Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Cooperative channel assignment for VANETs based on multiagent reinforcement learning*

Yun-peng WANG†, Kun-xian ZHENG†‡, Da-xin TIAN†, Xu-ting DUAN†, Jian-shan ZHOU

Beijing Advanced Innovation Center for Big Data and Brain Computing,
School of Transportation Science and Engineering, Beihang University, Beijing 100191, China

†E-mail: ypwang@buaa.edu.cn; zhengkunxian@buaa.edu.cn; dtian@buaa.edu.cn; duanxuting@buaa.edu.cn

Received June 21, 2019; Revision accepted Jan. 3, 2020; Crosschecked June 11, 2020

Abstract: Dynamic channel assignment (DCA) plays a key role in extending vehicular ad-hoc network capacity and mitigating congestion. However, channel assignment under vehicular direct communication scenarios faces mutual influence of large-scale nodes, the lack of centralized coordination, unknown global state information, and other challenges. To solve this problem, a multiagent reinforcement learning (RL) based cooperative DCA (RL-CDCA) mechanism is proposed. Specifically, each vehicular node can successfully learn the proper strategies of channel selection and backoff adaptation from the real-time channel state information (CSI) using two cooperative RL models. In addition, neural networks are constructed as nonlinear Q-function approximators, which facilitates the mapping of the continuously sensed input to the mixed policy output. Nodes are driven to locally share and incorporate their individual rewards such that they can optimize their policies in a distributed collaborative manner. Simulation results show that the proposed multiagent RL-CDCA can better reduce the one-hop packet delay by no less than 73.73%, improve the packet delivery ratio by no less than 12.66% on average in a highly dense situation, and improve the fairness of the global network resource allocation.

Key words: Vehicular ad-hoc networks; Reinforcement learning; Dynamic channel assignment; Multichannel
https://doi.org/10.1631/FITEE.1900308 **CLC number:** U495

1 Introduction

As an important component of intelligent transportation systems, vehicular ad-hoc networks (VANETs) provide data interaction services for vehicle infrastructure cooperative systems and automatic driving. Multichannel technology can significantly increase the capacity of VANETs and support future large-scale vehicle data interaction. Currently, the IEEE 802.11p/1609.4 protocol specifies a 75-MHz bandwidth in the 5.9-GHz frequency band for vehicular communications and divides the 75-MHz bandwidth into seven channels, including one control channel (CCH) and six service channels (SCHs) (Ahmed SAM et al., 2013) (Fig. 1). Channel assignment plays a key role in reaping the potential benefit of VANETs. Some works assign the SCHs based on the publish-subscribe model (Almohammedi et al., 2017; Ouyous et al., 2017). In other words, the service provider issues service advertisement with channel information, and then the user accesses the corresponding SCH. However, this method does not consider the interaction between the channel selection decisions made by different communication node pairs (CNPs) and the evolution of nodes' local environment state. Therefore, the performance

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. 61672082 and 61822101), the Beijing Municipal Natural Science Foundation, China (No. 4181002), and the Beihang University Innovation and Practice Fund for Graduate, China (No. YCSJ-02-2018-05)
ORCID: Kun-xian ZHENG, https://orcid.org/0000-0002-2887-9294
© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2020

of this method is greatly limited in the case where the node density is high. There are many channel assignment mechanisms based on the assistance of roadside units (RSUs) such that these mechanisms may not be applicable to fully distributed scenarios (Li XH et al., 2015; Atallah et al., 2017).

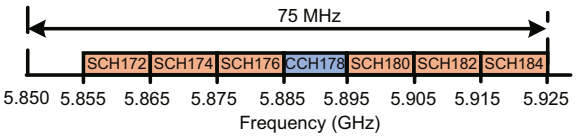

Fig. 1 Multichannel technology based on the IEEE 802.11p/1609.4 protocol

Generally, there exist several significant challenges that need to be addressed for the practical fulfilment of distributed and adaptive channel assignments under the vehicular direct communication scenario. First, accurate state information of the global network may not be practically available to each vehicular node. Second, it is not feasible to directly coordinate dynamic decision-making behaviors of CNPs in channel selection due to the absence of a central control. In addition, the complexity of joint optimization of channel selection and data collision avoidance can be increased with a large-scale network. However, few efforts have been made to simultaneously tackle all these challenges in dedicated short-range communication (DSRC) based vehicular direct communications.

