

Distributed Cooperative Reinforcement Learning-Based Traffic Signal Control That Integrates V2X Networks' Dynamic Clustering

Weirong Liu*, *Member, IEEE*, Gaorong Qin, Yun He and Fei Jiang

Abstract—With the acceleration of urbanization in the world, urban traffic congestion has become an urgent challenge in most cities. Adaptive traffic signal control is the most approved control method to solve the problem, and accurate real-time traffic information is critical to this solution. This paper presents distributed cooperative reinforcement learning-based traffic control that integrates V2X networks' dynamic clustering algorithm. To obtain traffic flow information accurately and instantaneously, it is important to improve the cluster stability in V2X networks. A dynamic clustering algorithm is proposed based on the enhanced affinity propagation. The proposed clustering algorithm introduces the initial cluster partition to maintain a proper cluster size and adds the lane and destination factors to improve the cluster's stability. The algorithm can provide efficient and accurate traffic state information to traffic signal controls. By integrating the clustering algorithm, a cooperative reinforcement learning control scheme is proposed to balance the traffic load. To address the tough dimensionality curse of reinforcement learning, a distributed mechanism for intersection cooperation is introduced, and a fast gradient-descent function approximation method is proposed to improve the controls' real-time performance. The proposed intelligent traffic control scheme that integrates the stable clustering algorithm can effectively improve the traffic throughput, reduce the average waiting time and avoid congestion. Numerical simulations on real scenarios validate the performance of the proposed approach.

Index Terms—V2X networks, dynamic clustering algorithm, traffic signal control, cooperative reinforcement learning, function approximation

I. INTRODUCTION

Frequent traffic jams have emerged in most cities throughout the world due to the rapidly increasing population and vehicle usage [1]. One could try to solve the problem by extending the urban traffic infrastructures, but this solution may be expensive and could not go into effect immediately. A more feasible solution is to optimize the traffic signal controls based on real-time traffic information to balance the traffic load and decrease the waiting times, that is, to develop an intelligent

transportation system (ITS). Currently, the traffic-information-based signal control optimization is becoming more critical for large urban road networks.

An early method of traffic signal control is the fixed-time mechanism [2], which used a preset green time and a cycle time in operation and did not adapt to the change in traffic, e.g., a burst traffic flow. Many traffic signal control systems take adaptive mechanisms whose control inputs, such as green times and cycle times, are calculated according to the real-time traffic conditions sensed at urban intersections. SCOOT [3], SCAT [4] and GLIDE [5], are typical adaptive systems and have been implemented on practical urban networks successfully. Currently, most practical ITS systems use roadside or underground sensors installed close to intersections, such as loop detectors, to sense the traffic flow. However, roadside and underground sensors are difficult to maintain and update [6]. In addition, the underground installation may reduce a road's strength.

With the development of wireless communication, one approach that can help to collect real-time traffic information for adaptive signal controls is to enhance VANET-based V2X Networks [7-9], which can achieve lower costs and less complexity by using a short-range wireless communication standard [10]. This approach has offered new methods for vehicle detection and intersection cooperation. When vehicles wait in front of a traffic light, they send their identification numbers (ID), types, positions, speeds as well as the timestamps to the intersection agents. Thus, the real-time traffic road and network information can be obtained by VANET.

More studies have been devoted to the collection of information with V2X networks that serve intelligent transportation systems with convenient traffic environments [11,12]. V2X networks can include two communication modes: V2I (Vehicles to Infrastructure) mode and V2V (Vehicles to Vehicles, that is, VANET) mode, which will form a large mobile ad hoc communication network to support traffic control optimization [9]. In common with other large wireless networks, the cluster structure is a highly efficient communication organization for V2X networks. Recently, many works have proposed varieties of cluster-dividing mechanisms. Whereas some mechanisms are based on node self-information [13-16], other mechanisms mainly focus on communication performance [17-19]. In contrast to traditional ad hoc networks, most VANET nodes are vehicles with high mobility and their own destinations. Thus, it has become critical to maintain the cluster stability for clustering mechanisms in VANET [20].

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are with School of Information Science and Engineering, Central South University, Changsha 410083.

E-mail: frat@csu.edu.cn (Corresponding author*); gaorong_q@163.com; sylan401@126.com; jiangfei0819@126.com.

This work is partially supported by the National Natural Science Foundation of China No. 61672539, 61003233, 61379111, 61202342, Specialized Research Fund for Doctoral Program of Higher Education No. 20110162110042, and CSC State Scholarship Fund No. 201406375017.

The key point to improving the cluster stability is to avoid unnecessary re-divisions of clusters. Z. Zhang et al. proposed a stability enforcement mechanism that introduces multiple-hop distances as a clustering metric to extend the cluster's lifetime [19]. In [20], S. Kuklinski et al. used a comprehensive metric combining the network connection, the node density, the passing condition and the communication quality to form stable and long-life clusters. A life counter is applied for every node in one cluster to adjust the cluster range dynamically rather than re-divide the cluster frequently.

Similarly, VWCA [21] is a clustering mechanism that uses even more complex metrics to form clusters. For example, this algorithm adds the trustless value, neighbor number and dynamic transmitting distance as the cluster-head-selection factors. Furthermore, DMCA [22] adds the adjusting mechanism to accommodate a node with different turn directions at an intersection. The above clustering algorithm may produce an excessive clustering cost for sparse VANET.

Comparatively, in [23], B. Hassanabadi et al. proposed a mobile clustering algorithm based on affinity propagation whose clustering metric includes the responsibility value and availability value. The idea is to avoid considering complex network connection conditions. However, this algorithm does not consider the vehicles' turn directions, and may generate large clusters.

Even when it is possible to obtain real-time and sufficient traffic information, traffic signal control optimization is a challenging work. Because of the difficulty of transportation modeling, the closed-form solution for an overall urban environment is almost unexpected. Thus, varieties of artificial intelligence (AI) methods have been introduced into the field, such as evolutionary algorithms, neural networks, fuzzy logic, and reinforcement learning. [24-27]. Among these AI methods, reinforcement learning (RL) continuously interacts with the environment to adapt to new circumstances rapidly [28]; hence, this type of method is suitable for urban traffic requirement. These methods have shown great potential for traffic signal controls in a stochastic traffic environment. Recently, many variants of the RL algorithm have been proposed for implementation of intelligent traffic control [29-32].

In [29], Wiering used a model-based RL algorithm for a small grid traffic network to realize traffic control. In [30], RL is utilized by introducing the Bayesian interpretation of probability to adapt to the high dynamics of a traffic network. This research has been extended by [31] to multi-objective traffic optimization. In [32], S. Richter et al. used an actor-critic method that integrates a neural network into reinforcement learning to optimize traffic control.

Some approaches utilize the intersection's own state information and reward to produce an action strategy. This approach will be more flexible for burst events when decision cooperation among intersection agents is introduced. For instance, in [33], the metro intersections can utilize the Q-values of their outer intersections, which are transferred by vehicles to the city center. A. Salkham et al. utilized Collaborative Reinforcement Learning (CRL) based on an Adaptive Round Robin (ARR) phase switching model [34]. Although it could be synergetic and beneficial that the agents are regulated by

optimally coordinating the operations of multiple intersections simultaneously, such cooperation increases the complexity of the intelligent control and may result in the curse of dimensionality. The curse of dimensionality for intersection-based controls is a phenomenon in which the computation exponentially grows with an increase of the dimension. Thus, Mohamed A. Khamis and Walid Gomaa in [35] proposed a vehicle-based state-space representation in which the number of states will grow linearly in terms of the number of lanes and vehicles' positions, enhancing the reinforcement learning application in intelligent traffic control.

Another solution method to address the high-dimension problem is utilizing function approximation techniques [28]. Function approximation has shown great potential for the RL algorithm to address the curse of dimensionality in traffic signal control systems. In [36], an adaptive traffic signal controller was designed by using dynamic programming to update the approximation. In [37], Q-learning with a feed-forward neural network for function approximation was proposed to reduce the traffic congestion and average delay. In [38], L. A. Prashanth et al. incorporated state-action features to approximate the traffic state information in a reinforcement learning algorithm for traffic signal control.

Nevertheless, most of these approximation methods can have slow convergence. To solve this problem, a promising approximation method named "fast gradient descent" could be applied [39][40]. However, it is necessary to research how to select the approximation features for large traffic systems and how to use the fast gradient-descent method in a distributed formulation.

In this paper, to adapt to the dynamics of a complex multi-intersection urban road network, a stable clustering algorithm for V2X networks and a distributed cooperative reinforcement learning-based control with the gradient-descent function approximation are proposed to optimize the traffic signal control. The main contributions are as follows:

- 1) In the stable clustering algorithm, the initial cluster size is limited according to the vehicle communication range. Then the cluster head is elected by an affinity propagation algorithm with the enhanced similarity function considering the lane and destination. Furthermore, by introducing a cluster maintenance mechanism, this algorithm can effectively avoid unnecessary cluster re-divisions and provide the traffic information accurately and instantaneously for traffic control.

- 2) The proposed reinforcement learning traffic control uses cooperative Q-values among adjacent intersections. By adopting the cooperative mechanism, the action selection strategy of one intersection is dependent not only on its own reward, but also on its neighboring intersections' rewards, which is helpful for balancing the traffic among intersections. To address the curse of dimensionality due to the cooperative mechanism, the proposed cooperative reinforcement learning uses a distributed mode for intersections and applies a function approximation with a fast gradient-descent method to accelerate the convergence. Because the proposed mechanism effectively balances the traffic flow and improves the utilization of each road in large-scale networks, it avoids traffic congestion and reduces the vehicles' waiting times.

The rest of the paper is organized as follows: Section II presents the dynamic clustering algorithm for V2X networks, and Section III describes cooperative reinforcement learning with the function approximation algorithm. In Section IV, the proposed algorithm is applied to traffic signal control. In Section V, the simulation results integrating V2X networks and traffic controls are shown and analyzed, and in Section VI the conclusions are drawn and future work is described.

II. DYNAMIC CLUSTERING ALGORITHM FOR V2X NETWORKS (DC-TDCA)

A. V2X Network Model and Assumptions for the Dynamic Clustering Algorithm

As shown in Fig. 1, V2X networks are divided into two parts: V2V communications among vehicles on the road and V2I communications between the vehicles and the intersection control agents. On the road, the data transmission mainly depends on the vehicles. The vehicles are clustered to form a temporal network topology with a certain stability, which decreases the cost of network reorganization. When vehicles move into the communication range of the intersection control agents, the vehicles can transfer data to the agents directly. Remote traffic information is brought to the intersection control agents via multi-hop communications between the vehicle cluster heads.

