a sequential decision-making problem. Extensive research has been conducted using reinforcement learning for isolate intersection and multi-intersection traffic signal control. For multiple intersections, the existing methods can be classified into two categories, i.e., the centralized control methods and the decentralized control methods. The algorithms employing the centralized control train a global agent to control the traffic signal of the entire road network [17], [18]. However, they are not free from the ''curse of dimensionality'', since the dimensionality of the state and action space will grow exponentially as the number of intersections increases. The algorithms employing the decentralized control formulate the multi-intersection signal light control as a multi-agent system, in which each agent controls a single intersection and only observes and percieves parts of the traffic environment [19]–[26].

Traditional reinforcement learning method builds up state space by human-crafted intersection features. To avoid the excessive volume of state space, it usually simplifies state representation. However, this strategy will result in the loss of some important information. For example, the strategy of expressing the state space by vehicle queue length [18], [19] ignores the information of the moving vehicles and the position and speed of the vehicles. The average vehicle delay [17] strategy only reflects the history traffic data and cannot satisfy the real-time traffic demand. These strategies are based on partial information of the intersections, so they will not always guarantee to generate optimal decisions.

Recently, deep reinforcement learning has attracted much attention due to the effectiveness of deep Q-network (DQN) [27], [28]. Further, many researchers have introduced deep reinforcement learning to adaptive traffic signal control, including single intersection signal control [29]–[32] and multi-intersection signal control [33], [34]. Deep reinforcement learning utilizes the automatic feature extraction ability of deep models to extract intersection state information from raw real-time traffic data. The models of convolutional neural network (CNN) [35] and deep stacked auto-encoder (SAE) [36] enable agents to make full use of intersection state information for optimal decision-making. However, there are less research on deep reinforcement learning for cooperative traffic light control at multiple intersections. Thus the effectiveness of such methodologies for cooperative multi-intersection control remains to be studied.

In this paper, based on multi-agent system, we propose a cooperative deep Q-network with Q-value transfer (QT-CDQN) for adaptive multi-intersection signal control. The main contributions are twofold: 1) To balance the traffic flow at each intersection from the perspective of regional control, the influence of the neighboring intersections is taken into account by integrating Q-value transfer strategy into the cooperative Q-network; 2) To extract the intersection state information effectively, a CNN estimation network is established to automatically extract the features from the original traffic state and approximate the optimal Q-values. Importantly, the proposed QT-CDQN can be extended to different number of intersections without the curse of dimensionality for the state-action space, and there is no restriction on the structure of each intersection.

The remainder of this paper is organized as follows. Section II gives the reviews of reinforcement learning and deep reinforcement learning for traffic signal control. Section III introduces reinforcement learning and the DQN algorithm. Section IV gives the details of the proposed cooperative deep Q-network with Q-value transfer focusing on the multiple intersections. Section V presents the experimental studies on different conditions and compares the performance with the state-of-the-art algorithms. Section VI concludes the paper.

II. RELATED WORK

Algorithms utilizing reinforcement learning and deep reinforcement learning have many attractive properties. On one hand, reinforcement learning is a goal-oriented learning method from the environment, which focuses on interaction with the environment. On the other hand, deep learning possesses strong hierarchical feature extraction ability and nonlinear approximation ability. In this section, we first review the traffic signal control algorithms based on reinforcement learning and deep reinforcement learning. Then we analyze

the existing challenges and discuss the motivations of this paper.

Most traffic signal control studies based on reinforcement learning focus on the traffic scenes at a single intersection [37]–[41]. The intersection state space will grow exponentially as the number of intersections increases, and it is not feasible to express the values of all actions for each possible state. Thus, traditional tabular-based reinforcement learning algorithms are difficult to be extended to multiple intersections. To address this problem, the algorithms employing multi-agent reinforcement learning are proposed for adaptive traffic signal control in regional traffic scenario [42]. The control strategies of such algorithms can be classified into the independent mode and the integrated mode.

In the independent mode, each intersection has an RL agent working independently of other agents. Abdoos *et al.* [19] modeled a relatively large traffic network as a multi-agent system. Each agent is responsible for controlling the traffic signal at one intersection. It uses only local information of the intersection, i.e., average queue length, to estimate the states. This method based on independent mode does not consider the influence of neighboring intersections. In the integrated mode, the agents coordinate the signal control actions with their neighbors by different strategies. In [20], two types of agents, i.e., central agent and outbound agents, are designed. The outbound agents control traffic signals by the longest-queue-first algorithm and collaborate with the central agent by providing relative traffic flow. The central agent learns value function driven by its local and neighboring traffic conditions. In this way, only the central agent is coordinated and the outbound agents work independently. Kuyer *et al.* [43] proposed a coordinated traffic signal control method based on collaboration diagrams. The neighboring agents interact with each other to obtain local state information. However, the max-plus algorithm adopted to find the optimal joint action is inclined to converge to local optimum and computationally intensive. In a word, the insufficient utilization of regional traffic status information and the poor scalability limit the application of this kind of algorithms in multi-intersection signal control.

Due to the encouraging performance of deep Q-network (DQN) on uncertain sequential decision problem [27], [28], deep reinforcement learning has recently been applied to adaptive traffic signal control. Deep stacked auto-encoder (SAE) has been introduced into reinforcement learning to estimate the optimal Q-values at the single intersection, and the traffic state is represented by the number of queued vehicles and the reward is taken as the queue difference between different roads in orthogonal directions [29]. Genders *et al.* [30] adopted deep convolutional neural network (CNN) to extract features of vehicle position and speed, and to approximate the optimal Q-value. The constructed deep reinforcement learning agent is then trained by Q-learning with experience replay for single intersection traffic control. Although the algorithm achieves better performance, it suffers from the instability due to the potential correlations between the action values and the target values. To address the instability problem, a strategy of target network was used by Gao *et al.* [31]. Besides, Jeon *et al.* [44] argued the traffic parameters in most previous RL studies cannot fully represent the complexity of an actual traffic state, and they directly used video images of an intersection to represent traffic state. More recently, Van der Pol *et al.* [33] applied multi-agent deep reinforcement learning to control the signals of simple multiple intersections without left turnings. In [33], a Q-function for smaller source problems involving two agents is trained and then the Q-function is transferred to other problems. Finally, the max-plus algorithm is used to find the optimal joint action in a coordinated fashion at multiple intersections. The max-plus algorithm is applied to cooperative multi-agent systems represented as coordination graphs, but it does not guarantee to converge to the optimal solution. Besides, transferring Q function for different sub-problems requires that the state dimension and the number of phases at each intersection are identical, which needs to restrict or approximate the structure of intersections.

To address the difficulties mentioned above, we aim to devise a cooperative deep Q-learning with Q-value transfer that is expected to make full use of state information of intersection and the influence of the neighboring intersections. The multi-intersection traffic network in a region is first modeled as a multi-agent system. Each agent only controls one intersection through a deep Q-network and transfers the latest optimal Q-value obtained from its neighbors to its own loss function during the training process. In this way, this method can optimize the overall traffic signal plans for the regional traffic scenario and balance the congested traffic for each intersection. Moreover, this algorithm can be extended to more intersections without the restriction on intersection structure.

III. DEEP REINFORCEMENT LEARNING

In reinforcement learning, the environment can be modeled as a Markov Decision Process (MDP). An MDP is defined as a five-tuple $\langle S, A, P, R, \gamma \rangle$, where S is a finite set of discrete states in the environment, A is a finite set of actions available for the agent, P is the state transition probability matrix, R denotes the reward function and $\gamma \in [0, 1]$ is a discount factor used to measure the importance of the future and immediate rewards. A reinforcement learning agent continuously interacts with the environment and learns an optimal policy by a trial-and-error process. At each time step t, the agent receives a state input $s_t \in S$ based on the observations of the environment. Then the agent selects and executes an action a_t. The state of the environment can be transformed to the next state $s_{t+1} \in S$ according to the transition probability matrix. And the agent receives an immediate reward r_t according to a reward function R. If the agent's behavior leads to positive environmental reward, then the tendency of producing this behavior by the agent will be strengthened, and vice versa. The goal is to maximize the cumulative discounted reward. The discounted future reward R_t at time t is defined as

follows:

$$R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r_k = r_t + \gamma R_{t+1} \quad (1)$$

It is impossible to obtain all rewards to calculate future discount rewards for each state, and different actions in each state will lead to different rewards. Therefore, an action-value Q-function is introduced to estimate the potential value for selecting action a at state s. Q-function is a prediction of the expected, accumulative and discounted future reward. Specifically, it can be formulated as:

$$Q(s_t, a_t) = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t, a_t] \quad (2)$$

Q-learning is a typical algorithm in RL. It is a model-free reinforcement learning algorithm which does not need to build the model of the environment's transition, but rather directly estimates the value of taking an action a at state s. The update mechanism of Q-value is formulated as follows:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[y_t - Q_t(s_t, a_t)] \quad (3)$$

where y_t is the target value. Because it is not known in advance, the agent uses the immediate reward value and the maximum Q-value of the next state as the approximation of the target value. That is $y_t = r_t + \gamma \max_{a' \in A} Q_t(s_{t+1}, a')$. $\alpha \in (0, 1]$ is the learning rate defining the level of dependence between the past knowledge and the new knowledge. The Q-learning algorithm stores the Q-value associated with each state-action pair in a look-up table. Therefore, it is also called tabular Q-learning. And it can guarantee to converge to the optimal value if the agent keeps visiting state-action pairs for an infinite number of times [45].