In DSRC-based vehicular direct communication systems, dynamic channel assignment (DCA) algorithms should consider the dynamic communication requirements, the increased complexity due to the existence of large-scale transmission pairs, and the lack of centralized control and infrastructure. The difficulty lies in accurately modeling this time-varying multiagent coordination characteristic without global information and in making better channel allocation decisions. Another difficulty is the maximization of system performance over a long-term period in a dynamic vehicle mobility environment. The limitation of existing works and the challenges in connected vehicles as mentioned motivate us to explore a novel methodological framework.

By carefully looking into the DCA optimization problem, it is noted that this problem can be mathematically mapped into a sequential decision-making problem embedded in complicated time-varying multiagent environments. To handle this issue, we resort to the reinforcement learning (RL) based theory. As we can see, the emerging RL method shows great prospects in sequential decision-making problems (Mnih et al., 2013, 2015; Li L et al., 2016; Silver et al., 2017), and a variety of RL-based solutions have been applied in dynamic resource management and other domains (Mao et al., 2016; He et al., 2017; Liu N et al., 2017; Xu et al., 2017). In particular, the integration of the RL theory and artificial neural network has been believed to be a promising general artificial intelligence (AI) paradigm. For instance, the program AlphaGo's 4-1 victory was a historic stride for computer programmers and AI researchers trying to create software that can outwit humans in board games (Maddison et al., 2014). The AI system developed by OpenAI beat the world's top professionals at 1v1 matches of Dota 2 under standard tournament rules on August 11, 2017 (https://openai.com/blog/dota-2/). OpenAI Five won back-to-back games versus Dota 2 world champions OG at the finals on April 13, 2019, becoming the first AI system that beats the world champions in an Esports game (https://openai.com/blog/openai-five/). To sum up, the success of the RL theory in various domains, such as game, control, and stochastic optimization, motivates us to resort to the general AI approach based on the RL theory and artificial neural network.

In this study, we regard each vehicular node as a mobile agent. Then, motivated by the RL theory, we propose two cooperative RL models for each vehicular agent to jointly learn the proper policies of its channel selection and backoff adaptation with the goal of avoiding channel collision in a large-scale vehicular network. Furthermore, we construct neural networks as Q-function approximators, which are combined with the RL mechanism such that the vehicular agents can map their continuously observed state input to the policy output. To drive the agents to achieve collaborative optimization of their channel assignments, we propose a novel design of consensus reward by locally sharing and incorporating individual rewards. With the consensus reward, multiple agents are induced to dynamically coordinate their learning behaviors and policy adaptation in a distributed manner. To the best of our knowledge, our work presents the first application of a multiagent dual RL framework to jointly optimize the decision-making behaviors of channel selection and

backoff adaptation in DSRC-based vehicular communications (Kaelbling et al., 1996; Audhya et al., 2011; Cheeneebash et al., 2012; Arulkumaran et al., 2017; Wang W et al., 2017). In particular, formulating a local consensus reward for the RL of each agent can induce global agents to dynamically coordinate their channel access. This facilitates the mapping of the problem of distributed channel resource optimization into the cooperative decision-making process of multiple agents with RL.

The main contributions of this study are summarized as follows:

1. Although the existing RL theory has achieved success in a variety of domains, its applicability has previously been focused on gaming or robotic control domains or other domains where system features can be handcrafted. Few efforts have been made to explore the applicability of RL theory in promoting joint channel selection and medium access control (MAC) layer backoff for vehicle-to-vehicle (V2V) communication and networking. We present the first work to demonstrate the effectiveness and the potential of RL-based methodology in the joint design of channel selection decision-making and access backoff adaptation for enhancement of V2V communications.

2. Moreover, while our study is based on the existing RL theory, the methodological framework proposed in our study is not a direct application of existing RL algorithms. To facilitate the application of the RL theory in the field of connected vehicles, we propose a novel RL model in which two new components are designed and combined to adapt to the dynamically changing V2V communication network and improve the decision-making performance of RL agents in V2V communications: a dual Q-network structure for joint optimization on channel selection and backoff adaptation and a distributed consensus reward mechanism for boosting cooperative decision-making behaviors among learning agents.