Based on the real-time traffic information collected by V2X networks, intersection control agents can implement adaptive traffic signal control. The schedule of the phase of the traffic signal, that is, the decision when to permit vehicles to pass in certain directions, is vital to the traffic performance. The phase decision can be modeled as an MDP framework, which enables the utilization of intelligent control based on reinforcement learning. The queue lengths of the lanes and vehicles' waiting times can be taken as traffic performance indices to evaluate the actions adopted by the intelligent agents of intersections.

According to the complicated conditions of the urban environment, this paper makes the following assumptions:

- Each vehicle node is equipped with a vehicle GPS and can communicate with other vehicles.
- Each vehicle node knows its own travel lane.
- Each intersection has an intersection control agent to achieve V2I communication between the vehicles and the intersections.
- Only vehicles traveling in the same direction can be divided into the same cluster.
- Some vehicles can support traffic guidance and receive the drivers' input to obtain the destination information.

According to the DSRC standard [41], the data link layer can provide a transmission range of up to 1,000 m for a channel. V2X applications can use a longer range r_c for the control channel so that a cluster head can communicate with neighboring cluster heads for safety message disseminations. On the other hand, V2X applications can use a shorter communication range r_s for a service channel that is intended for intra-cluster information exchanges.

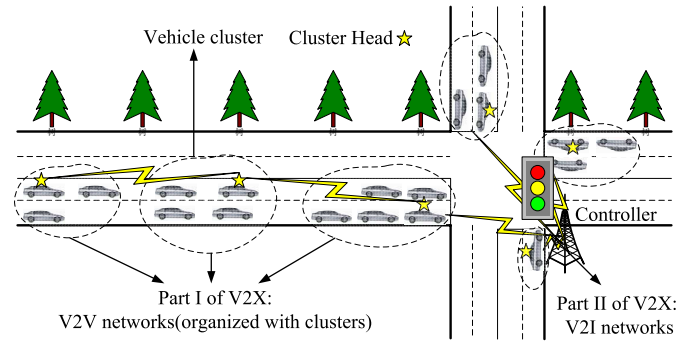


Fig. 1. V2X networks communication and control model.

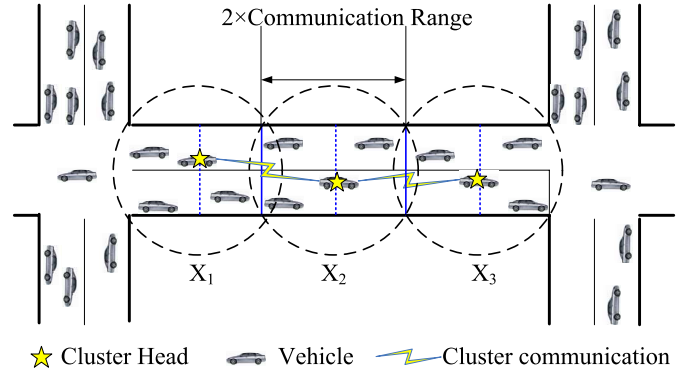


Fig. 2. The initial clusters for vehicles.

B. Dynamic Clustering algorithm

The main objective of clustering is to achieve a relatively stable cluster structure under the mobility of nodes. Because cluster reconfiguration usually generates mass communication loads, the cluster stability is important for reducing communication cost and improving the network performance. Because the effective cluster size is related to both the radio transmission range and the vehicle traffic density, it is time varying due to the traffic variations. To form a suitably sized and stable VANET cluster, the proposed clustering procedure is divided into the clustering period, which includes 3 phases: 1) initial cluster formation, 2) cluster head election, and 3) cluster maintenance. That is, at beginning of every clustering period, the initial clusters with proper sizes are formed. Then, the head of every cluster is elected by the similarity of the nodes to realize stability. When one node leaves the old cluster and enters a new cluster, the cluster maintenance mechanism is activated to let the node join a proper cluster.

1) *Initial Cluster Formation*: To avoid forming a cluster that is too large or too small, each road segment (each section of street between two intersections) is dissected into small cells according to the road length and short communication range r_s . Generally, the length of the cell is $2 \times r_s$ to guarantee that the inter-cluster communication will be in 1-hop. This cell boundary is identified by the demarcation of the road, which is depicted by the blue lines in Fig. 2.

Depending on the vehicle density, one cell can have one or more vehicle clusters. A maximum number of vehicles in one cluster N_c is preset. If the vehicle number in one cell

is greater than N_c , the vehicles in the cell will be divided until the number of vehicles in one cluster does not exceed the threshold N_c . At every beginning of a clustering period, the vehicles are allocated into road-cell-based clusters that are formed along the road. This allocation can be realized using GPS information, as shown in Fig. 2.

2) *Cluster head election*: To elect the most suitable head in one cluster, affinity propagation is adopted. This propagation is based on message passing and was initially proposed by B. J. Frey et al. in 2007 [42]. The main idea of this algorithm is to pass a responsibility message and availability message with each other continuously and gradually to achieve the optimal election of cluster head node. Then, the cluster head node has the maximum similarity $s(i, j)$ with the other nodes in the cluster.

For the mobile ad hoc network, when the cluster head has more similarity with the other nodes in the cluster, these nodes will be closer to the head and have less probability of leaving the communication range of the cluster head. Consequently, the cluster structure will have more stability for a comparatively long time. Therefore, the intersection agent can obtain more reliable and prompt traffic information, which will improve the traffic control performance indirectly. To apply the algorithm to vehicle networks, it is critical to define the similarity function to reflect the relations among the vehicle nodes. In the proposed algorithm, the relative position, the lane and the destination information are used to form the similarity function.

In this paper, $s(i, j)$ is the similarity of vehicle node i to node j . To increase the head stability, the following factors are considered: the current positions of nodes i and j , the next positions of nodes i and j , the lanes of nodes i and j , and the destinations of nodes i and j . From the discussion, the similarity function can be computed by the following formula:

$$s(i, j) = -l_j k_j (\mu_{s1} \|X_i - X_j\| + \mu_{s2} \|X'_i - X'_j\| + \mu_{s3} S_lane(Lane_i, Lane_j) + \mu_{s4} \|Des_i - Des_j\|) \\ X_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad X'_i = \begin{bmatrix} x_i + v_{x,i}\tau \\ y_i + v_{y,i}\tau \end{bmatrix} \quad (1)$$

X_i is the present position vector of the node at the current time. X'_i is the estimated position vector of the node after the estimated time τ , which is based on the current speed of the node. In addition, k_j is the number of neighbor nodes of node j , and l_j denotes node j 's driving lane weight. According to the number of vehicles per lane in the cluster, the lane weight is designed as follows:

$$l_j = \vartheta \cdot e^{-m_j/m_{sum}} \quad (2)$$

ϑ is a weight factor, m_i is the number of vehicles within the lane where node j is located, and m_{sum} is the total number of vehicles in the cluster.

$Lane_i$ is the lane where node i is driving, and $Lane_j$ is the lane where node j is driving. S_lane is a binary judgement function, which is described as follows:

$$S_lane(Lane_i, Lane_j) = \begin{cases} 0 & \text{if } lane_i = lane_j \\ 1 & \text{if } lane_i \neq lane_j \end{cases}$$

Des_i is the destination of node i , and Des_j is the destination of node j . The destination can be obtained from the traffic guidance system, if nodes i and j are both equipped with it. μ_{s1} , μ_{s2} , μ_{s3} , and μ_{s4} are normalizing parameters.

The iterative process in the affinity propagation algorithm updates the responsibility value $r_{res}(i, j)$ and the availability value $a(i, j)$ continuously and alternately. For any vehicle, this process calculates the sum of $r_{res}(i, j)$ and $a(i, j)$ for all of its neighbor nodes. Then, the cluster head node will be elected. The update procedure is as follows:

Step 1: Calculate the responsibility value for the neighbor nodes:

$$r_{res}(i, j) = s(i, j) - \max_{j' \in K, j' \neq j} \{a(i, j') + s(i, j')\} \quad (3)$$

Step 2: Update and store the responsibility value:

$$r_{res}(i, j) = (1 - \lambda)r_{res}(i, j)^{new} + \lambda r_{res}(i, j)^{old} \quad (4)$$

where $r_{res}(i, j)^{new}$ is the current responsibility value and $r_{res}(i, j)^{old}$ is the responsibility value of the last time iteration. To smooth the responsibility value, the damping factor $\lambda \in [0, 1]$ is introduced.

Step 3: Calculate the availability values for the neighbor nodes:

$$a(i, j) = \min\{0, r_{res}(j, j) + \sum_{i' \neq j} \max\{0, r_{res}(i', j)\}\} \quad (5)$$

Step 4: Update and store the responsibility values:

$$a(i, j) = (1 - \lambda)a(i, j)^{new} + \lambda a(i, j)^{old} \quad (6)$$

Step 5: Calculate the self-availability values:

$$a(j, j) = \sum_{i' \in K, i' \neq j} \max\{0, r_{res}(i', j)\} \quad (7)$$

Step 6: Determine the cluster head node. The node that has the maximum value for the sum of $r_{res}(i, j)$ and $a(i, j)$ is elected as the cluster head node if $r_{res}(i, j)$ and $a(i, j)$ of node j are positive; otherwise, return to Step 1.

$$CH_i = \arg \max_j \{a(i, j) + r_{res}(i, j)\} \quad (8)$$

3) *Cluster Maintenance*: Due to the highly dynamic nature of VANET, vehicles continuously join and leave clusters frequently, which causes extra maintenance overhead. The events that trigger the maintenance procedure can be summarized as follows:

a) *Joining a cluster*:

When a standalone (non-clustered) vehicle comes within the short range r_s of a nearby cluster head, the cluster head accepts the vehicle and adds it to the cluster member list. If there is more than one cluster head in the vicinity that can be joined, the isolated node selects the cluster with the largest T_r to join. T_r is calculated as follows:

$$T_r = \begin{cases} \frac{r_s - d(i, CH_n)}{\Delta v} & v_i > v_{CH_n} \\ \frac{r_s + d(i, CH_n)}{\Delta v} & v_i < v_{CH_n} \end{cases} \quad (9)$$

Δv is the speed difference between the isolated node and the cluster head, and $d(i, CH_n)$ is the distance between the isolated node and the cluster head.

b) Leaving a cluster:

When a cluster member moves out of the cluster's radius, it loses contact with the cluster head over the service channel. As a result, this vehicle is removed from the cluster member list maintained by the cluster head. The vehicle changes its state to standalone if there is no nearby cluster to join or there is no other nearby standalone vehicle to form a new cluster.

c) Cluster merging:

When two cluster heads are so close that at least one head satisfies the joining condition mentioned above, if their speed difference is less than a threshold v_{merge} , the cluster merging mechanism is triggered. The cluster head that has fewer members gives up its cluster-head role and becomes a cluster member in the new cluster. This cluster's other members will become isolated nodes and try to join the neighboring clusters using the joining algorithm. The nodes that cannot join the nearby clusters will start the clustering mechanism to form a new cluster.