While tabular Q-learning works well for the problems with small-scale state and action space, it has difficulties in solving real-world problems with continuous large-scale state and action space. Under the circumstances, it is impossible to enumerate all state-action pairs.

To address this problem, existing researches usually adopt function approximation [46] or hierarchical reinforcement learning [47]–[49]. Reinforcement learning can be combined with various function approximation methods, such as linear and nonlinear function approximation. The linear approximation fits the Q-function through a series of linear combinations of features. While nonlinear function approximation typically utilizes neural networks for function approximation. In recent years, deep neural networks such as convolutional neural network (CNN), recurrent neural network (RNN), and stacked auto-encoder (SAE) have been widely used as nonlinear function approximators for large-scale reinforcement learning tasks [28], [50].

Deep reinforcement learning builds a mapping from the state vector to the Q-value for each possible action by a deep neural network, instead of estimating the Q-value of each state-action pair separately. Moreover, deep neural network can extract features from high-dimensional raw data automatically without prior knowledge and it is effective for large-scale state space problems. This paper adopts a deep Q-learning network (DQN) to approximate the Q-function. In DQN, the deep network is implemented by a CNN.

When using a neural network to approximate Q function, the Q-learning algorithm is not stable. The reasons are twofold: 1) The sequentially generated training data are correlative and they don't satisfy the assumption of independent and identical distribution. 2) A slight change in Q-value would cause an oscillation of policy, which in turn will change the distribution of incremental training data.

The strategies of experience replay and target network freezing have been developed for alleviating these problems [27], [28]. Experience replay builds a memory pool of past experiences. At each time step, the experience (s_t, a_t, r_t, s_{t+1}) generated by the agent is stored in the experience pool M. The deep Q-learning network is trained by using the data uniformly sampled from the experience pool instead of using the real time data. This strategy can disrupt the correlation between samples. The target network is an additional network. It has the same structure but different parameters with the evaluation network. The evaluation network estimates the Q-value of the current state-action pair, i.e., $Q_t(s_t, a_t)$, while the target network computes the target value y_t. That is, the target network is used to estimate $\max_{a' \in A} Q_t(s_{t+1}, a')$. Thus the deep Q-learning can be more stable by freezing the parameters of the target network for a period of time.

IV. PROPOSE QT-CDQN FOR MULTI-INTERSECTION SIGNAL CONTROL

In this section, the model based on multi-agent reinforcement learning for multi-intersection control is given first, then the cooperative deep reinforcement learning with Q-value transfer (QT-CDQN) is proposed and the training process is given in detail.

A. MODELING BASED MULTI-AGENT REINFORCEMENT LEARNING

For the convenience of describing the problem of multi-intersection signal control, here we take a heterogeneous four-intersection road network as an example. The structure of the road network is shown in Fig. 1, where the intersection 3 is a four-legged signalized intersection and the others are three-legged signalized intersections. Each intersection has a signal light. There are three roads entering the intersection for three-legged intersection, where each road consists of two lanes. For each road, the inner lane is for vehicles going straight or turning left and the outer lane is for vehicles going straight or turning right.

To perform the multi-intersection signal control via cooperative deep reinforcement learning, each intersection is first modeled as an agent. The state space S, the action space A and the reward R are defined as follows.

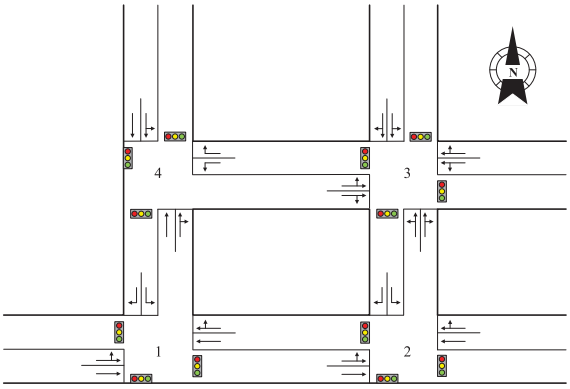
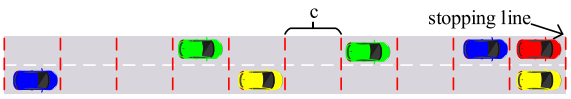


FIGURE 1. The structure of road network for the four intersections.

1) STATE SPACE

The main information needed for traffic signal control is the vehicle status information, that is, the position and the speed of the vehicles on each lane entering the intersection. In this paper, the discrete traffic state encoding is used to represent the traffic state space. For each lane entering the intersection, a segment of length l starting from the stopping line is discretized into small units of length c, as illustrated in Fig. 2(a). The selection of c should be moderate. If the value of c is much larger than the average vehicle length, it is easy to neglect the dynamic behavior of individual vehicle. If too small, it will result in extensive computation. Vehicle position and speed for the road k at the intersection i are recorded by using vehicle position matrix P_i^k and vehicle speed matrix V_i^k respectively. If there is a vehicle head on a cell, the corresponding value of the matrix P_i^k is set to 1, otherwise it is set to 0. The normalized value of vehicle speed with respect to the road speed limit is taken as the value of the corresponding cell of matrix V_i^k. For the four-legged intersection i, the vehicle position matrix P_i and speed matrix V_i are expressed by $P_i = [P_i^0, P_i^1, P_i^2, P_i^3]^T$ and $V_i = [V_i^0, V_i^1, V_i^2, V_i^3]^T$ respectively. At time step t, the state of intersection i is recorded as $s_t^i = (P_i, V_i) \in S_i$, where S_i represents the state space of intersection i.



(a)

0	0	0	1	0	0	1	0	1	1
1	0	0	0	1	0	0	0	0	1

(b)

0	0	0	0.8	0	0	0.4	0	0	0
0.9	0	0	0	0.7	0	0	0	0	0

(c)

FIGURE 2. The discrete state encoding of traffic information in the road. (a) snapshot of traffic in a certain road. (b) matrix of vehicle position. (c) matrix of normalized vehicle speed.

2) ACTION SPACE

At time step t, after observing the state s_t^i of intersection i, the agent selects one action $a_t^i \in A_i$, where A_i represents the action space of intersection i, and then executes the selected action. In this paper, the agent's possible actions are the traffic signal phase configurations. The intersections with different structures have different action spaces, as shown in Table 1. There are three different phases for three-legged intersections, and four different phases for four-legged intersections. The time for each phase is a fixed minimum unit time interval with length τ_g. At time step $t+1$, the agent observes the new state s_{t+1}^i, which is affected by the latest action a_t^i, and selects the next action a_{t+1}^i. Note that the agent may adopt the same action at time steps $t+1$ and t.

TABLE 1. Stage plans for the four intersections.

Intersection Number	Phase 1	Phase 2	Phase 3	Phase 4
1, 2				None
3				
4				None

3) REWARD

The reward function is the reward signal obtained in the process of interacting with environment. After the agent observes the state of the environment s_t^i, it selects an action a_t^i to perform. Then the agent receives a scalar reward to evaluate the selected action. The goal of the agent is to find a policy that maximizes the cumulative rewards.

There are various reward functions for traffic signal control, such as the changes of the length of queued vehicles, cumulative vehicle delay and vehicle throughput. This paper takes the changes of average queue length of vehicles at the intersection as the reward function. Let L_t^i and L_{t+1}^i be the average queue length of vehicles entering intersection i at time step t and $t+1$, respectively. Then the reward r_t^i at time step t is defined as follows:

$$r_t^i = L_t^i - L_{t+1}^i \tag{4}$$

If the reward value r_t^i is positive, it means that the action taken at time t has a positive effect on the environment and the average queue length of vehicles decreases.