2 Related works

Since AlphaGo achieved an overwhelming victory in the board game Go against a top human player, AI has once again received wide attention. As AlphaGo's core component, RL has been a hot topic in the field of AI for a long time (Maddison et al., 2014). The RL agent obtains reward r after taking action a and updates the action strategy of state s to adapt to the unknown environment (Barto and Sutton, 1998).

Extensive applications of RL can be found in adaptive optimal channel decision-making of wireless networks (Nie and Haykin, 1999; Louta et al., 2014; Ahmed T and Le Moullec, 2017; Ahmed T et al., 2017; Liu SJ et al., 2018). Nie and Haykin (1999) proposed a classical method using RL for channel allocation, i.e., Q-learning-based DCA, which uses RL to solve DCA problems in a cellular mobile communication system. The cellular network is divided into many cells with base stations, and neighboring cells cannot use the same frequency to avoid interferences induced by co-channels. Due to the limited quantity of channels, a reasonable channel assignment mechanism is important for reducing interferences and maximizing channel utilization. The base station as the RL agent assigns channels to each CNP and sets the quantity of available channels in the cells as state s, channel k assigned for the current communication request as action a, and the interference among neighboring cells as reward r. However, Q-learning-based DCA is a single-agent RL (SARL) mechanism, as well as a centralized one. Therefore, Q-learning-based DCA may not be applicable to the fully distributed scenario, and the possibility of synergy between various agents to optimize network performance is neglected.

After studying the performances of both SARL and multiagent RL (MARL) in cognitive wireless networks, Yau et al. (2010) concluded that MARL is more stable than SARL. In a distributed system, each RL agent improves its own efficiency in a selfish manner, which may lead to vicious competition between each other. On the contrary, multiagent coordination can achieve better system optimization. Seah et al. (2007) designed a wireless sensor network coverage scheme based on multiagent collaborative RL. The agent determines its own action based on its neighboring nodes' actions to reduce power consumption and increase the network lifetime. Agents share local rewards to estimate the impact of their actions in the global environment. However, in VANET, due to the rapid movement of nodes, limited channel coverage, and short data exchange time slots, it is almost impossible to obtain a global reward among agents. Therefore, we propose a more realistic multiagent coordination scheme to

replace the global optimization with regional consistency optimization.

In Atallah et al. (2017), deep RL was applied to VANET. Specifically, it was used to implement the vehicle-to-infrastructure communication that meets the quality of service requirements. The roadside unit can distinguish different system states according to its remaining battery, the quantity of mobile nodes, and the communication requests before assigning a suitable SCH to each mobile node. Atallah et al. (2017) used deep RL to learn the optimal strategy from high-dimensional continuous input states, proving that deep RL can effectively express a complex VANET environment and learn from experience. However, the mechanism proposed by Atallah et al. (2017) is based on the assistance of RSUs, which may be not applicable to a fully distributed scenario such as Q-learning-based DCA.

In the research on DSRC-based vehicular communications (Wang Q et al., 2012; Almohammedi et al., 2017; Ouyous et al., 2017), researchers focused on the optimization of CCH and SCH intervals. Almohammedi et al. (2017) proposed a measurement-based congestion detection method, which enables the RSU to work as follows: estimate the nearby traffic density at the end of each CCH interval, compare the CCH busy ratio with a predefined threshold, and finally decide the type of channel access scheme. Wang Q et al. (2012) proposed a variable CCH interval multichannel MAC scheme, which can dynamically adjust the length ratio between CCH and SCH intervals. These studies are still based on the traditional publish-subscribe model in channel selection and random backoff in channel access. Traditional channel allocation and channel access are both simple and mechanical, and cannot optimally adapt to the external network environment.

At present, there is little research on RL-based DCA-oriented solutions under a vehicular direct communication scenario. In this study, we demonstrate the applicability, effectiveness, and advantages of the RL-based methodology in this field, and enrich the intellectual system of connected vehicles. We propose a dual-network multiagent RL model and a consensus reward mechanism to solve the DCA issue, which is not a direct application of existing RL models.