C. Dynamic Traffic Data Collection

According to the proposed cluster mechanism, the node in a cluster transmits its traffic data to the cluster head using IEEE 802.11p DCF (Distributed Coordination Function). Each cluster head receives traffic data including each node's ID number, lane number, position, speed, vehicle type, and time stamp. Therefore, the cluster head will calculate the number of nodes in the cluster, the number of nodes in each lane, and the average speed.

When one vehicle is an isolated node (with no cluster head node within the communication range of the vehicle), it will store its information packets until a cluster head or an intersection agent is within the communication range r_s . When a cluster head is not within the communication range of an intersection agent, it will broadcast a traffic information data packet using the control channel to the neighboring cluster head that is driving in the same direction. Once the cluster head is within the communication range of an intersection control agent, the cluster head node will send a traffic information packet to the intersection control agent directly. Intersection control agents regulate the traffic signal according to the received traffic information from the V2X networks. To implement adaptive traffic signal control, it is essential to collect the lanes' queue lengths and the average waiting times, which will be used to construct the reward function of cooperative reinforcement learning with the function approximation in the following section.

III. COOPERATIVE REINFORCEMENT LEARNING WITH FUNCTION APPROXIMATION (CRLFA)

A. Distributed Multi-agent Cooperative Reinforcement Learning

Reinforcement Learning makes an optimal policy by interacting with the environment. Q-learning is a type of model-free reinforcement learning in which the agents do not need to know prior information about the environment. Q-learning

uses the Q-value [43] to represent the maximum discounted sum of long-term rewards by following the optimal policy.

$$Q(s, a) = r(s, a) + \gamma \sum_{s' \in S} p(s, s', a) \max_{a' \in A(s')} Q(s', a') \quad (10)$$

Equation (10) is called the Q-Bellman equation. The Q-value is defined as $Q(s, a)$, $s, s' \in S$ (the state set), $a \in A(s)$ (the action set), and $r(s, a) \in S \times A(s) \rightarrow \mathbb{R}$ is the reward function. $0 \leq \gamma < 1$ is a discount factor that determines the importance of future rewards. When it approaches 1, reinforcement learning will strive for a long-term high reward, which also prevents the unboundedness of the Q-values. Reinforcement learning is an incremental update stochastic algorithm that can be defined as follows:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha(t)[r(s, a) + \gamma \max_{a' \in A(s')} Q_t(s', a') - Q_t(s, a)] \quad (11)$$

This algorithm is started by initializing $Q(s, a)$ arbitrarily, and $r(s, a)$ is often initialized to 0. $\alpha(t)$ is the step-size, and it should satisfy the following conditions:

$$\sum \alpha(t) = \infty, \quad \sum \alpha^2(t) < \infty \quad (12)$$

The traffic signal control system can be taken as a multi-agent system that attempts to find the optimal policy in a dynamic environment. In this paper, agents exchange their maximal Q-values to achieve collaboration. Thus, the action selection strategy of one intersection is dependent not only on it owns reward, but also on its neighboring intersections' rewards, which is helpful for balancing the traffic among intersections. After introducing the neighboring cooperative items into Eq. (11), the cooperative reinforcement learning is updated as Eq. (13):

$$\begin{cases} \hat{Q}_{t+1}^i(s_i, a_i) = Q_t^i(s_i, a_i) + \alpha(t)[r(s_i, a_i) + \gamma \max_{a' \in A(s'_i)} Q_t^i(s'_i, a'_i) - \hat{Q}_t^i(s_i, a_i)] \\ Q_{t+1}^i(s_i, a_i) = \hat{Q}_{t+1}^i(s_i, a_i) + \sum_{j \in M} \mu(i, j) \hat{Q}_{t+1}^j(s_j, a_j) \end{cases} \quad (13)$$

where M is the set of neighbor agents of agent i . $\hat{Q}_{t+1}^i(s_i, a_i)$ is this agent's Q-value calculated by the classic Q-value update law in Eq. (11). Then, the neighboring Q-value $\hat{Q}_{t+1}^j(s_j, a_j)$ is added to agent i 's Q-value $\hat{Q}_{t+1}^i(s_i, a_i)$, which comes from the neighboring intersection agent via the communication network of intersections.

The intersection agents exchange their Q-values to realize cooperation. $\mu(i, j)$ is the weight associated with the Q-value obtained from neighboring agent j and sent to agent i . The simplest strategy for computing the weights $\mu(i, j)$ is to consider the total number of agents in the neighborhood, i.e., $\mu(i, j) = 1/M$, in which $\sum_j \mu(i, j) = 1$. More complex strategies could be introduced to take into account the fact that not all neighbors are equally affected by the actions of an agent.

B. Fast Function Approximation

In reinforcement learning, the number of state-action pairs generally grows exponentially with the number of states and actions. For example, in a small road network, let there be 8 lanes and 2 traffic lights. If each lane corresponds to 10 states and each traffic light can take 10 actions, the number of state-action pairs will be $10^8 \times 10^2 = 10^{10}$. Because of the curse of dimensionality, junction-based reinforcement learning is considered to be of limited applicability. Large sets of states cause difficulties, and the function approximation could be an effective method to solve this problem.

In this paper, a reinforcement learning with linear function approximation is introduced. The learning algorithm calculates the per-time-step costs linearly in term of the features with no restriction on the feature selection. Here the linear function approximation is defined as follows to approximate the Q-value:

$$Q_\theta(s, a) = \theta^T \varphi(s, a) \quad (14)$$

where $(s, a) \in S \times A$, and $\varphi(s, a) \in \mathbb{R}^d$ is the d -dimensional feature vector for the state-action pairs. $\theta \in \mathbb{R}^d$ is the d -dimensional parameter vector to be tuned in the step-size approximation.

The Q-learning mechanism adopts the off-policy [28] in which the policy used to select an agent's action may be different from the policy used to evaluate the agent's action. Off-policy training is useful in dealing with the exploration-exploitation trade-off. To produce the optimal parameter θ , the approximation performance is evaluated by the projected Bellman error, as defined in [39] and [40]. This error is as follows:

$$J = E[\delta_{t+1}(\theta) \varphi_t]^T E[\varphi_t \varphi_t^T] E[\delta_{t+1}(\theta) \varphi_t] \quad (15)$$

In the above formula, $\delta(\theta)$ is the TD (temporal difference) error, which is described as follows:

$$\delta_{t+1}(\theta) = r_{t+1} + \gamma \theta^T \hat{\varphi}_{t+1}(\theta) - \theta^T \varphi_t \quad (16)$$

where $\varphi_t = \varphi(s_t, a_t)$ is the feature at time t , and $\hat{\varphi}_{t+1}(\theta_t)$ is the estimation of the sub-gradient of $\bar{V}_{s(t+1)}(\theta)$, which is the expectation of the value function $V_{s(t+1)}(\theta)$ on the next state $s(t+1)$. In the proposed algorithm, $\hat{\varphi}_{t+1}(\theta)$ is estimated by $\varphi_{t+1}(s, a')$, where a' is the action to maximize $Q_\theta(s_{t+1}, \cdot)$. A detailed description can be found in [39] and [40].

It is known from Eq. (15) that the sub-differential of J to $\delta_{t+1}(\theta)$ is

$$b_{t+1}(\theta) = r_{t+1} + \gamma \hat{\varphi}_{t+1}(\theta) - \varphi_t \quad (17)$$

Thus, the sub-differential to $\frac{1}{2}J(\theta)$ could be derived as follows:

$$E[b_{t+1}(\theta) \varphi_t]^T E[\varphi_t \varphi_t^T]^{-1} E[\delta_{t+1}(\theta) \varphi_t] = -E[\delta_{t+1}(\theta) \varphi_t] + \gamma E[\hat{\varphi}_{t+1}(\theta) \varphi_t^T] \omega^*(\theta) \quad (18)$$

Here,

$$\omega^*(\theta) = E[\varphi_t \varphi_t^T]^{-1} E[\delta_{t+1}(\theta) \varphi_t] \quad (19)$$

In contrast to the traditional function approximation, which has one iteration vector θ , a new correction term of the weights $\omega_t \in \mathbb{R}^d$ is introduced. This term follows the LMS (Least

Mean Square) rule to obtain fast gradient descent. As proved in [39], this function can converge to a local optimum or equilibrium point. Then, the update rules of greedy reinforcement learning algorithms can be represented as follows:

$$\begin{cases} \omega_{t+1} = \omega_t + \beta_t [\delta_{t+1}(\theta_t) - \varphi_t^T \omega_t] \varphi_t \\ \theta_{t+1} = \theta_t + \alpha_t [\delta_{t+1}(\theta_t) \varphi_t - \gamma (\omega_t^T \varphi_t) \max \varphi_{t+1}(\theta'_t)] \end{cases} \quad (20)$$

In the above iterations, $\max \varphi_{t+1}(\theta'_t)$ means that in state s_{t+1} , the selected a'_{t+1} can maximize $Q_{\theta_t}(s_{t+1}, a'_{t+1})$. In addition, α_t and β_t are sequences of positive step-size parameters under the following assumptions:

$$\begin{aligned} \sum_{t=0}^{\infty} \alpha_t &= \sum_{t=0}^{\infty} \beta_t = +\infty \\ \sum_{t=0}^{\infty} \alpha_t^2 + \beta_t^2 &< +\infty \\ \alpha_t / \beta_t &\rightarrow 0 \end{aligned} \quad (21)$$

IV. THE ALGORITHM FOR TRAFFIC SIGNAL CONTROL

In this paper, a cooperative reinforcement learning with function approximation is designed for traffic signal control. The control agents can collect their own intersection traffic information and obtain the neighboring maximum Q-value by exchanging traffic information among themselves. Similar to [38], the state set, action set, and reward function are defined. However, different from the states and actions in [38] for the entire traffic map, the distributed method for the intersections is used in this paper. Hence, our states and actions are just for one intersection with small dimensions. In addition, the proposed algorithm uses a reduced tile coding and has a different feature selection. The formulation is as follows:

a) State set

For intersection i , according to its queue's length ($q_j^i, j = 1, 2, \dots, N_i$) and the queue's waiting time ($w_j^i, j = 1, 2, \dots, N_i$) which are obtained from V2X networks, the intersections' input state is as follows:

$$s^i = (q_1^i, q_2^i, \dots, q_{N_i}^i; w_1^i, w_2^i, \dots, w_{N_i}^i)$$

N_i is the number of lanes for one intersection. s^i is a $2 \times N_i$ -dimensional vector, and each element represents each piece of lane traffic information, namely, the queue length or the queue's average waiting time.