B. COOPERATIVE DEEP REINFORCEMENT LEARNERS WITH Q-VALUE TRANSFER

In this section, a cooperative deep Q-network with Q-value transfer (QT-CDQN) for adaptive multi-intersection signal control is proposed. QT-CDQN first models a multi-intersection road network in a region as a multi-agent system. Each agent controls an intersection through a deep Q-network and tries to find the global optimal strategy in the dynamic environment. To make the agents control the signal cooperatively, they take into account the influence of the latest actions from their adjacencies. The structure of the QT-CDQN is illustrated in Fig. 3.

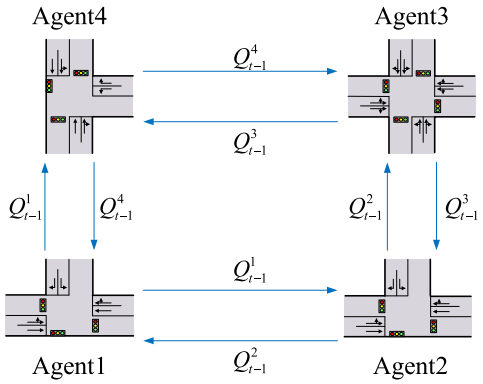


FIGURE 3. The structure of the QT-CDQN for the four intersections.

In QT-CDQN, the optimal Q-values of the neighbor agents at the latest moment are transferred to the loss function of the Q-network for policy learning, so that the multi-agent system can control the signal lights for multiple intersections cooperatively. The action selection of an intersection depends not only on its own Q value, but also on the Q value of its adjacent intersections. Such a cooperative mechanism helps to balance the traffic flow between intersections and to improve the overall performance of the regional road network.

After transferring the Q-values of the neighbor agents, the Q-values for each agent are updated as in (5):

$$\begin{aligned} Q^i_{t+1}(s^i_t, a^i_t) = {} & Q^i_t(s^i_t, a^i_t; \theta_i) + \alpha(t)[r^i_t \\ & + \gamma \max_{a' \in A_i} Q^i_t(s^i_{t+1}, a'; \theta'_i) - Q^i_t(s^i_t, a^i_t; \theta_i)] \\ & + \sum_{j \in N} \omega_{i,j} Q^j_{t-1}(s^j_{t-1}, a^j_{t-1}; \theta_j) \end{aligned} \tag{5}$$

where θ_i and θ'_i are the parameters of evaluation network and target network respectively, N is the number of neighbors for agent i, $\omega_{i,j}$ is the weights of the Q-value from agent j. Different weights can be set according to the effect of the neighbor agent j on the agent i. The closer the distance to the neighboring intersection and the more cars at the neighboring intersection, the greater the impact is.

In QT-CDQN, the separate CNN is adopted to estimate the Q value for state-action pairs at each intersection. The CNN is named as evaluation network. The CNN of each intersection has the same structure and different parameters.

Algorithm 1 Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control

Initialize DQN_i with random weights θ_i
Initialize target network with weights $\theta'_i = \theta_i$
Initialize ϵ, γ, M, N, τ_g, *min_size*, *max_size*, *step*, *sim_len* (hyper-parameter:the training time for each episode)
Initialize $Q^i_0 = 0$
for *episode* $= 1$ *to* N **do**
  for $t = 1$ *to sim_len* **do**
    Observe current intersection state s^i_t;
    The $agent_i$ randomly selects an action a^i_t with probability ϵ and selects an action $a^i_t = arg\max_{a' \in A_i} Q^i(s^i_t, a'; \theta_i)$ with probability 1-ϵ;
    Execute action a^i_t and observe $agent'_i s$ reward r^i_t and next state s^i_{t+1};
    $t = t + \tau_g$;
    if *len*(M_i) == *max_size* **then**
      delete $M_i[0]$; //delete the oldest experience
    Append $experience^i_t = (s^i_t, a^i_t, r^i_t, s^i_{t+1}, Q^i_t)$ to M_i;
    if *step*>*min_size* **then**
      Randomly sample *batch_size* experiences from M_i;
      Update θ_i using the loss function by RMSProp
      $\{r^i_t + \gamma[\max_{a' \in A_i} Q^i_t(s^i_{t+1}, a'; \theta'_i) + \sum_{j \in N} \omega_{i,j} Q^j_{t-1}(s^j_{t-1}, a^j_{t-1}; \theta_j)] - Q^i_t(s^i_t, a^i_t; \theta_i)\}$;
      // update the parameters of the target network every M steps
      Every M steps:
        set $\theta'_i = \theta_i$
    step $++$;
  if $\epsilon > 0.1$ **then**
    $\epsilon = \epsilon - 0.000625$;

The input of the network is the discrete state encoding of traffic information at the intersection and the output is the vector formed by the estimated Q-values for all actions under the observed state. The CNN estimation network can automatically extract the features from original traffic state of intersection and approximate the Q-value function by using a gradient-based training algorithm. More specially, the CNN consists of two convolution layers and two fully connected layers. The first convolution layer takes 16 filters of size 4×4 with stride 2. The second convolution layer takes 32 filters of size 2×2 with stride 1. The last two fully connected layers have 128 and 64 hidden nodes, respectively. In these layers, the activation function adopts rectified linear unit (ReLU). In the output layer, softmax activation function is used and the number of neurons is equal to the size of the action space of the corresponding intersection.

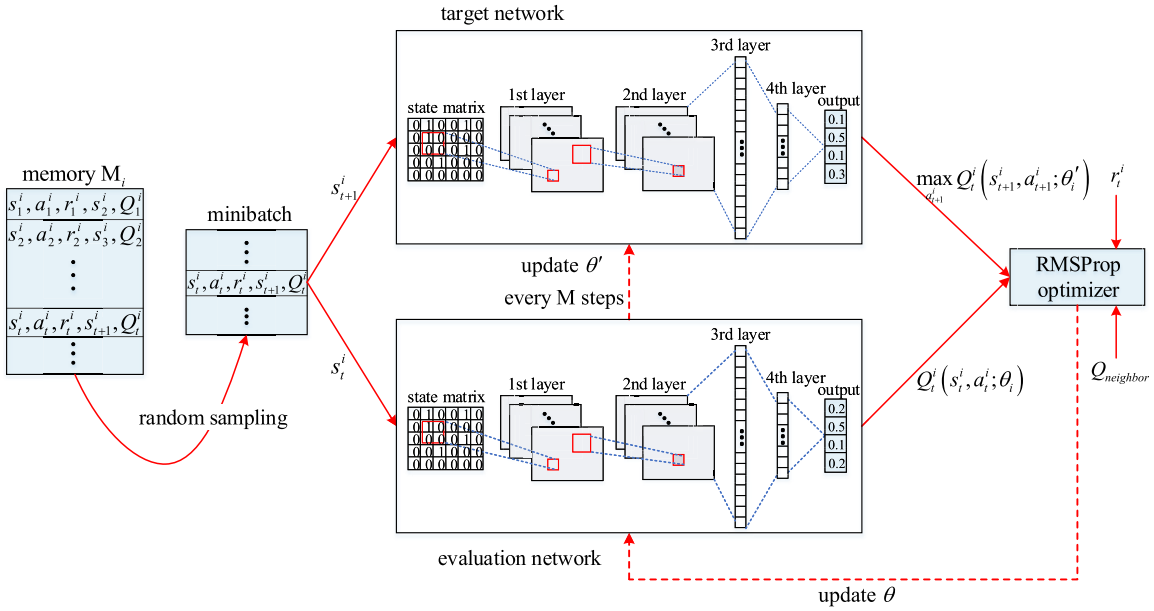


FIGURE 4. The structure of evaluation network and target network of agent i.

In order to alleviate the problem of the policy oscillation caused by a slight change in Q-values, a target network is introduced as an auxiliary network for each intersection. The target network has the same network structure and different parameters as the evaluation network as shown in Fig. 4. The target network estimates the target Q-values y_t^i, where $y_t^i = r_t^i + \gamma \max_{a' \in A_i} Q_t^i(s_{t+1}^i, a'; \theta_i')$. By freezing the parameters of the target network within a certain number of steps, the deep Q-learning algorithm becomes more stable.