3 System model and problem formulation

We consider a typical DSRC-based vehicular network with K available channels $\mathcal{K} = \{1, 2, ..., K\}$. According to the IEEE 802.11p/1609.4 specification, a vehicular source needs to first negotiate with its destination over the CCH to select a good SCH. Then, after the source switches to the selected target SCH from the CCH, the source additionally schedules a proper backoff before actually transmitting its data over this SCH. In this situation, without an appropriate channel assignment mechanism, the source–destination pair may fail in data transmission or the performance may significantly decrease due to the potential serious collisions when large-scale communication pairs are activated over the same SCH at the same time. Therefore, we need to design a channel assignment mechanism for each communication pair. Moreover, the following challenging aspects should be taken into consideration:

1. Dynamic communication requirements. Diverse vehicular applications will impose heterogeneous and time-varying requirements on vehicular communications, which can lead to time-varying occupancy of SCHs. Thus, it is highlighted that the channel assignment needs to adapt itself to the dynamic communication requirements.

2. Increased complexity with existence of large-scale transmission pairs. The large-scale vehicular nodes can exacerbate the complexity of the overall scenario and the intensity of data collisions, which further results in unpredictable and highly time-varying channel evolution.

3. The lack of centralized control and infrastructure. Each vehicular node can obtain neither the global information on the network performance nor the decision-making actions of all the other nodes. This implies that a direct global optimization paradigm may be infeasible for the fully distributed situation.

By incorporating the above considerations, we map the problem into the RL formulation as follows:

1. State. Let $\mathbb{S}_i = \{\mathcal{S}_{i1}, \mathcal{S}_{i2}, \ldots, \mathcal{S}_{iT}\}$ be the set of time-varying states that can be directly observed by node $i \in \mathcal{M}_t$. Specifically, node i observes $\mathcal{S}_{iT}$ by counting the sending and receiving information of clear-to-send (CTS)/request-to-send (RTS) frames over CCH, as well as that of receipt

acknowledgement (ACK) frames over the SCH, where $\mathcal{S}_{it}$ $(1 \le t \le T)$ represents the state in the t^{th} time slot and $\mathcal{M}_t$ is the set of network nodes in the t^{th} time slot. The duration of each time slot is set to 100 ms, which is the synchronization interval (SI) specified in IEEE 802.11p/1609.4. Specifically, $\mathcal{S}_{it} = [\boldsymbol{H}_{it}, l_{it}, z_{i(t-1)}]$ consists of three types of CSI: (1) $\boldsymbol{H}_{it}$ is the vector containing the number of CNPs selecting each channel in the t^{th} SI, i.e., $\boldsymbol{H}_{it} = [h_{1it}, h_{2it}, \ldots, h_{Kit}]$, where h_{kit} $(1 \le k \le K)$ denotes the number of CNPs intending to transmit packet over the t^{th} channel; (2) l_{it} denotes the local communication requirement of node i; (3) $z_{i(t-1)}$ denotes the backoff window size of the $(t-1)^{\text{th}}$ SI.

2. Action. $\mathcal{A}_{it} = [k_{it}, w_{it}] \in \mathbb{A}_i = \{\mathcal{A}_{i1}, \mathcal{A}_{i2}, \ldots, \mathcal{A}_{iT}\}$, where k_{it} represents the index of the SCH selected in the t^{th} SI by node i ($k_{it} \in \mathcal{K} = \{1, 2, \ldots, K\}$) and w_{it} represents the index of each type of backoff action. Specifically, we consider that each transmitter can take three types of backoff actions, denoted by the set $\mathcal{W} = \{W_1, W_2, W_3\}$, where W_1 denotes that the transmitter maintains the current backoff window size as $z_{it} = z_{i(t-1)}$, W_2 denotes that the transmitter increases the backoff window size as $z_{it} = 2z_{i(t-1)} + 1$, and W_3 is the action that the transmitter reduces the backoff window size to $z_{it} = [z_{i(t-1)} - 1]/2$.