b) Action set

As shown in Fig. 3, the classical eight non-conflicting traffic signal phases are adopted in this paper. The initial action set is $A = \{(1, 2), (1, 5), (2, 6), (3, 4), (3, 7), (4, 8), (7, 8)\}$. Based on the learning strategies, the intersection agent selects a pair of phases from eight non-conflicting phases as its taken action. According to its own and neighboring intersections' traffic information, the control agent chooses a phase sequence adaptively and determines every phase time. If there is a new action, the agent records it and adds it to the historical action set.

c) Reward function

To reduce the average waiting time of the vehicles (its rational calculation is given in [35]), the lane-dominant set M_N is defined. The reward function for each intersection









1	2	3	4
			
5	6	7	8
			

Fig. 3. Classical eight non-conflicting traffic signal phases.

consists of two parts to ensure fairness, i.e., to ensure that no vehicle will wait for too long at a red light. The first part is the sum of the queue length, and the second part is the sum of the queue's waiting time. To maximize the reward, we use the negative value of the queue length and the waiting time, so the reward function is defined by

$$r_t^i(s_t^i, a_t^i) = -\eta_1 \sum_{n \in N_M} (q_n^i(t) + \xi w_n(t)) - \eta_2 \sum_{n \notin N_M} (q_n^i(t) + \xi w_n(t)) \quad (22)$$

where ξ is the coefficient from the queue's waiting time to the queue's length, and η_i is the weight factor. The weight factor satisfies

$$\eta_1 + \eta_2 = 1, \eta_1 > \eta_2 > 0$$

d) Feature selection

To balance the approximation performance and storage cost, the proposed reinforcement learning algorithm uses a reduced tile coding method to determine the feature of the state. The feature is related to both the current queue length of the state and the action adopted by the control agent. The maximum queue length, that is, the range of the queue length for every lane, is divided into tiles by the fixed tile width W_{tile} . When the current queue length is located in one tile, the feature value is 1; otherwise, this value is 0.

$$\varphi_{k,j} = \begin{cases} 1 & q_k \in tile_{k,j} \\ 0 & q_k \notin tile_{k,j} \end{cases}$$

For example, the maximum queue length $L_{max,k}$ for lane k is 2000 meters and the tile width W_{tile} is 400 meters. If the current queue length q_k is 930, the feature value for the tile corresponding to the interval $[800, 1200]$ is $\varphi_{k,3} = 1$, and the other feature values for lane k are zeros. As a result, for the queue length for lane k (which is one dimension of the state), though the number of tiles is $T_k = \lceil L_{max,k}/W_{tile} \rceil$, only one tile is active. For one action, the feature is the action-index listed in Fig. 3. Thus, the total number of tiles is $1 + \sum_{k=1}^{N_l} T_k$, where N_l is the number of lanes. Comparatively, if the classical tile coding is adopted [28], the total number of tiles is $1 + N_{tiling} \prod_{k=1}^{N_l} T_k$, where N_{tiling} is the number of tilings. Consequently, the storage requirement is enormous for classical tile coding when the dimension is high.

For example, for an intersection with 12 lanes, if the number of tiles for one lane is just 5 and N_{tiling} is 10, the number

of tiles for the classical tile coding is $1 + 10 \times 5^{12} = 2,441,406,251$. This situation would require a large storage space for saving the approximation parameter θ . In contrast, in the reduced tile coding method proposed here, the number of tiles is $1 + 5 \times 12 = 61$, which does not use the parameter N_{tiling} .

The tile coding features denote coarse information of the waiting queue rather than the exact queue length in a lane, which is generally difficult to obtain in practice. However, the coarse information can be estimated by some graded thresholds. Furthermore, the graded thresholds to divide the queue can be optimized to adapt to various traffic conditions [44].

Algorithm I lists the steps for the learning and cooperative procedures designed in the proposed CRLFA (Cooperative Reinforcement Learning with Function Approximation) algorithm for traffic signal control by pseudo code in TABLE I.

Remark: Algorithm I integrates the cooperative learning and fast function approximation, which introduces nontrivial difficulties if one seeks to analyze the convergence performance. For the proposed cooperative learning mechanism given by Eq. (13), it can be transferred to a special cooperative stochastic game process [45] in which the game only occurs among neighboring agents. The algorithm's convergence performance could be analyzed by extending the stochastic game to design a new equilibrium conception. Further discussion is beyond the scope of this paper. However, when the cooperative game interweaves with the fast function approximation, it is not immediately possible to obtain insight into the convergence analysis. Fortunately, for traffic control problems, the adaptability is a preferable goal to be achieved due to the dynamics of traffic flows. Nevertheless, in the following simulation, the convergence performance is presented by numerical experiments under different traffic scales.

V. SIMULATION

In this section, simulation experiments are conducted to test the performance of the proposed algorithms, including the dynamic clustering algorithm and the cooperative RL-based signal control algorithm. First, the simulation scenario and parameters are presented. Then, the cluster stability and communication performance are evaluated. Subsequently, the joint V2X networks and traffic control simulations are implemented to validate the proposed algorithm in term of traffic performance.

A. Simulation Scenario and Parameter Setting

1) *Simulation tools and scenario:* Because the simulation needs to validate the communication performance as well as the traffic performance, the simulation environment includes a wireless network simulator and a traffic simulator. In the simulation, the two simulators should be integrated in their execution so that they can synchronize each other and exchange information. To realize this integration, NS3 [46] is used as the network simulator and SUMO [47] is used as the traffic simulator. NS3 and SUMO are connected by the TRACI interface [48].

TABLE I
ALGORITHM I: CRLFA FOR TRAFFIC SIGNAL CONTROL

Algorithm I: CRLFA for traffic signal control

Initialize $\gamma, a, \beta, \delta_0$;
Initialize $\theta_0 \in R^d, \omega_0 \in R^d, \varphi_0 \in R^d$;
 $t = 0, Q_c = 0$;
For each iteration step $t + 1$, do:
 For each agent i , do:
 a. Observe s_{t+1}^i, a_{t+1}^i ;
 b. Observe the local reward:

$$r_{t+1}^i(s_{t+1}^i, a_{t+1}^i) = -\eta_1 \sum_{n \in N_M} (q_n^i(t+1) + \xi w_n(t+1))$$

$$-\eta_2 \sum_{n \notin N_M} (q_n^i(t+1) + \xi w_n(t+1))$$
 ;
 c. Seek the optimal strategy:
 For all $a' \in A'(s_{t+1}^i)$:

$$\hat{a} = \max_{a'} Q_\theta(s_{t+1}^i, a') = \max_{a'} \theta_t^{iT} \varphi_t^i(s_{t+1}^i, a')$$
 Let $\hat{\varphi}_{t+1}^i(\theta_t^i) = \varphi^i(s_{t+1}^i, \hat{a})$;
 d. Update δ^i : $\delta_{t+1}^i(\theta_t^i) = r_{t+1}^i + \gamma[\theta_t^{iT} \hat{\varphi}_{t+1}^i(\theta_t^i) + Q_c] - \theta_t^{iT} \varphi_t^i$;
 e. Update ω^i : $\omega_{t+1}^i = \omega_t^i + \beta[\delta_{t+1}^i(\theta_t^i) - \varphi_t^{iT} \omega_t^i] \varphi_t^i$;
 f. Update θ^i : $\theta_{t+1}^i = \theta_t^i + \alpha[\delta_{t+1}^i(\theta_t^i) \varphi_t^i - \gamma(\omega_t^{iT} \varphi_t^i) \hat{\varphi}_{t+1}^i(\theta_t^i)]$;
 g. Broadcast the local Q-value $\hat{Q}_\theta^i(s_{t+1}^i, a_{t+1}^i) = \theta_{t+1}^{iT} \varphi^i(s_{t+1}^i, a_{t+1}^i)$
to all of its neighboring agents j ;
 h. For all neighboring agents j , receive the neighboring local Q-value:

$$\hat{Q}_\theta^j(s_{t+1}^j, a_{t+1}^j)$$
 from agent j ,
 calculate the cooperative Q-value:

$$Q_c = \sum_{j \in M} \mu(i, j) \hat{Q}_{t+1}^j(s_j, a_j)$$
 ;
 i. Take action $a = \hat{a}$ with ε or take any action $a \in A'(s_{t+1}^i)/\hat{a}$ with probability $1 - \varepsilon$;
 Endfor
 $t = t + 1$;
Endfor

NS3 is a free discrete-event network simulator that develops a preferred open simulation environment. The simulator's core supports both IP-based and non-IP-based networks, as it is abundant in wireless/IP simulation models and supports a variety of static or dynamic routing protocols. The SUMO simulator is a C++-based, open-source, highly portable, microscopic road-traffic-simulation platform designed to handle large road networks. In SUMO, each vehicle is explicitly modeled as one entity, that has its own route and moves individually through the road network.

To connect the two simulators, the TRACI interface is utilized to produce a closed-loop simulation environment. TRACI is a client/server architecture in which SUMO is configured as a server. A client application, such as NS3 accesses SUMO via the TCP socket. The client application sends TRACI commands to SUMO to control the simulation run and to influence each vehicle's movement or to request environmental details, such as the vehicles' speed or the road's width and length. SUMO responds to the client application with a status response to each command. Both the commands to SUMO and the traffic traces from SUMO are transported using TCP. By using this method, TRACI can be used to connect the NS3 instance to send a series of commands to the

traffic signal controller, which will influence the intersections and vehicles' actions in SUMO.