Considering the effect of the adjacent intersections, the optimal Q-value of neighboring agents at the latest time step is transferred to the loss function of the current Q-network. The loss function of each agent is defined as follows:

$$MSE(\theta_i) = \frac{1}{m}\sum_{t=1}^{m}\{r_t^i + \gamma[\max_{a' \in A_i} Q_t^i(s_{t+1}^i, a'; \theta_i') + \sum_{j \in N} \omega_{i,j} Q_{t-1}^j(s_{t-1}^j, a_{t-1}^j; \theta_j)] - Q_t^i(s_t^i, a_t^i; \theta_i)\}^2 \tag{6}$$

where m is batch size, $\max Q_t^i(s_{t+1}^i, a'; \theta_i')$ is the optimal target Q-value for all actions under the state s_{t+1}^i. θ_i' is target network parameters for $agent_i$. $Q_t^i(s_t^i, a_t^i; \theta_i)$ is the output of evaluation network.

At each time step t, the state s_t^i observed by agent i is input into the evaluation network. Agent i chooses one action a_t^i to execute by using ϵ-greedy method according to the output value Q_t^i, and the agent receives a reward r_t^i and enters the next state s_{t+1}^i. Then the $experience_t^i = (s_t^i, a_t^i, r_t^i, s_{t+1}^i, Q_t^i)$ is stored in an experience memory pool M_i. The maximal capacity of each experience pool is denoted as max_size. When the experience pool is full, the earliest experience will be discarded and the latest experience will be stored. Training begins only when there are at least min_size experiences in the experience pool. To train the evaluation network more effectively, the parameter θ_i of CNN_i is updated by stochastic gradient descent algorithm *RMSProp* using $batch\_size$ experiences sampled from M_i randomly. The schematic diagram of the training process is also shown in Fig. 4. For training agent i cooperatively, the optimal Q-value of neighboring agents at the latest time step will be transferred to the loss function of agent i. Therefore, after agent i samples from M_i, it is necessary to sample the corresponding experiences from the experience memory pools of neighboring agents.

In the training process, a decreasing ϵ-greedy strategy is adopted for action selection. The agent randomly chooses one action with probability ϵ (exploration) and chooses the action with the maximum Q-value with probability $1 - \epsilon$ (exploitation). The value of ϵ decreases as the training episode goes on, which means that the role of the agent gradually turns from exploration to exploitation. *RMSProp* gradient descent algorithm with learning rate of 0.0002 is used in each estimation network. We firstly froze the parameters θ' of the target network during the training process, and they are updated to the latest values from the evaluation network by copying parameter θ to θ' for every M time steps. When the evaluation network can approximate the action value function sufficiently, the optimal control is achieved by selecting the maximum value of the output in the current state. The pseudo code of the proposed QT-CDQN is shown in Algorithm 1.

V. EXPERIMENTAL STUDIES

A. EXPERIMENTAL SETTINGS

To validate the performance of the proposed QT-CDQN for adaptive multi-intersection signal control, the experiments

TABLE 2. Comparison results of different algorithms at four intersections under different vehicle densities obtained by 15 independent runs.

		QT-CDQN			CDRL [33]			MADQN [31]			Distributed QL [51]		
		AQL	AS	AWT	AQL	AS	AWT	AQL	AS	AWT	AQL	AS	AWT
four	low	**1.37±0.02**	**9.89±0.03**	**5.24±0.15**	-[1]	-	-	1.55±0.02	9.66±0.02	5.47±0.07	1.92±0.01	9.37±0.01	8.21±0.06
	medium	**2.32 ± 0.02**	**9.38 ± 0.01**	9.7 ± 0.29	-	-	-	2.48 ± 0.02	9.19 ± 0.01	**9.31 ± 0.31**	3.13± 0.04	8.74±0.02	13.43±0.44
	high	**2.7 ± 0.03**	**9.19 ± 0.02**	**10.88 ± 0.18**	-	-	-	2.94 ± 0.02	8.96 ± 0.01	11.26 ± 0.1	3.48 ± 0.03	8.57 ± 0.01	15.2 ± 0.13
six	low	**1.32 ± 0.02**	**10.06 ± 0.01**	5.29 ± 0.17	1.42±0.01	9.93±0.01	**5.11±0.01**	1.57 ± 0.01	9.78 ± 0.01	5.89 ± 0.04	1.86±0.01	9.6±0.01	8.46±0.12
	medium	**2.41 ± 0.02**	**9.48 ± 0.02**	**9.44± 0.11**	2.69±0.01	9.32±0.01	9.68±0.12	2.77 ± 0.02	9.21 ± 0.01	10.45 ± 0.12	3.34±0.03	8.95±0.02	15.78±0.15
	high	**6.81 ± 0.1**	**7.49 ± 0.04**	**36.21 ± 2.84**	7.18±0.04	7.35±0.03	38.86±1.19	7.35 ± 0.08	7.27 ± 0.02	41.51 ± 1.22	7.28±0.14	7.55±0.05	40.15±1.13

[1] '-' represents that the CDRL is infeasible for heterogeneous intersection structures. CDRL is based on transfer learning trained for the same structures which is infeasible for the scene of the heterogeneous four intersections.

TABLE 3. The best, worst and middle results in 15 independent runs and the variance of 200 episodes in each run at four intersections. The rank for 15 runs is arranged according to the value of AQL.

			QT-CDQN			CDRL [33]			MADQN [31]			Distributed QL [51]		
			AQL	AS	AWT	AQL	AS	AWT	AQL	AS	AWT	AQL	AS	AWT
four	low	best	**1.33±0.07**	**9.92±0.07**	**5.03±0.35**	-[1]	-	-	1.52± 0.1	9.69± 0.11	5.33± 0.48	1.91±0.07	9.38±0.06	8.11±0.39
		middle	**1.37±0.07**	**9.89±0.06**	**5.09±0.34**	-	-	-	1.54± 0.11	9.66± 0.11	5.43± 0.5	1.93±0.07	9.37±0.06	8.24±0.4
		worst	**1.4±0.07**	**9.85±0.06**	**5.52±0.33**	-	-	-	1.57± 0.1	9.63± 0.11	5.57± 0.5	1.94±0.07	9.36±0.062	8.28±0.39
	medium	best	**2.3±0.27**	**9.39±0.11**	9.52±4.92	-	-	-	2.46± 0.21	9.20± 0.14	**9.21± 2.54**	3.09±0.23	8.76±0.11	12.91±5.45
		middle	**2.31±0.25**	**9.38±0.11**	9.48±4.22	-	-	-	2.49± 0.2	9.19± 0.14	**9.08± 1.24**	3.13±0.32	8.74±0.12	13.43±5.87
		worst	**2.37±0.31**	**9.38±0.12**	10.13±5.93	-	-	-	2.50± 0.22	9.19± 0.15	**9.39± 2.62**	3.2±0.46	8.7±0.13	14.21±9.89
	high	best	**2.65±0.26**	**9.21±0.11**	**10.59±4.62**	-	-	-	2.92± 0.3	8.97± 0.17	11.3± 4.67	3.43± 0.33	8.58± 0.11	14.8± 4.75
		middle	**2.69±0.31**	**9.19±0.12**	**11.06±6.89**	-	-	-	2.93± 0.24	8.96± 0.15	11.12± 2.26	3.47± 0.33	8.57± 0.12	14.92± 5.67
		worst	**2.72±0.29**	**9.19±0.12**	**10.94±5.2**	-	-	-	2.95± 0.28	8.95± 0.17	11.37± 2.74	3.5± 0.29	8.56± 0.13	15.05± 7.46
six	low	best	**1.3± 0.04**	**10.07± 0.03**	5.14± 0.22	1.41±0.04	9.93±0.04	**5.09±0.27**	1.56± 0.07	9.79± 0.08	5.84± 0.43	1.85±0.04	9.61±0.03	8.29±0.28
		middle	**1.32± 0.03**	**10.07± 0.03**	5.27± 0.21	1.42±0.04	9.92±0.03	**5.11±0.24**	1.57±0.08	9.78± 0.08	5.92± 0.44	1.87±0.04	9.59±0.04	8.66±0.26
		worst	**1.36± 0.03**	**10.04± 0.03**	5.648± 0.22	1.44±0.05	9.9±0.04	**5.12±0.27**	1.58± 0.07	9.77± 0.08	5.99± 0.42	1.88±0.04	9.59±0.03	8.51±0.26
	medium	best	**2.38± 0.07**	**9.5± 0.04**	**9.41± 0.38**	2.65±0.1	9.33±0.05	9.67±0.51	2.75± 0.13	9.23± 0.08	10.31± 0.71	3.31±0.07	8.97±0.04	15.59±0.52
		middle	**2.41± 0.07**	**9.49± 0.04**	**9.51± 0.4**	2.68±0.1	9.29±0.05	9.7±0.57	2.78± 0.13	9.22± 0.08	10.55± 0.66	3.32±0.07	8.97±0.04	15.76±0.53
		worst	**2.43± 0.07**	**9.46± 0.04**	**9.59±0.36**	2.7±0.09	9.29±0.05	9.68±0.56	2.81± 0.13	9.19± 0.08	10.62± 0.7	3.4±0.07	8.92±0.04	15.75±0.54
	high	best	**6.67± 0.54**	**7.56± 0.16**	**34.18± 8.4**	7.11±0.55	7.39±0.14	36.4±8.63	7.22± 0.57	7.30± 0.14	39.31± 9.5	7.08±0.07	7.61±0.18	38.66±8.97
		middle	**6.77± 0.57**	**7.54± 0.16**	**37.74± 9.65**	7.19±0.57	7.32±0.16	38.03±9.73	7.32± 0.5	7.28±0.13	41.21± 10.16	7.33±0.93	7.55±0.2	41.4±10.3
		worst	**7.02± 0.49**	**7.43± 0.14**	**41.4± 10.70**	7.3±0.58	7.29±0.14	41.57±8.7	7.48± 0.56	7.24± 0.14	42.46± 10.43	7.41±0.78	7.48±0.17	40.39±8.05