3. Reward. We use $r_{it}(\mathcal{S}_{it}, \mathcal{A}_{it})$, $r_{it} \in \mathbb{R}_i = \{r_{i1}, r_{i2}, \ldots, r_{iT}\}$, to denote the reward perceived by a source node i in the t^{th} SI. To be specific, we take the demand dynamics of the upper-layer application and the communication performance into consideration, so we formulate the node's reward by

$$r_{it}(\mathcal{S}_{it}, \mathcal{A}_{it}) = \left(\frac{m_{\text{recv}}}{m_{\text{trans}}}\right)^{\psi} \left(\frac{m_{\text{recv}}}{m_{\text{trans}} + m_{\text{queue}}}\right)^{\beta}, \tag{1}$$

where ψ and β are two positive weights, m_{trans} denotes the number of packets transferred by node i in the t^{th} SI, m_{queue} denotes the number of packets waiting in the buffer queue that need to be transferred in the t^{th} SI, and m_{recv} denotes the number of packets successfully delivered to the receiver by node i in the t^{th} SI. Note that the sum number of the packets already transferred and those in the buffer, i.e., $m_{\text{trans}} + m_{\text{queue}}$, represents the time-varying demand of an upper-layer application, assuming that the set of other n neighboring CNPs' local rewards is $\mathbb{R}'_{it} = \{r'_{it1}, r'_{it2}, \ldots, r'_{itN}\}$, where r'_{itn} is the local reward of node i's n^{th} $(n = 1, 2, ..., N)$ neighboring CNP in the t^{th} SI.

4. State transition probability. Given the current state $\mathcal{S}_{it}$ and action $\mathcal{A}_{it} = [k_{it}, w_{it}]$ in the t^{th} SI, the probability that node i transits to another state $\mathcal{S}_{i(t+1)}$ in the $(t+1)^{\text{th}}$ SI is denoted by $p_{it}(\mathcal{S}'_{i(t+1)} \mid \mathcal{S}_{it}, \mathcal{A}_{it})$, where $\mathcal{S}'_{i(t+1)} \in \mathbb{S}'_i = \{\mathcal{S}'_1, \mathcal{S}'_2, \ldots, \mathcal{S}'_U\}$ ($\mathbb{S}'_i$ is the set of all possible states observed by node i).

To achieve collaborative optimization of multiple agents in a distributed manner, we propose a novel reward formulation with a weighted sum strategy, termed "consensus reward," which is motivated by locally sharing and combining individual rewards. Specifically, the consensus reward is constructed as follows:

$$R_{it}(\mathcal{S}_{it}, \mathcal{A}_{it}) = r_{it}(\mathcal{S}_{it}, \mathcal{A}_{it}) + \sum_{r'_{itn} \in \mathbb{R}'_{it}} \alpha_{itn} r'_{itn}(\mathcal{S}_{it}, \mathcal{A}_{it}), \tag{2}$$

where α_{itn} is the weight of the n^{th} neighboring CNP. Given the current state $\mathcal{S}_{i1}$ observed by node i, we need to find the best action sequence $\mathbb{A}_i$ to maximize the following overall expected reward:

$$V_i(\mathcal{S}_{i1}, \mathcal{A}_{i1}) = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} R_{it}(\mathcal{S}_{it}, \mathcal{A}_{it})\right], \tag{3}$$

where $\gamma \in [0, 1]$ is the discount factor. Let π_i be a strategy of node i, i.e., a probability distribution over sequences $\mathcal{S}_i$ and actions $\mathcal{A}_i$. According to the transition probabilities, we can further obtain

$$\begin{aligned} V_i^{\pi_i}(\mathcal{S}_{i1}) = &\mathbb{E}\Big[R_{i1}(\mathcal{S}_{i1}, \mathcal{A}_{i1}) \\ &+ \gamma \sum_{\mathcal{S}'_{i2} \in \mathbb{S}'_i} p_{i1}(\mathcal{S}'_{i2} \mid \mathcal{S}_{i1}, \mathcal{A}_{i1}) V^{\pi_i}(\mathcal{S}'_{i2})\Big]. \end{aligned} \tag{4}$$

Finally, the problem of looking for the action sequence that maximizes $V_i(\mathcal{S}_{i1}, \mathcal{A}_{i1})$ in Eq. (3) is transformed into finding the action strategy π_i^* that maximizes $V_i^{\pi_i}(\mathcal{S}_{i1})$ in Eq. (4). That is, π_i^* satisfies the following expression:

$$\pi_i^* = \arg\max_{\pi_i \in \Pi} V_i^{\pi_i}(\mathcal{S}_{i1}), \tag{5}$$

where Π is the set of action strategies.