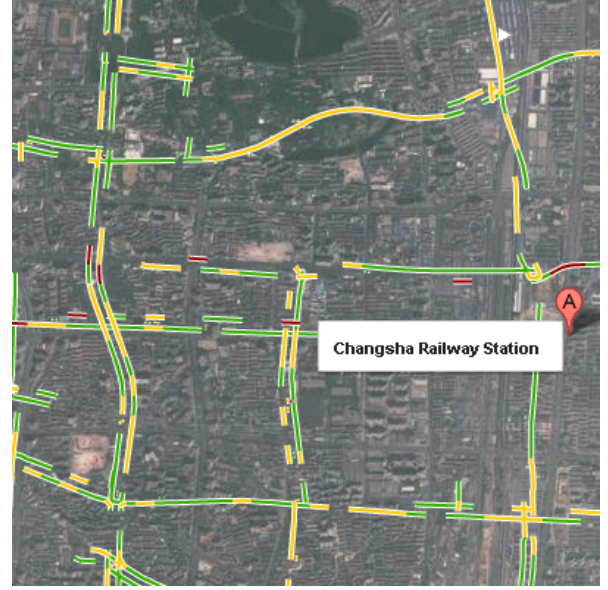


Fig. 4. A partial map of Changsha near the railway station. This map came from Google Earth.

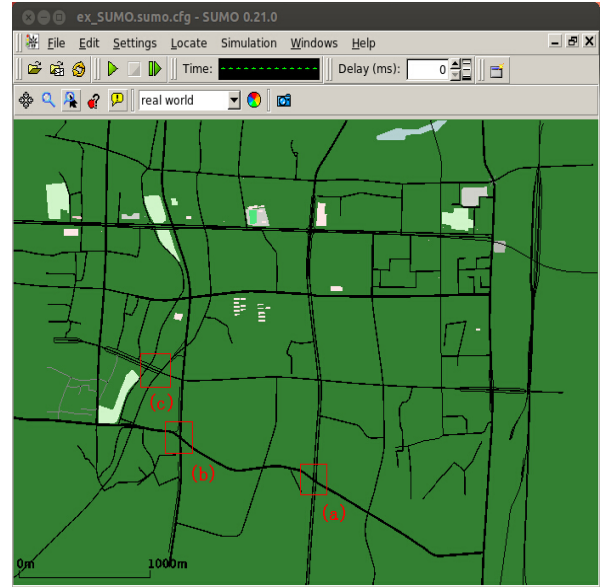


Fig. 5. The road network corresponding to the railway region of Changsha City, which has been imported into SUMO.

Every vehicle and every intersection in SUMO would be mapped to a VANET node in NS3, including the new vehicles that are newly injected into SUMO. Intersections correspond to fixed nodes. SUMO executes in discrete time steps, and the traffic mobility generated by SUMO is parsed by a manager module and communicated to the corresponding node in NS3. The VANET nodes in NS3 execute the data collection through a dynamic clustering structure. Then, the fixed nodes corresponding to intersections will generate the traffic information and produce adaptive traffic signal control commands using a

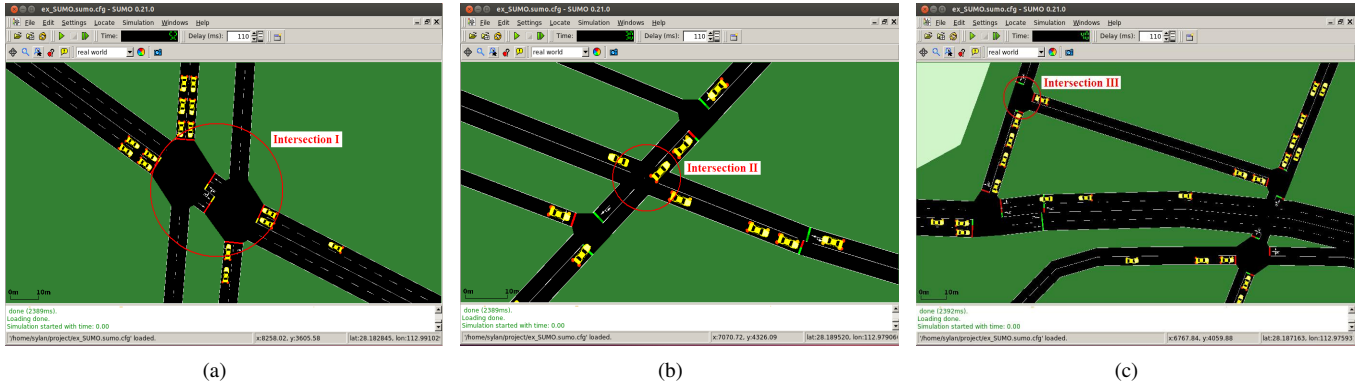


Fig. 6. Three sub-scenarios in the railway station region of Changsha City for parameter tuning in the simulation.

specific control algorithm. The control commands will be sent back to the traffic simulator SUMO in the TRACI command format. During the simulation, for each control interval, the manager module triggers the execution of one time step of the road traffic simulator SUMO, updating and sending the position to the network simulator NS3. NS3 simulates the data collection process in VANET and sends back the adaptive traffic control signal to SUMO.

To effectively test the performance of our proposed algorithm, a real scenario is taken into consideration. Fig. 4 is a part of a map of Changsha (the capital city of Hunan province, China) near the railway station. This area is approximately 4,000 m×4,000 m, and the simulation scenario is built at the proportion of 1:40,000; it consists of 78 road segments and 96 intersections, and the arterial roads in the scenario are bidirectional and comprise four lanes. In the real traffic network, not all intersections have four directions. Fig. 5 is the simulation scenario imported into SUMO; it is a typical high-dimension scene. In the distributed reinforcement learning algorithm, every agent must deal with the lane information of its own intersection as the input state, which generally has less than 20 dimensions. The designed reduced tile coding decreases the dimensions further.

2) *The simulation parameter setting:* For the simulation scenario, according to the real conditions of the urban traffic environment, the vehicle speed v_{avr} can be set in the range of 0 ~ 60 km/h. For each vehicle, its instant speed will fluctuate by $\pm 50\% \times v_{avr}$. For example, if one vehicle's average speed is set to 20 km/h, the instant speed will be in the range 10 km/h ~ 30 km/h, that is, the speed-varying threshold Δv_{th} is 10 km/h. In sub-section V-B, which contains the V2X network simulations and analysis, the vehicle arrival rate is simulated to be linearly increasing from 0.01 (veh/second) to 0.05 (veh/second). In sub-section V-C, which contains the simulations integrating VANET and traffic signal control, the vehicle arrival rate is coming from the traffic police detachment, Yuhua District, Changsha City, which has cooperated with the authors with research funds.

The wireless network simulator NS3 implements IEEE 802.11p. The mobile nodes transmit messages including the following traffic information: the vehicle ID, vehicle position, vehicle speed, vehicle destination, and time stamp. The size of the packets to be transmitted is 1 kB. The transmission

rate was set to 10 packets/second. The settings of the specific network parameters are summarized in TABLE II.

TABLE II
VANET SIMULATION PARAMETERS

Parameter	Value
Service channel transmission range d_f	150 ~ 300 m
Control channel transmission range d_c	800 ~ 1000 m
Minimum value of competition window CW_{min}	15 idle slots
Maximum value of competition window CW_{max}	1023 idle slots
Data transmission rate	10 packet/s
Periodic data transmission interval	100 ms
Packet size	1000 bytes
Predicting time	30 s
MAC protocol	IEEE 802.11p

The parameters of the proposed reinforcement learning algorithm include the tile width W_{tile} , the updating step-size α , β , and the discount factor γ . To obtain the optimal parameters, the local simulation is run 200 times on each of the three sub-scenarios in the railway station region of Changsha City. The three sub-scenarios are shown in Fig. 6 (a), (b), and (c). The traffic flow is as described previously. By comparing the statistics of the average intersection waiting time in these sub-scenarios, the most optimal parameters can be selected by numerical simulation.

In the simulation, the tile width W_{tile} has a more important effect than the other parameters. Thus, the simulation mainly focuses on the determination of W_{tile} . When decreasing the tile width W_{tile} , the approximation ability of the linear function in Eq. (14) will be improved, and the proposed algorithm can assume fine control to decrease the queue length and average waiting time, as in the representation of Fig. 7. However, a tile width W_{tile} that is too small will increase the number of tiles; consequently, the storage cost will increase. In Fig. 7, it is shown that when the tile width is less than 15, the performance improvement is not obvious for the three sub-scenarios. Therefore, the tile width is set to 15 in the following simulations. After tuning the parameters in the simulations and referring to other works (such as [35] [39] [40]), the updating step-size α is set to 0.03, β is set to 0.125, the discount factor γ is set to 0.9, and ε is set to 0.1. The initial value of θ is set

to 1, and the initial value of ω is set to 0.

Fig. 8 illustrates the averages of θ for Intersection I, Intersection II, and Intersection III in the sub-scenarios (a), (b) and (c), respectively. The three intersections have different lane scales. From Fig. 8, the three averages of θ can converge within approximately 2000 iterations even under different traffic scales.

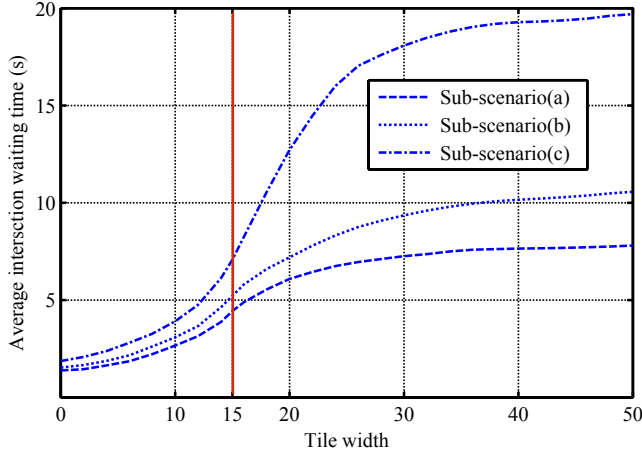


Fig. 7. The statistics of the average intersection waiting times for different tile widths after running 200 simulations in the three sub-scenarios.

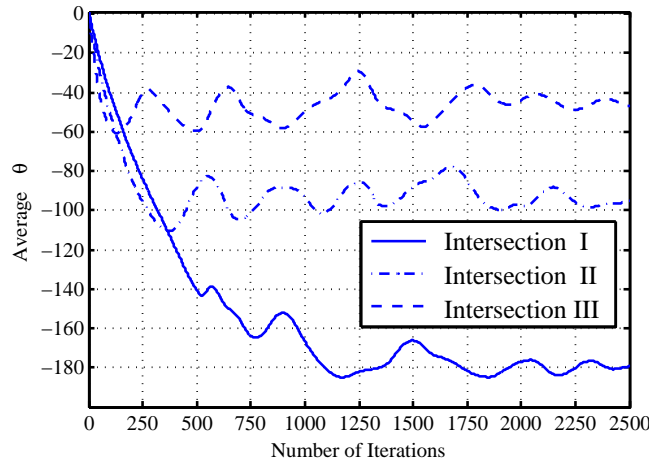


Fig. 8. The convergence of the average θ for three intersections with different lane scales when the tile width = 15.