[1] '-' represents that the CDRL is infeasible for heterogeneous intersection structures. CDRL is based on transfer learning trained for the same structures which is infeasible for the scene of the heterogeneous four intersections.

are conducted on two different network scenarios: a 2 × 2-grid network (four intersections) and a 2 × 3-grid network (six intersections). All experiments are conducted using the Simulation of Urban Mobility (SUMO). Experiments are implemented using the Python API provided by SUMO. The CNN is implemented by Tensorflow. The experiments are executed on an Ubuntu PC with an Intel CPU (i5-2400, @3.1GHz), 16GB RAM and a Tesla P40 GPU.

B. PARAMETER SETTINGS AND PERFORMANCE METRIC

For the two road network, as shown in the asymmetric and heterogeneous four intersections of Fig. 1 and the symmetric six intersections of Fig. 5, we set the length of road to 200 m and the length of vehicle to 4 m. In the discrete traffic state encoding, the length of the discretized segment l and the length of small unit c are taken as 120 m and 5 m respectively. Normally, the traffic flow volume is constant in simulation. The ratio of traffic density is set to 1:1.5:2 in low, medium and high traffic conditions. The basic traffic density of four intersections is set to 3000 veh/h, and that of six intersections is set to 5000 veh/h.

All the experiments are trained for 2000 episodes and each episode is 4500 seconds of simulated traffic. The minimum unit time τ_g for each phase is taken as 6 seconds. The *batch_size* is 32. The *min_size* and *max_size* of experience memory are taken as 5×10^3 and 2×10^5 respectively. The exploration rate ϵ decreases from 1 to 0.1 as the training episode goes on.

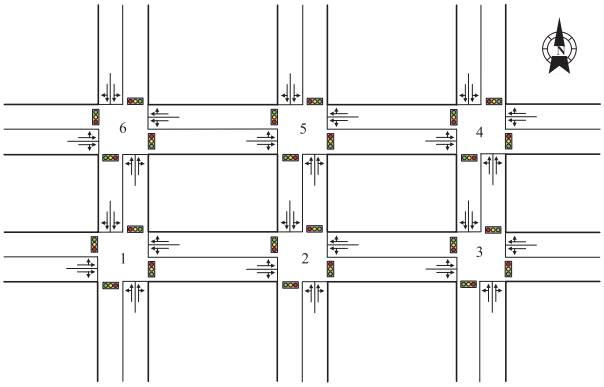


FIGURE 5. The structure of road network for the six intersections.

The performance metrics adopted in this section includes average queue length (AQL), average speed (AS) and average waiting time (AWT).

C. EXPERIMENTAL RESULTS AND ANALYSES

1) RESULTS AND COMPARATIVE ANALYSES FOR DIFFERENT ALGORITHMS

To test the performance of the QT-CDQN, the results are compared with other algorithms including coordinated deep reinforcement learners (CDRL) [33], multi-agent deep

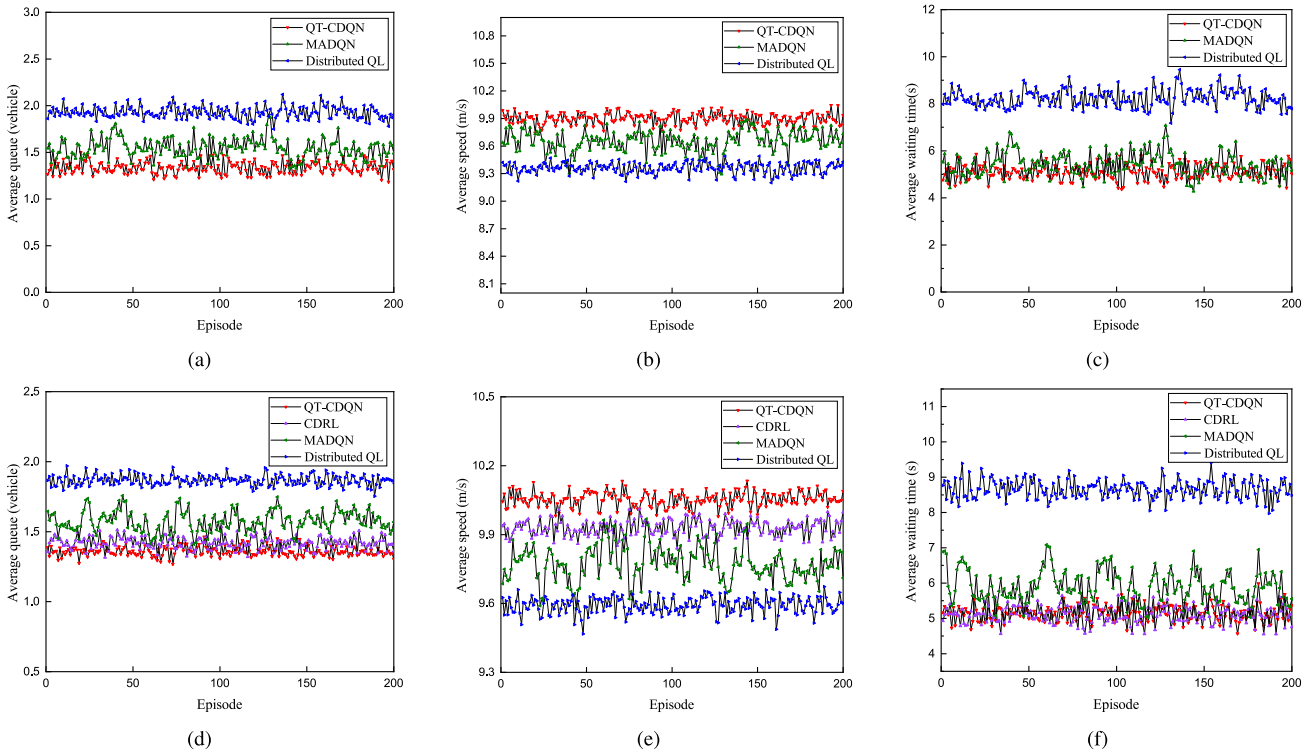


FIGURE 6. Overall performance comparisons for adaptive signal control in the scenario of multi-intersection for different metrics. (a), (b) and (c) represent the real-time changes of the AQL, AS and AWT for the four intersections under a certain traffic density respectively. (d), (e) and (f) represent the real-time changes of the AQL, AS and AWT for the six intersections under a certain traffic density respectively. The curves are obtained by running 200 episodes on the trained networks.

Q-learning (MADQN) [31] and distributed Q-learning (Distributed QL) [51]. In CDRL, a Q-function for smaller source problems involving two agents is trained and then the Q-function is transferred to other problems. Finally, the algorithm represented the cooperative multi-agent systems as coordination graphs, adopt max-plus to find the optimal joint action coordinately. CDRL is based on transfer learning trained for the same structures which is infeasible for the heterogeneous intersection structures. Therefore, the experiment of CDRL is only conducted at the scene of six intersections. MADQN designs an agent for each intersection. Each agent is trained by deep Q-learning algorithm independently and there is no cooperation among these agents. The distributed Q-learning (Distributed QL) designs a controller for each intersection. Each controller is trained by tabular Q-learning. And the controllers cooperate each other by considering the vehicles coming from the neighbor intersections in state space.