4 The proposed channel assignment mechanism

4.1 Multiagent RL-CDCA

Here, we resort to the RL theory to solve problem (5). In reality, there may exist a quite large number of states and actions in the multiagent system such that it is impractical to record a Q-table to map every state–action pair into a Q-value. Thus, the conventional Q-learning mechanism is impractical to address the curse of dimensionality. Instead, we refer to the neural network as a nonlinear function approximator to estimate the large-scale action-value mapping. Specifically, we construct two neural networks, i.e., two dual neural networks, one of which—denoted by $Q(\boldsymbol{S}_{it\theta}, k_{it}; \theta)$—is used for channel selection and the other—denoted by $Q(\boldsymbol{S}_{it\delta}, w_{it}; \delta)$—is used for backoff adaptation. Here, θ and δ denote the weights of the dual Q-networks, and $\boldsymbol{S}_{it\theta} = [\boldsymbol{H}_{it}, l_{it}]$ and $\boldsymbol{S}_{it\delta} = [l_{it}, z_{i(t-1)}]$ denote the inputs of the dual Q-networks. Fig. 2 shows our methodological framework.

Another issue to be handled for implementation of RL with the dual Q-networks is to guarantee the convergence of the Q-networks. In this study, we use experience replay and target networks to improve the stability of the Q-networks. During the training process, each newly generated experience tuple $(\mathcal{S}_{it}, \mathcal{A}_{it}, R_{it}, \mathcal{S}_{i(t+1)})$ is stored in the replay memory Ω with a size Ω_{q}. The Q-networks begin to be trained when the number of tuples in replay memory Ω reaches a threshold C. To speed up the training, every F steps, we randomly sample a set of minibatch tuples $\mathbb{SA}_i = \{(\mathcal{S}_{is}, \mathcal{A}_{is}, R_{is}, \mathcal{S}_{i(s+1)})\}$ from Ω with a size I_{q}, where s denotes the SI index of a sample. Then, each Q-network can be trained by minimizing the following loss function over the minibatch tuple $\mathbb{SA}_i$:

$$L(G) = \mathbb{E}\left\{[Y_{isG} - Q(\mathcal{S}_{isG}, \mathcal{A}_{isG}; G)]^2\right\}, \quad (6)$$

where $G = \theta$ or δ and $\mathcal{A}_{isG} = k_{is}$ or w_{is}. Thus, Y_{isG} is the target value, and can be evaluated as follows:

$$Y_{isG} = R_{is} + I_{\mathcal{A}_{i(s+1)G} \neq \varnothing}(\mathcal{A}_{i(s+1)G}) \cdot \gamma \max_{\mathcal{A}_{isG} \in \mathcal{A}} \hat{Q}(\mathcal{S}_{i(s+1)G}, \mathcal{A}_{isG}; G^-), \quad (7)$$

where $\mathcal{A} = \mathcal{K}$ or $\mathcal{W}$ and $I_{\mathcal{A}_{i(s+1)G} \neq \varnothing}(\mathcal{A}_{i(s+1)G}) = 1$ when $\mathcal{A}_{i(s+1)G} \neq \varnothing$. Otherwise, $I_{\mathcal{A}_{i(s+1)G} \neq \varnothing}(\mathcal{A}_{i(s+1)G}) = 0$, and $\hat{Q}(\mathcal{S}_{i(s+1)G}, \mathcal{A}_{isG}; G^-)$ is the target network with the same structure. The training algorithm is provided in Algorithm 1.

It is noteworthy that our DCA method follows a distributed online multiprocess implementation. In other words, the policy learning process and decision-making process can be performed in