B. V2X Network Simulation and Analysis

To evaluate the performance of the proposed DC-TDCA algorithm, this section analyzes the network topology stability and communication performance.

1) The network topology stability: the average lifetime of clusters is used to evaluate the performance in terms of the network topology stability.

2) Communication performance: communication analysis is performed on the following three factors:

a) Packet delivery ratio, which is defined as the ratio of the number of packets received and the number of total sent packets. The factor denotes the success ratio of the data packet transmission in VANET.

b) Network communication cost, which is defined as the number of repetitive packets received by the destination node in the process of data transmission.

c) Average delay, which is defined as the average delay of all received packets.

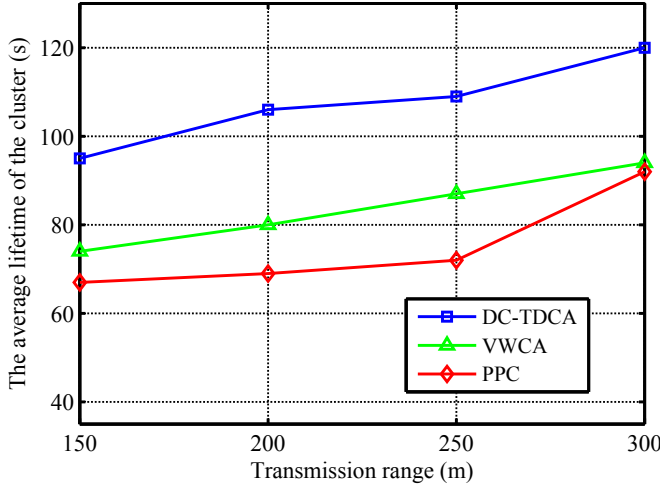
The V2X network performance is analyzed by NS3 to evaluate the proposed algorithm's performance in terms of the dynamic clustering and data collection algorithm.

1) *The network topology stability*: To reduce the networking cost, the cluster structure of V2X networks should be comparatively stable under the node's moving condition, that is, the cluster configuration should not change frequently with the node's movement. In highly varying VANET, vehicle nodes join and leave clusters dynamically along roads. The average lifetime of clusters is different based on the clustering algorithm. For comparison, two other clustering algorithms are also implemented: PPC [16], which uses the node position as the clustering metric, and VWCA [21], which uses a composite metric to integrate the trustless value, neighbor number and distance. The performance of the three clustering algorithms is analyzed in the simulation. The simulation result is shown in Fig. 9.

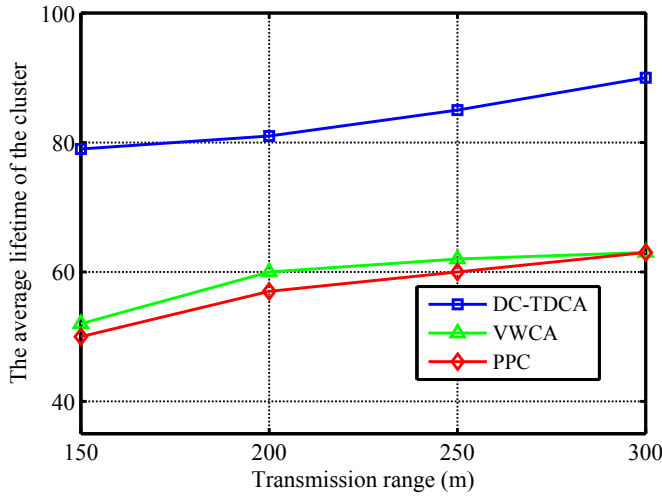
In Fig. 9, with the increase of the vehicle speed, the clusters change more frequently. The lifetime of clusters is reduced for all three algorithms. Compared with VMCA and PPC, DC-TDCA increases the lifetime of clusters by 20-40% on average, as in the VMCA and PPC algorithms, the condition to combine the clusters is easier to meet. The proposed DC-TDCA algorithm needs to meet the following moving condition: the average velocity difference of two cluster heads must be less than Δv_{th} . Thus, due to the movement of the vehicle nodes, the cluster nodes and cluster heads will be separated quickly without meeting the condition of the threshold Δv_{th} , unlike in the VMCA and PPC algorithms, especially in the clusters with fewer nodes. Moreover, the lane weights, the destination and the number of neighboring nodes are taken into consideration to improve the average lifetime of the clusters when the similarity function in the affinity propagation algorithm is redefined in the paper.

2) *Communication performance*: In VANET, the traffic flow density will significantly affect the connectivity of the network. In the case of different traffic flow densities, the proposed algorithm is compared with the other two algorithms (VMCA and PPC) in terms of the communication performance. The communication simulation results are shown in Fig. 10~12.

Fig. 10 shows the packet delivery ratio under different traffic flow densities for DC-TDCA, VMCA, and PPC. From Fig. 10, when the traffic density increases, the reliable data transmission path and the packet delivery ratio increase in VANET. However, with an increase in the number of packets that the vehicles request to send in VANET, the data flow increases. As a result, the node processing burden will increase and the efficiency will decrease. For VMCA and PPC, with an



(a) $v=20\text{km/h}$, $\Delta v_{th}=10\text{km/h}$



(b) $v=40\text{km/h}$, $\Delta v_{th}=10\text{km/h}$

Fig. 9. The comparison of the average lifetimes for different clustering algorithms under different transmission ranges.

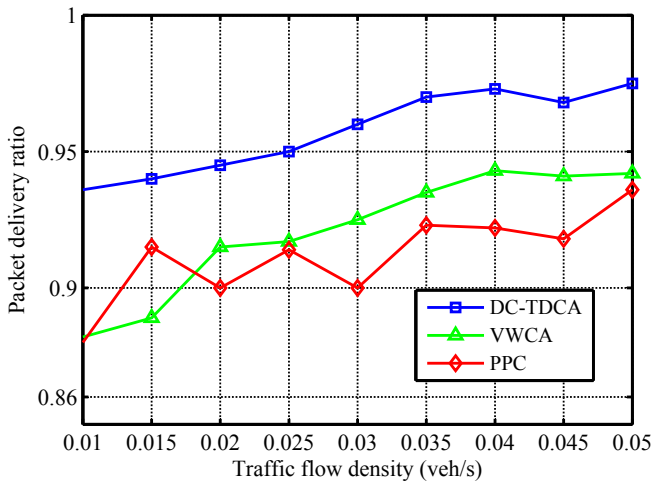


Fig. 10. The comparison of the packet deliveries for different clustering algorithms under different traffic flow densities.

increase of the vehicle density, the data transmission conflict will increase dramatically; hence, the growth of the packet delivery rate will decline. However, for the proposed DC-TDCA algorithm, VANET is divided into comparatively stable clusters. Our proposed algorithm decreases the number of packets and effectively reduces the network congestion. Furthermore, the data transmission conflict has been reduced by using DCF model mechanism in DC-TDCA, and the success rate of the data delivery has also been improved.

Fig. 11 shows that the data communication cost changes with different traffic flow densities for DC-TDCA, VMCA, and PPC. It is shown that the communication cost of the proposed DC-TDCA algorithm is far less than the communication costs of VMCA and PPC. Compared with PPC, the communication cost of the proposed algorithm decreases by 60% when the data transmission rate reaches 0.05veh/s because DC-TDCA can produce clusters with proper size. In this way, the communication cost can be significantly decreased when the traffic flow density increases. Similarly, with the guarantee of network topology stability, the dynamics of the cluster structure drops down. Therefore, the communication cost declines for data collection both inside the cluster and outside the cluster, which is advantageous for collecting traffic information, such as the traffic flow densities.

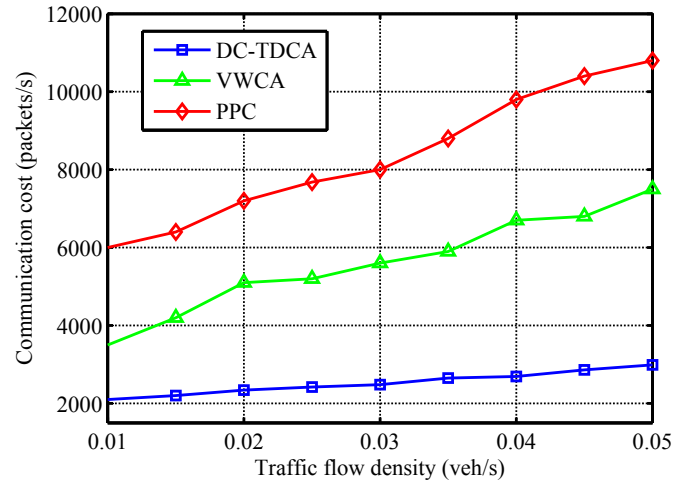


Fig. 11. The comparison of the communication costs for different clustering algorithms under different traffic flow densities.

The average delays under different traffic flow densities are shown in Fig. 12. When the traffic flow density is low, the vehicle node needs to store the packets until it finds the next-hop node. Therefore, there is a transmission delay under a lower traffic flow density for the three algorithms. With the increase of the traffic flow density, the collision of the link layer will increase correspondingly, resulting in an increased time delay. As shown in Fig. 12, when the vehicle density reaches 0.02veh/s, the time delays of VMCA and PPC present an increasing tendency, while that of the proposed DC-TDCA is relatively stable. When the vehicle density reaches 0.05veh/s, the time delay of the DC-TDCA algorithm is reduced nearly 30% and 60% compared to VMCA and PPC, respectively.

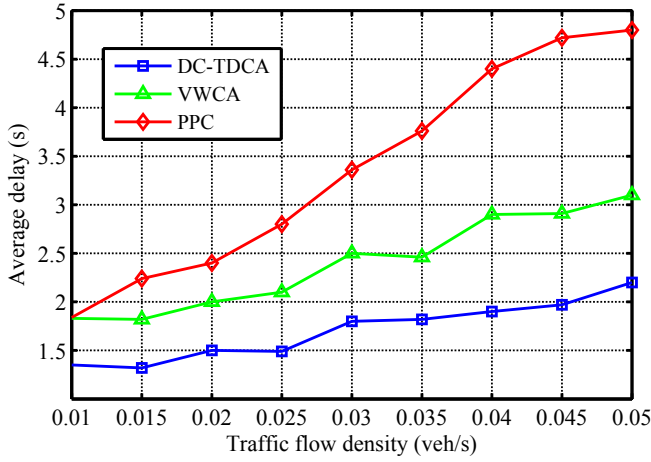


Fig. 12. The comparison of the average delay for different clustering algorithms under different traffic flow densities.