We use different traffic densities (low, medium and high) to test the adaptive control ability of the QT-CDQN. TABLE 2 presents the comparison results of different algorithms for four intersections and six intersections respectively. The results are obtained by running 200 episodes on the trained networks. All data presented are averaged over 15 independent runs. The best result is shown in bold. As the traffic density increases from low to high, the average queue length and waiting time obtained by the four algorithms increase, and the average speed decreases. For the two road networks with different traffic densities, the QT-CDQN achieves the best results in most cases for the three metrics of AQL (vehicle), AS (m/s) and AWT (s) except for the metric AWT in several cases. TABLE 3 presents the best, worst and middle(the 8th rank) results in 15 independent experiments and the variance of 200 episodes in each run for four intersections and six intersections. It can be seen from TABLE 3 that the variance obtained by the QT-CDQN in each run is less than those obtained by the other algorithms for different metrics in most cases. The results are obtained under constant traffic flow, therefore the results can reflect the stability of road network controlled by QT-CDQN.

Fig. 6 shows the overall performance comparisons for adaptive signal control in the scenario of four intersections and six intersections for different metrics. Specially, it shows the real-time changes of the AQL, AS and AWT under a certain traffic density, and the curves are also obtained by running 200 episodes on the trained networks. It can be seen that the performance of QT-CDQN is more stable throughout the signal control process. Fig. 7 shows the performance comparisons of each intersection for different metrics in the scenario of four intersections and six intersections. For four intersections, QT-CDQN achieves the best performance at each intersection for different metrics except for the intersection 1 under the metric AWT. For six intersections, QT-CDQN achieves the best performance at each intersection

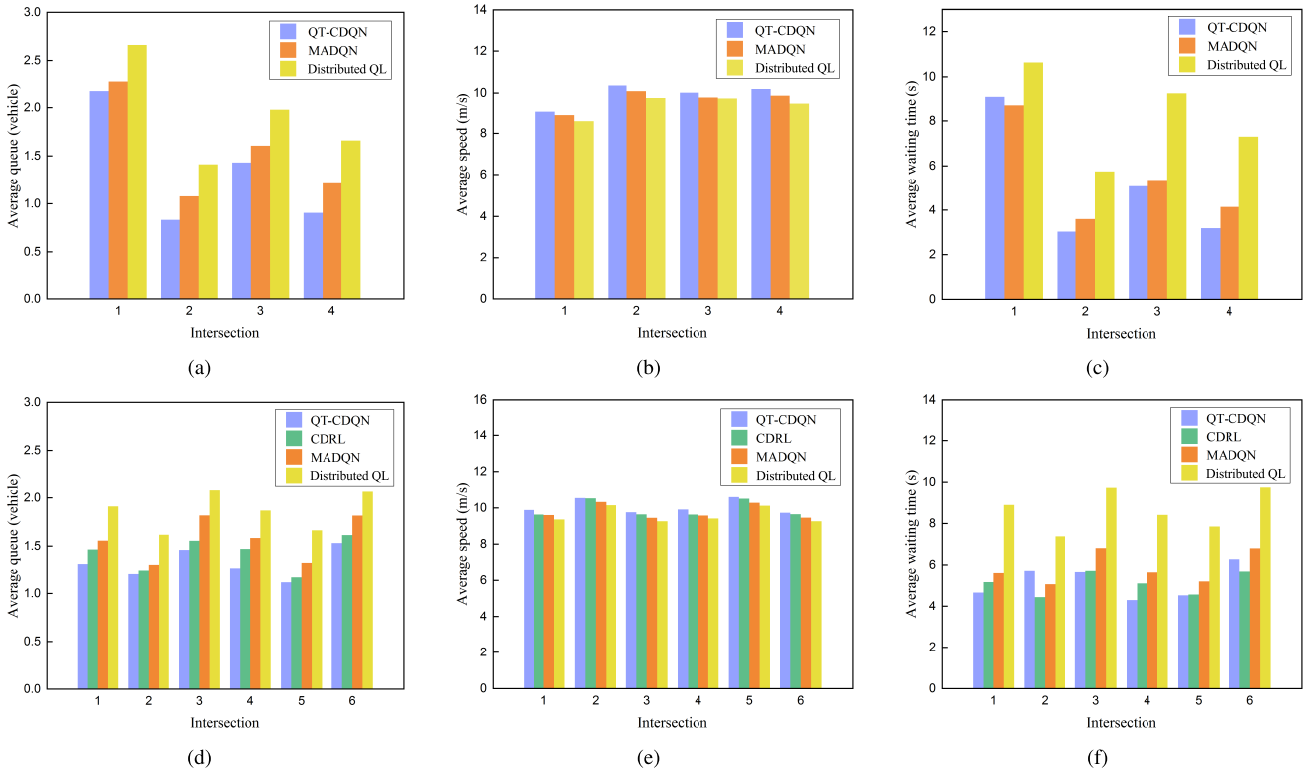


FIGURE 7. Performance comparisons of each intersection for different metrics in the scenario of multi-intersection. (a), (b) and (c) represent the AQL, AS and AWT for the four intersections respectively. (d), (e) and (f) represent the AQL, AS and AWT for the six intersections respectively.

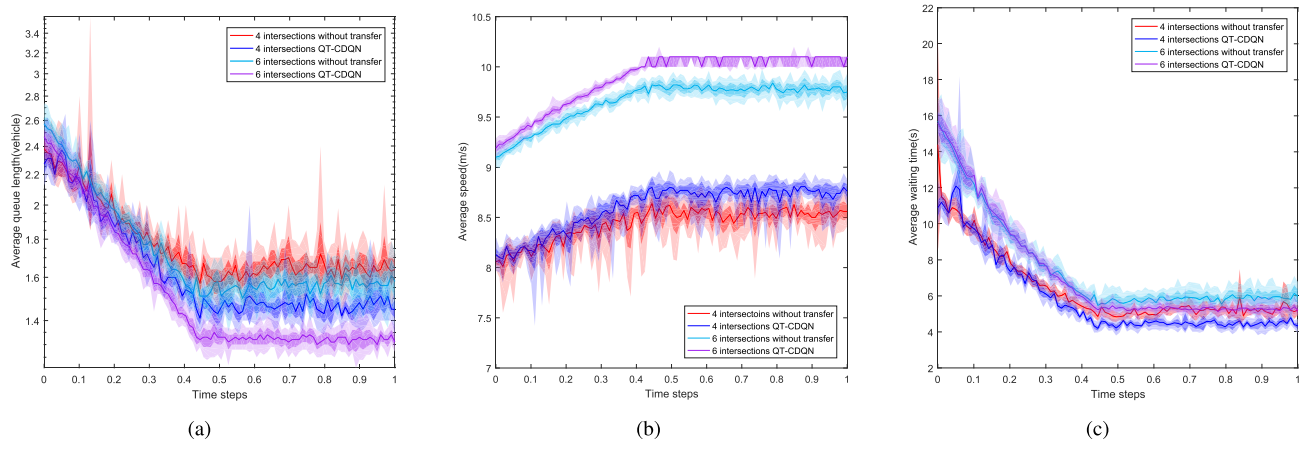


FIGURE 8. Convergence curves with or without Q-value transfer in the training process for different metrics: AQL, AS and AWT. (To make the contrast more obvious, the curves are drawn from 500th episode). The light semi-transparent areas show the minimum and the maximum, the dark areas show the confidence interval with $\alpha = 0.05$ and the thick curves show the mean. These data are gathered in the 15 runs. The abscissae are the percentage of consumed episodes.

for different metrics except for the intersection 2 and 6 under the metric AWT.