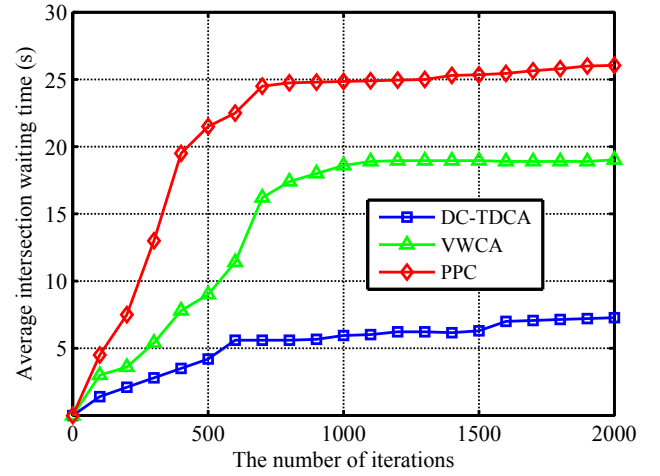
C. Simulation Integrating VANET and traffic signal control

1) *Traffic performance analysis for different clustering algorithms:* As described in the above experiments, some simulation experiments have been implemented to compare the performance of the communications with different clustering algorithms. For the transportation applications of VANET, it is insufficient to improve the communication performance. The more important issue is to improve the traffic performance. To make a convenient comparison, simulation experiments are conducted with the same proposed traffic signal control based on the distributed cooperative reinforcement learning algorithm.

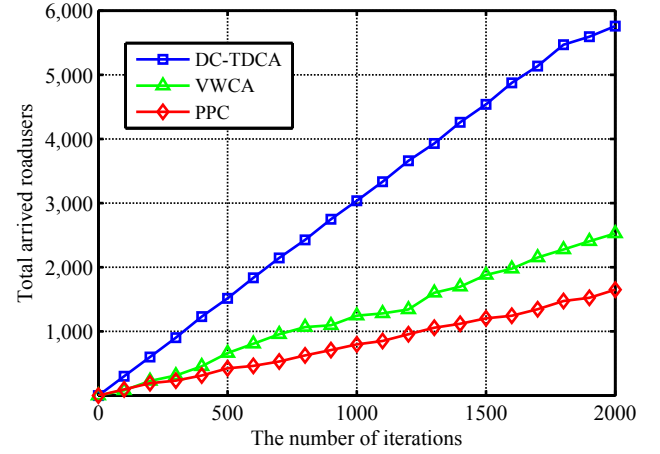
To evaluate the traffic performance for different clustering algorithms, three different performance indices are considered: 1) the average intersection waiting time (AIWT): the average waiting time of all of the vehicles that have been waiting for a green light at all intersections; 2) the total arrived road users (TARs): the total vehicles that have reached their destination in a given time step; and 3) the total queue length (TQL): the number of vehicles that have been generated but are still waiting to enter the network because their outbound lanes are still full.

Fig. 13 illustrates the traffic performance for different clustering algorithms. As shown in the figure, the DC-TDCA algorithm outperforms the other two algorithms. DC-TDCA has a more impressive communication performance, as it reduces the packet time delay and drop ratio. The intersection agent can become more real-time and utilize more trustful traffic information based on VANET, and hence, it can take more proper action. The results also show that the improvement of the communications for VANET enables the traffic performance of intelligent transportation systems.

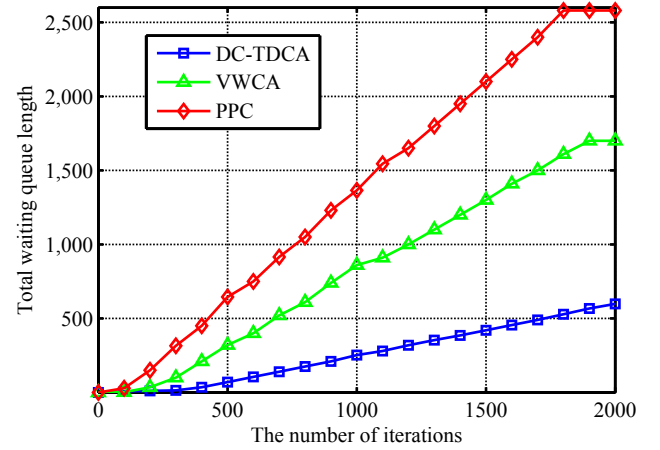
2) *Traffic performance analysis for different traffic signal control schemes:* In this section, simulation experiments are implemented to test the performance of the cooperative RL-based intelligent control algorithm that integrates the proposed clustering algorithm. The simulation results show that it is beneficial to introduce the cooperative mechanism and function



(a) AIWT comparison for different clustering algorithms under the proposed control scheme.



(b) TARs comparison for different clustering algorithms under the proposed control scheme.



(c) TWQ comparison for different clustering algorithms under the proposed control scheme.

Fig. 13. Traffic performance comparison for the clustering algorithms in the simulation scenario.

approximation for RL-based intelligent traffic signal control. For comparison, three other algorithms implemented are as follows:

(1) Fixed-timing TLC. This algorithm periodically cycles through preset traffic signal configurations and does not consider the traffic load on the lanes of the road network. To be close to the traffic control in the region, the green time for the arterial road is set to 70 s, and 40 s is selected for the branch road direction.

(2) LQF. In the longest-queue-first (LQF) algorithm, the controller will switch traffic lights so that the traffic light with the longest queue of waiting road users will be set to green first.

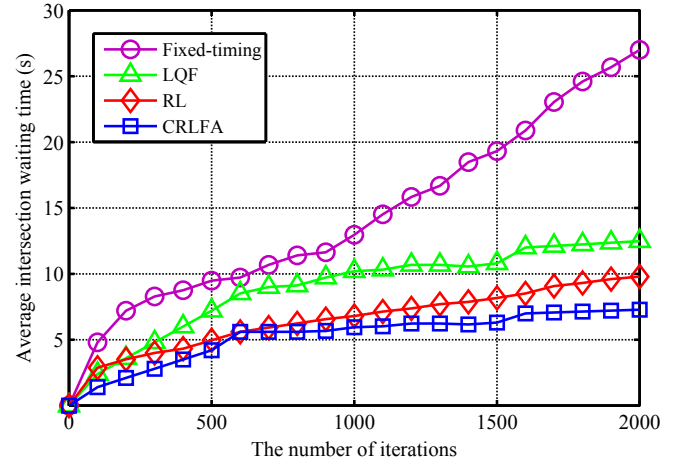
(3) RL. In this algorithm, agents control the traffic light by using traditional reinforcement learning, which is the same as the method presented in Section 6.5, reference [28]. One intersection agent uses its own lanes' queue lengths and average waiting times as the states, without the cooperation of the neighboring intersection agents. In addition, the reinforcement learning does not use the function approximation.

For the sake of comparison, the three signal control schemes adopt our proposed clustering algorithm to obtain traffic information. The traffic performance indices are also the average intersection waiting time (AIWT), total arrived road users (TARs), and total queue length (TQL). The results of the scenario are shown in Fig. 14.

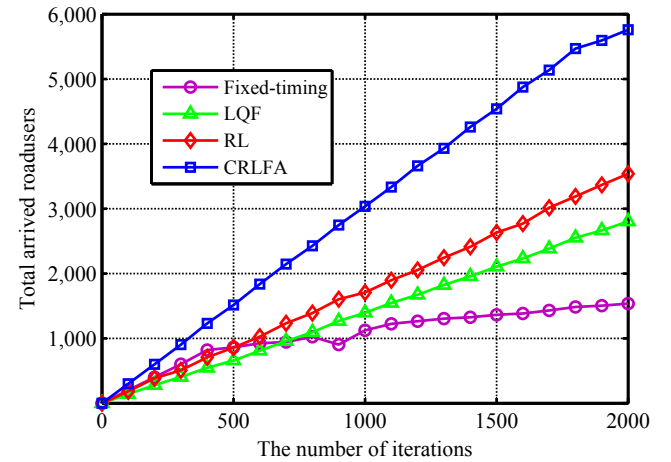
As shown in Fig. 14, the traffic performance of CRLFA is significantly better than that of the other traffic signal control algorithms. After running for nearly 1000 iterations, the other three algorithms have an obvious performance reduction. Thus, some traffic jams appear. The simulation results illustrate that the CRLFA algorithm has the ability to resolve urban traffic congestion effectively, as it has a smaller average intersection waiting time, a smaller total waiting queue length, and a larger total number of arrived road users. From the simulation, it is reasonable to deduce that the cooperation among the neighboring intersections is very important for traffic control, and the fast function approximation also contributes to the improved performance.

VI. CONCLUSIONS AND FUTURE WORK

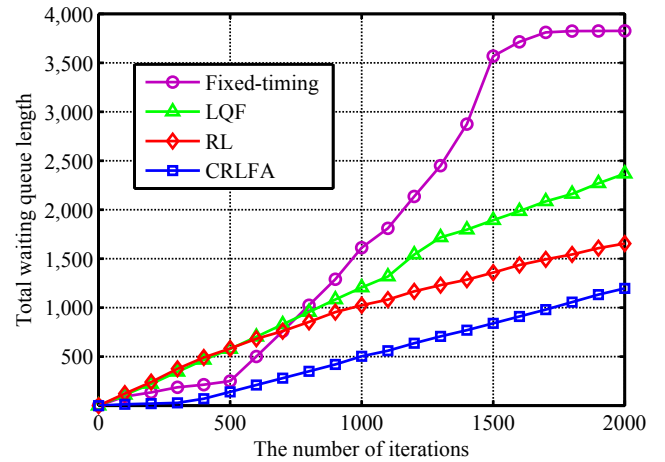
This paper presents a reinforcement learning traffic control scheme that integrates a stable dynamic clustering algorithm. This algorithm includes explicit cooperative behaviors between neighboring agents based on V2X networks' data collection. To improve the cluster stability in V2X networks, a dynamic clustering algorithm based on enhanced affinity propagation is proposed by considering the lane and destination information. For intersection control agents, the cooperation among the intersections is achieved by a cooperative reinforcement learning algorithm. To address the curse of dimensionality effectively, a fast gradient-descent function approximation method is applied to seek the optimal policy. Empirical results on the traffic networks demonstrate that the proposed control scheme integrating the stable clustering algorithm outperforms the traditional adaptive signal control method. Furthermore, the results provide a new understanding that a cooperation



(a) AIWT comparison for different control schemes under the proposed clustering algorithm.



(b) TARs comparison for different control schemes under the proposed clustering algorithm.



(c) TWQ comparison for different control schemes under the proposed clustering algorithm.

Fig. 14. Traffic performance comparison for the signal control schemes in the simulation scenario.

mechanism with the function approximation is promising for reinforcement learning-based controls in large-scale urban regions.

In the proposed intelligent traffic control algorithm, the control agent action is based on its own information and the neighbor agents' information. This scheme does not require the manipulation of a centralized urban traffic management center, and avoids transmission congestion among intersections and computing overloads in the management center, which is the advantage of distributed cooperative control. In the super metropolis, it is necessary to partition the urban area into districts to control, which produces the district traffic balance demand. Using distributed cooperative control completely could cause delays in realizing the traffic balance among districts. Therefore, our future work will be focused on introducing a hierarchical structure to incorporate the centralized control and the distributed cooperative control. The hierarchical framework utilizes a centralized district controller to manage the intersection controllers in one district. One of the key issues is how to select and merge the characteristics of the intersection controllers to realize the trade-off between the communication cost and the control performance. It will also be critical to design the coordination strategy among the district controllers in our future work.

REFERENCES

- [1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and W. Yibing. "Review of road traffic control strategies." *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043-2067, Dec. 2003.
- [2] D. I. Robertson. "TRANSYT: Traffic network study tool." *Proceedings of 4th International Symposium on the Theory of Traffic Flow*, Karlsruhe, Germany, 1968.
- [3] D. I. Robertson and R. D. Bretherton. "Optimizing networks of traffic signals in real time-the SCOOT method." *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 11-15, Feb. 1991.
- [4] A. G. Sims and K. W. Dobinson. "The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits." *IEEE Transactions on Vehicular Technology*, vol. 29, no. 12, pp. 130-137, May. 1980.
- [5] C. K. Keong. "The GLIDE system—Singapore's urban traffic control system." *Transport Reviews*, vol. 13, no. 4, pp. 295-305, 1993.
- [6] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode. "Adaptive traffic lights using car-to-car communication." *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, Dublin, pp. 21-25, Apr. 2007.
- [7] Y. Toor, P. Muhlethaler and A. Laouiti. "Vehicle Ad Hoc networks: applications and related technical issues." *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 74-88, Sep. 2008.
- [8] G. Karagiannis, O. Altintas and E. Ekici. "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions." *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 584-616, Jul. 2011.
- [9] K. Pandit, D. Ghosal, H. M. Zhang and Chen-Nee Chuah. "Adaptive traffic signal control with Vehicular Ad hoc Networks (VANET)." *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1459-1471, Jan. 2013.
- [10] D. Jiang and L. Delgrossi. "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments." *Proceedings of IEEE Conference on Vehicular Technology*, Singapore, pp. 2036-2040, May. 2008.
- [11] N. Maslekar, M. Boussedjra, J. Mouzna and H. Labiod. "VANET based adaptive traffic signal control." *Proceedings of IEEE Conference on Vehicular Technology*, Budapest, pp. 15-18, May. 2011.
- [12] I. Salhi, M. Cherif and S. Senouci. "A new architecture for data collection in vehicular networks." *Proceedings of IEEE International Conference on Communications*, Dresden, Germany, pp. 1-6, Jun. 2009.
- [13] M. Jerbi, S. M. Senouci and T. Rasheed. "An infrastructure-free traffic information system for vehicular networks." *Proceedings of IEEE Conference on Vehicular Technology*, Dublin, Ireland, pp. 2086-2090, Oct. 2007.
- [14] O. Kayis and T. Acarman. "Clustering formation for inter-vehicle communication." *Proceedings of IEEE Conference on Intelligent Transportation Systems*, Seattle, Washington, USA, pp. 636-641, Oct. 2007.
- [15] N. Maslekar, M. Boussedjra and J. Mouzna. "Direction based clustering algorithm for data dissemination in vehicular networks." *Proceedings of IEEE Conference on Vehicular Networking*, Tokyo, Japan, pp. 1-6, Oct. 2009.
- [16] Z. Wang, L. Liu and M. C. Zhou. "A position-based clustering technique for ad hoc intervehicle communication." *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, pp. 201-208, Mar. 2008.
- [17] S. S. Wang and Y. S. Lin. "Performance evaluation of passive clustering based techniques for inter-vehicle communications." *Proceedings of IEEE Annual Wireless and Optical Communications Conference*, Shanghai, China, pp. 1-5, 2010.
- [18] R. T. Goonewardene, F. H. Ali and E. Stipidis. "Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks." *IET Intelligent Transport Systems*, vol. 3, no. 2, pp. 148-158, 2009.
- [19] Z. Zhang, A. Boukerche and R. Pazzi. "A novel multi-hop clustering scheme for vehicular ad-hoc networks." *Proceedings of ACM International Symposium on Mobility Management and Wireless Access (ACM)*, Miami, FL, USA, pp. 19-26, Oct. 2011.
- [20] S. Kuklinski and G. Wolny. "Density based clustering algorithm for VANET." *Proceedings of IEEE International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops*, Washington, DC, pp. 1-6, Apr. 2009.
- [21] A. Daeinabi, A. G. Pour Rahbar and A. Khademzadeh. "VWCA: An efficient clustering algorithm in vehicular ad hoc networks." *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 207-222, Jan. 2011.
- [22] G. Wolny. "Modified DMCA clustering algorithm for VANET." *Proceedings of IEEE International Conference on Systems and Networks Communications*, Sliema, Malta, pp. 268-273, Oct. 2008.
- [23] B. Hassanabadi, Shea C and Zhang L. "Clustering in vehicular ad hoc networks using affinity propagation." *Ad Hoc Networks*, vol. 13, pp 535-548, Feb. 2014.
- [24] R. Hoar, J. Penner and C. Jacob. "Evolutionary swarm traffic: if ant roads had traffic lights." *Proceedings of the Congress on Evolutionary Computation*, Honolulu, HI, pp. 1910-1915, 2002.
- [25] J. J. Sanchez, M. Galan and E. Rubio. "Genetic algorithms and cellular automata: a new architecture for traffic light cycles optimization." *Proceedings of the Congress on Evolutionary Computation*, vol. 2, pp. 1668-1674, Jun. 2004.
- [26] S. Mikami and Y. Kakazu. "Genetic reinforcement learning for cooperative traffic signal control. Evolutionary Computation." *Proceedings of the First IEEE World Congress on Computational Intelligence*. Orlando, FL, pp. 223-228, Jun. 1994.
- [27] T. K. Ho. "Fuzzy logic traffic control at a road junction with time-varying flow rates." *Electronics Letters*, vol. 32, no. 17, pp. 1625-1626, Aug. 1996.
- [28] R. S. Sutton and A. G. Barto. "Reinforcement learning: an introduction." *Cambridge University Press*, vol. 1, 1998.
- [29] M. A. Wiering. "Multi-agent reinforcement learning for traffic light control." *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, Stanford University, Stanford, CA, USA, Jul. 2000.
- [30] M. A. Khamis, W. Gomaa, A. El-Mahdy and A. Shoukry. "Adaptive traffic control system based on Bayesian probability interpretation." *Electronics, Communications and Computers (JEC-ECC)*, 2012 *Japan-Egypt Conference on*, Alexandria, pp. 151-156, Mar. 2012.
- [31] M. A. Khamis, W. Gomaa, and Hisham El-Shishiny. "Multi-objective traffic light control system based on Bayesian probability interpretation." *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, pp. 995-1000, Sep. 2012.
- [32] S. Richter, D. Aberdeen and J. Yu. "Natural actor-critic for road traffic optimization." *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, pp. 1169-1176, 2007.
- [33] M. A. Khamis and W. Gomaa. "Enhanced multiagent multi-objective reinforcement learning for urban traffic light control." *Machine Learning and Applications (ICMLA)*, 2012 *11th International Conference on*, Boca Raton, FL, pp. 586-591, Dec. 2012.
- [34] A. Salkham, R. Cunningham, A. Garg, and V. Cahill. "A collaborative reinforcement learning approach to urban traffic control optimization." *Proceedings of International Conference on the Web Intelligence and Intelligent Agent Technology*, Sydney, Australia, pp. 560-566, Dec. 2008.

- [35] M. A. Khamis, W. Gomaa. "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework." *Engineering Applications of Artificial Intelligence*, Vol. 29, pp. 134-151, Mar. 2014.
- [36] C. Cai, C. K. Wong and B. G. Heydecker. "Adaptive traffic signal control using approximate dynamic programming." *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 5, pp. 456-474, Oct. 2009.
- [37] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls. "Reinforcement learning-based multi-agent system for network traffic signal control." *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128-135, Jun. 2010.
- [38] L. A. Prashanth and S. Bhatnagar. "Reinforcement learning with function approximation for traffic signal control." *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 412-421, Dec. 2011.
- [39] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvari and E. Wiewiora. "Fast gradient-descent methods for temporal-difference learning with linear function approximation." *Proceedings of the International Conference on Machine Learning*, Montreal, Quebec, pp. 993-1000, Jun. 2009.
- [40] H. R. Maei, C. Szepesvari, S. Bhatnagar and R. S. Sutton. "Toward off-policy learning control with function approximation." *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, pp. 719-726, Jun. 2010.
- [41] IEEE Working Group. "Standard specification for telecommunications and information exchange between roadside and vehicle systems-5 GHz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY) specifications." ASTM DSRC STD E2313-02, 2002.
- [42] B. J. Frey and D. Dueck. "Clustering by passing messages between data points." *Science*, vol. 315, no. 5814, pp. 972-976, Feb. 2007.
- [43] C. J. C. H. Christopher, Watkins, and P. Dayan. "Q-learning." *Machine Learning*, vol. 8, no. 3, pp. 279-292, May. 1992.
- [44] 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, Jul. 2012.
- [45] B. Lucian, R. Babuska and B. D. Schutter. "A comprehensive survey of multiagent reinforcement learning." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156-172, Mar. 2008.
- [46] G. F. Riley and T. R. Henderson. "The NS-3 network simulator, modeling and tools for network simulation." *Springer Berlin Heidelberg*, pp. 15-34, 2010.
- [47] D. Krajzewicz, G. Hertkorn, and C. Rössel. "SUMO (simulation of urban mobility)." *Proceedings of the 4th Middle East symposium on simulation and modeling*, pp. 183-187, 2002.
- [48] A. Wegener, M. Pirkowski, and M. Raya. "TraCI: an interface for coupling road traffic and network simulators." *Proceedings of the 11th communications and networking simulation symposium. ACM*, pp. 155-163, Apr. 2008.