2) EFFECT OF Q-VALUE TRANSFER ON THE CONVERGENCE OF TRAINING PROCESS

To analyze the effect of Q-value transfer in cooperative deep Q-network, the QT-CDQN is compared with the model ignoring the interaction among multiple intersections. Fig. 8 shows the convergence curves with or without Q-value transfer in the training process for different metrics. In the figure, the light semi-transparent areas show the minimum and the maximum, the dark areas show the confidence interval with $\alpha = 0.05$ and the thick curves show the mean. The data are gathered during the 15 independent runs. It can be seen that the curves of AQL and AWT for the two algorithms decline sharply and then tend to be stable, while the

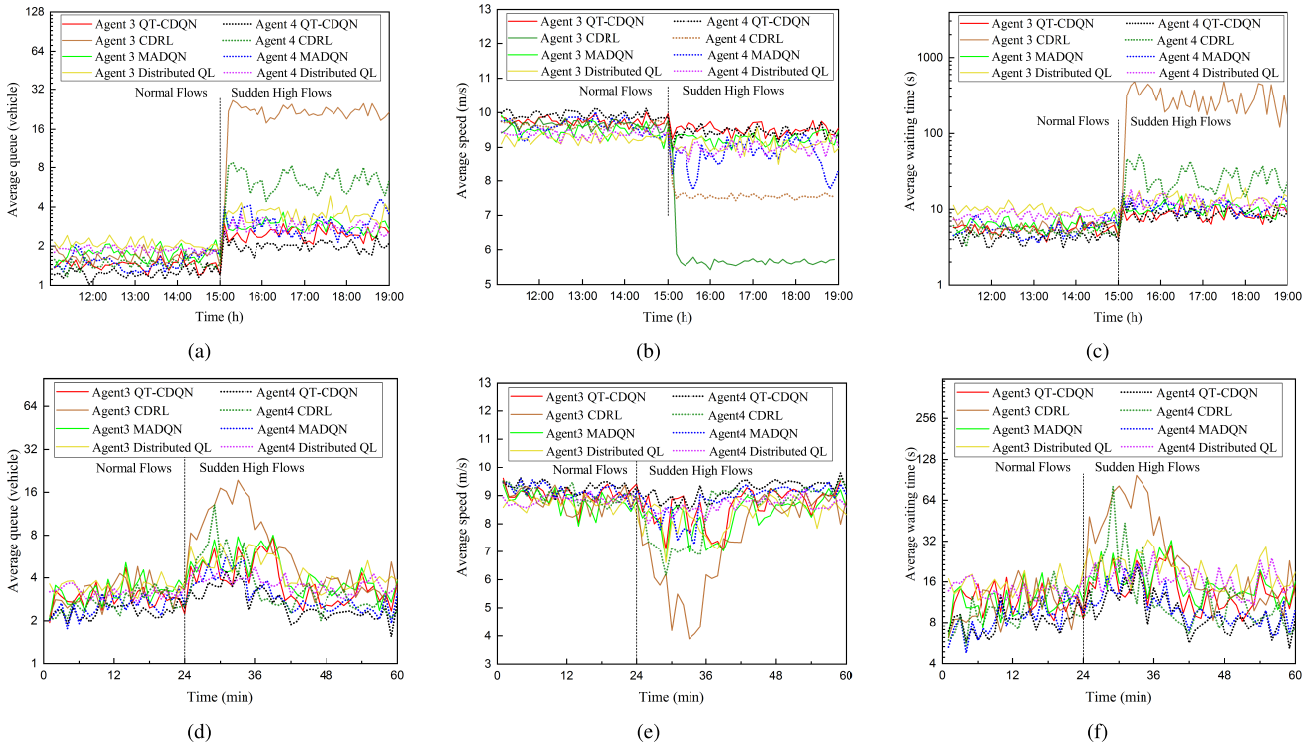


FIGURE 9. Adaptability comparisons through recurring congestion and incident congestion in the scenario of six intersections. (a), (b) and (c) represent the AQL, AS and AWT comparisons for recurring congestion respectively. (d), (e) and (f) represent the AQL, AS and AWT comparisons for the incident occasional respectively.

curve of AS rises firstly and then stabilizes with the training process going on. However, for different network structures, the proposed QT-CDQN with Q-value transfer among multiple intersections for adaptive multi-intersection signal control can achieve better results with faster convergence speed and better stability. These results validate the effectiveness of the cooperative deep Q-learning with Q-value tranfer. Besides, Q-value transfer has an impact on training time. TABLE 4 shows the training time with or without Q-value transfer. In each training, the training time mainly includes the time of sampling from experience pools and the time of updating network by stochastic gradient descent algorithm. It can be seen from Table 4 that the training time of QT-CDQN with Q-value transfer in each training is longer than that of without transfer. The time differences mainly come from the sampling time. In the training process, QT-CDQN not only samples from the current agent's experience pool, but also samples from the neighboring experience pool to obtain the neighboring Q-value. Thus, additional time is needed to obtain the sample index of experiences for neighboring agent and sample the corresponding experiences. However, the Q-value transfer has little effect on real-time signal control since the training time can satisfy the real-time requirement. For example, QT-CDQN takes about 0.033 seconds to update the network every time for six intersections. For a trained network, the network is updated online about every one minute, so Q-value transfer has little effect on the training time.

TABLE 4. Comparison results of training time with or without Q-value transfer in each training.

	four intersections		six intersections	
	with transfer	without transfer	with transfer	without transfer
network update time(s)	1.18e-2	1.21e-2	2.03e-2	2.08e-2
sampling time(s)	8.5e-3	3.8e-4	1.27e-2	5.99e-4
training time(s)	2.03e-2	1.25e-2	3.3e-2	2.14e-2

3) QT-CDQN FOR RECURRING CONGESTION AND OCCASIONAL CONGESTION

In this section, the experiments are conducted on the six-intersection road network to test the ability of QT-CDQN in dealing with recurring congestion and incident congestion on road networks.

In the experiments, the recurring congestion is added to the road from intersection 3 to intersection 4. The recurring congestion occurs between 15:00 and 19:00. In this duration, the vehicle density has been continuously increased. The curves of the different metrics before and after recurring congestion are shown in (a), (b) and (c) of Fig. 9. The instantaneous AQL and AWT increase dramatically, while AS also decreases dramatically. It can be seen that the CDRL has the worst adaptability to recurring congestion. Compared with the other three algorithms, the proposed QT-CDQN has the best adaptability to recurring congestion, and produces more stable results for the three metrics, especially at the fourth intersection.

In addition, the occasional congestion is also conducted. The occasional congestion occurs at the 24th minute. An increase in vehicle density for 6 minutes is implemented on the road from intersection 3 to intersection 4. The curves of the different metrics before and after occasional congestion are shown in (d), (e) and (f) of Fig. 9. It can be seen that the proposed QT-CDQN shows the best adaptability to occasional congestion through a sudden increase of vehicles, especially at the fourth intersection.

VI. CONCLUSIONS

This paper proposes a cooperative deep Q-network with Q-value transfer (QT-CDQN) for adaptive multi-intersection signal control. The QT-CDQN can extract the state information at the intersections effectively and enable multiple intersections to coordinate signal control according to regional traffic status. Importantly, the proposed QT-CDQN framework can be extended to the road networks with different structures and numbers of intersections. Experimental studies under various test environments show that QT-CDQN is competitive and efficient in terms of different metrics. Moreover, the experiments validate the adaptability of QT-CDQN to recurring congestion and occasional congestion. In the future work, the proposed algorithm can be extended to multi-objective signal control and can be combined with traffic assignment algorithms. In addition, multi-task reinforcement learning is expected to address the problem of multi-intersection signal control.

REFERENCES

[1] A. Stevanovic, *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice*. Olympia, WA, USA: Transportation Research Board, 2010.

[2] H. Yang, H. Rakha, and M. V. Ala, "Eco-cooperative adaptive cruise control at signalized intersections considering queue effects," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1575–1585, Jun. 2017.

[3] Y. Bi, X. Lu, Z. Sun, D. Srinivasan, and Z. Sun, "Optimal type-2 fuzzy system for arterial traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 3009–3027, Sep. 2018.

[4] M. J. S. Shirvani and H. R. Maleki, "Maximum green time settings for traffic actuated signal control at isolated intersections using fuzzy logic," *Int. J. Fuzzy Syst.*, vol. 19, no. 1, pp. 247–256, 2017.

[5] W. Yang, L. Zhang, Z. He, and L. Zhuang, "Optimized two-stage fuzzy control for urban traffic signals at isolated intersection and Paramics simulation," in *Proc. 15th IEEE Int. Conf. Intell. Transp. Syst.*, Anchorage, AK, USA: ITSC, Sep. 2012, pp. 391–396.

[6] M. H. Khooban and A. Liaghat, "A time-varying strategy for urban traffic network control: A fuzzy logic control based on an improved black hole algorithm," *Int. J. Bio-Inspired Comput.*, vol. 10, no. 1, pp. 33–42, 2017.

[7] H. Wan, "Research on traffic signal control algorithm based on self-organizing competitive neural network and optimization model," *Agro Food Ind. Hi-Tech*, vol. 28, no. 3, pp. 2911–2914, 2017.

[8] G. Shen and X. Kong, "Study on road network traffic coordination control technique with bus priority," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 3, pp. 343–351, May 2009.

[9] K.-H. Chao, R.-H. Lee, and M.-H. Wang, "An intelligent traffic light control based on extension neural network," in *Proc. 12th Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst.*, Zagreb, Croatia, vol. 5177, 2008, pp. 17–24.

[10] K. T. K. Teo, W. Y. Kow, and Y. K. Chin, "Optimization of traffic flow within an urban traffic light intersection with genetic algorithm," in *Proc. Int. Conf. Comput. Intell., Modelling Simulation (CIMSiM)*, Bali, Indonesia, 2010, pp. 172–177.

[11] Y. Zhang and Y. Zhou, "Distributed coordination control of traffic network flow using adaptive genetic algorithm based on cloud computing," *J. Netw. Comput. Appl.*, vol. 119, pp. 110–120, Oct. 2018.

[12] H. Asadi, R. T. Moghaddam, N. S. Pourorcid, and E. Najafi, "A new nondominated sorting genetic algorithm based to the regression line for fuzzy traffic signal optimization problem," *Sci. Iranica*, vol. 25, no. 3, pp. 1712–1723, 2018.

[13] A. S. Mihăiță, L. Dupont, and M. Camargo, "Multi-objective traffic signal optimization using 3D mesoscopic simulation and evolutionary algorithms," *Simul. Model. Pract. Theory*, vol. 86, pp. 120–138, Aug. 2018.

[14] L. Cao *et al.*, "Two intersections traffic signal control method based on ADHDP," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Beijing, China, Jul. 2016, pp. 1–5.

[15] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[16] J. X. Wang *et al.*, "Prefrontal cortex as a meta-reinforcement learning system," *Nature Neurosci.*, vol. 21, no. 6, pp. 860–868, 2018.

[17] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with average cost for adaptive control of traffic lights at intersections," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Washington DC, USA, Oct. 2011, pp. 1640–1645.

[18] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 412–421, Jun. 2011.

[19] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Traffic light control in non-stationary environments based on multi agent Q-learning," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Washington DC, USA, Oct. 2011, pp. 1580–1585.

[20] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, 2010.

[21] J. C. Medina and R. F. Benekohal, "Reinforcement learning agents for traffic signal control in oversaturated networks," in *Proc. Transp. Develop. Inst. Congr. Integr. Transp. Develop. Better Tomorrow*, Chicago, IL, USA, 2011, pp. 132–141.

[22] J. C. Medina and R. F. Benekohal, "Traffic signal control using reinforcement learning and the max-plus algorithm as a coordinating strategy," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Anchorage, AK, USA, Sep. 2012, pp. 596–601.

[23] S. El-Tantawy and B. Abdulhai, "Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC)," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Anchorage, AK, USA, Sep. 2012, pp. 319–326.

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

[25] K. J. Prabuchandran, K. A. N. Hemanth, and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst.*, Qingdao, China, Oct. 2014, pp. 2529–2534.

[26] Y. Liu, L. Liu, and W.-P. Chen, "Intelligent traffic light control using distributed multi-agent Q learning," in *Proc. 20th Int. IEEE Conf. Intell. Transp. Syst.*, Yokohama, Japan, Oct. 2017, pp. 1–8.

[27] V. Mnih *et al.* (2013). "Playing Atari with deep reinforcement learning." [Online]. Available: https://arxiv.org/abs/1312.5602

[28] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[29] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Automat. Sinica*, vol. 3, no. 3, pp. 247–254, Apr. 2016.

[30] W. Genders and S. Razavi. (2016). "Using a deep reinforcement learning agent for traffic signal control." [Online]. Available: https://arxiv.org/abs/1611.01142

[31] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori. (2017). "Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network." [Online]. Available: https://arxiv.org/abs/1705.02755?context=cs

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

[33] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. NIPS Workshop Learn., Inference Control Multi-Agent Syst.*, 2016.

[34] N. Casas. (2017). "Deep deterministic policy gradient for urban traffic light control." [Online]. Available: https://arxiv.org/abs/1703.09035
[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[36] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 3377–3381.
[37] S. Araghi, A. Khosravi, M. Johnstone, and D. Creighton, "Intelligent traffic light control of isolated intersections using machine learning methods," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Manchester, U.K., Oct. 2013, pp. 3621–3626.
[38] J. Jin and X. Ma, "Adaptive group-based signal control by reinforcement learning," in *Proc. Transp. Res. Procedia*, vol. 10, pp. 207–216, Jan. 2015.
[39] S. Touhbi *et al.*, "Adaptive traffic signal control: Exploring reward definition for reinforcement learning," in *Proc. Int. Conf. Ambient Syst., Netw. Technol.*, Madeira, Portugal, vol. 109, 2017, pp. 513–520.
[40] C.-H. Wan and M.-C. Hwang, "Value-based deep reinforcement learning for adaptive isolated intersection signal control," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 1005–1010, 2018.
[41] P. Ha-Li and D. Ke, "An intersection signal control method based on deep reinforcement learning," in *Proc. 10th Int. Conf. Intell. Comput. Technol. Automat.*, Changsha, China, 2017, pp. 344–348.
[42] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*. Cham, Switzerland: Springer, 2016, pp. 47–66.
[43] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*, Antwerp, Belgium, 2008, pp. 656–671.
[44] H. J. Jeon, J. Lee, and K. Sohn, "Artificial intelligence for traffic signal control based solely on video images," *J. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 433–445, 2018.
[45] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, p. 1054, Sep. 1998.
[46] X. Xu, L. Zuo, and Z. Huang, "Reinforcement learning algorithms with function approximation: Recent advances and applications," *Inf. Sci.*, vol. 261, pp. 1–31, Mar. 2014.
[47] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dyn. Syst.*, vol. 13, no. 4, pp. 343–379, 2003.
[48] B. Ghazanfari and N. Mozayani, "Enhancing nash Q-learning and team Q-learning mechanisms by using bottlenecks," *J. Intell. Fuzzy Syst.*, vol. 26, no. 6, pp. 2771–2783, 2014.
[49] S. S. Mousavi, B. Ghazanfari, N. Mozayani, and M. R. Jahed-Motlagh, "Automatic abstraction controller in reinforcement learning agent via automata," *Appl. Soft Comput.*, vol. 25, pp. 118–128, Dec. 2014.
[50] F. Abtahi and I. Fasel, "Deep belief nets as function approximators for reinforcement learning," in *Proc. AAAI Conf. Lifelong Learn.*, vol. 2, Aug. 2011, pp. 2–7.
[51] S. Araghi, A. Khosravi, and D. Creighton, "Distributed Q-learning controller for a multi-intersection traffic network," in *Proc. 22nd Int. Conf. Neural Inf. Process. (ICONIP)*, Istanbul, Turkey, 2015, pp. 337–344.

HONGWEI GE received the B.S. and M.S. degrees in mathematics from Jilin University, China, and the Ph.D. degree in computer application technology from Jilin University, in 2006. He is currently a Professor with the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests are machine learning, computational intelligence, optimization and modeling, deep learning, and intelligent traffic. He has published more than 80 papers in these areas. His research was featured in the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART A: SYSTEMS AND HUMANS, PATTERN RECOGNITION, INFORMATION SCIENCE.

YUMEI SONG received the B.S. and M.S. degrees from the Dalian University of Technology, Dalian, China, in 2012 and 2015, respectively, where she is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. Her main research interests include machine learning, deep reinforcement learning, and intelligent traffic.

CHUNGUO WU received the B.S. and M.S. degrees from the Department of Mathematics, and the Ph.D. degree from the College of Computer Science and Technology, Jilin University, Changchun, China, in 2006, where he is currently an Associate Professor. His research interests include machine learning, evolutionary computation, and adaptive systems. He has published two books, two inventions, and more than 60 articles in *Information Sciences* and *Physical Review E*.

JIANKANG REN received the B.Sc., M.E., and Ph.D. degrees in computer science from the Dalian University of Technology, China, in 2008, 2011, and 2015, respectively. He was a Visiting Scholar with the Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA, USA, from 2013 to 2014. He is currently a Lecturer with the School of Computer Science and Technology, Dalian University of Technology. His research interests include machine learning, cyber-physical systems (CPS), and computational intelligence.

GUOZHEN TAN received the B.S. degree from the Shenyang University of Technology, Shenyang, China, the M.S. degree from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, where he is currently a Professor with the College of Computer Science and Technology. His current research interests include the cyber-physical systems, intelligent transportation systems, and machine learning.