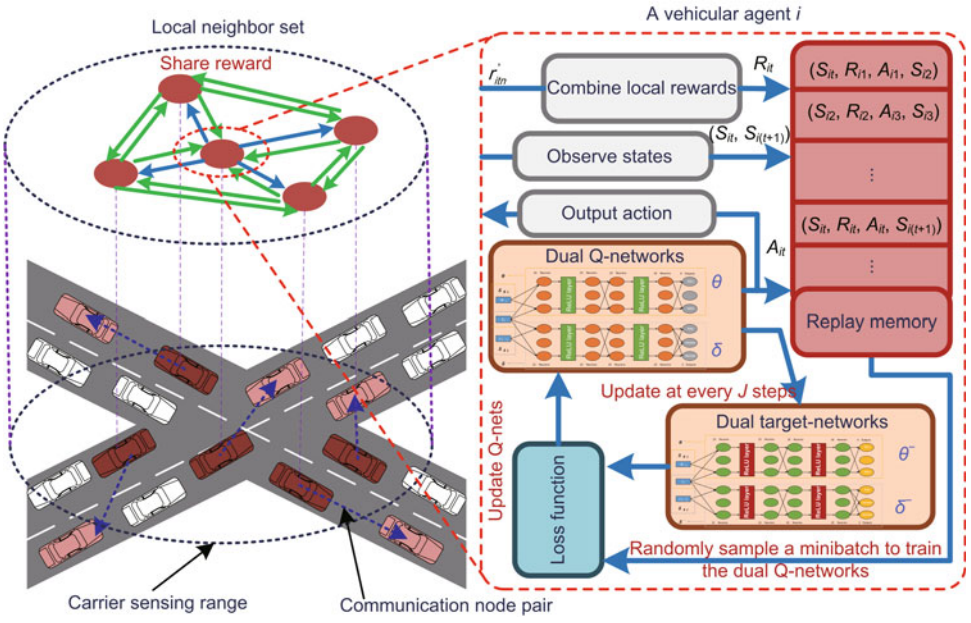

Fig. 2 Scenario of multiple coexisting communication node pairs (CNPs) driven by multiagent RL-CDCA

a parallel online manner. The long-term decision-making experience represented by a series of state actions and Q-value samples are continuously recorded based on the continuous interaction between agents and their environment, and the agents' policies are continuously updated according to their learned experience. Meanwhile, each agent can make decisions according to its learned policy in successive channel assignment slots.

Algorithm 1 Multiagent RL-CDCA

Require:

Initialize sampling threshold C, training cycle F, update cycle J, memory size Ω_{q}, minibatch tuple size I_{q}, the ϵ-greedy policy, Q-networks θ and δ with random weights, and target Q-networks $\theta^{-}=\theta$ and $\delta^{-}=\delta$

Ensure:

for $t=0\rightarrow+\infty$ **do**
  if in CCH interval **then**
    Each CNP coordinates SCH k_{it} by RTS/CTS
  end if
  if in SCH interval **then**
    Each CNP switches to its SCH k_{it}
    Source randomly backs off $t\in[0,z_{it}]$
    Source observes $\mathcal{S}_{i(t+1)}$
    Each CNP transmits service data
  end if
  if next CCH interval **then**
    Source calculates local reward by Eq. (1)
    Source broadcasts r_{it}
  end if
  if next SCH interval **then**
    Source calculates the consensus reward by Eq. (2)
    Store tuple $(\mathcal{S}_{it},\mathcal{A}_{it},R_{it},\mathcal{S}_{i(t+1)})$ in memory Ω
  end if
  At every F steps, randomly sample a set of minibatch tuples $\mathbb{SA}_i=\{(\mathcal{S}_{is},\mathcal{A}_{is},R_{is},\mathcal{S}_{i(s+1)})\}$
  Train the dual Q-networks by the root mean square propagation (RMSProp) algorithm
  At every J steps, update the target networks with weights
  $\theta^{-}=\theta$ and $\delta^{-}=\delta$
end for

4.2 Complexity analysis

The training algorithm includes normalization, replay buffer, dual Q-network approximators each with four neural networks, and target dual Q-network approximators, while the trained algorithm is made up of only normalization and dual Q-network approximators. Next, we separately analyze the time complexity (computations) and space complexity (memory) of the training and trained RL-CDCA, wherein the time complexity is represented by the number of floating point operations per second (FLOPS).

4.2.1 Training

Since the VANET network environment changes constantly, the normalization has to be conducted at all steps during the training stage. The time complexity of state normalization is $N(\mathcal{S}_{it})$, where $N(\mathcal{S}_{it})$ is the number of variables in the state set $\mathcal{S}_{it}$. To avoid repeated calculations, the algorithm has to record the means and standard deviations of the state variables, so the space complexity of normalization is $2N(\mathcal{S}_{it})$. The experience replay buffer in RL-CDCA occupies some space to store $\mathcal{S}_{it}$. Hence, the space complexity is the size of minibatch tuple $N(I_{\mathrm{q}})$.

For neural networks, the computation of activation layers needs to be analyzed. When calculating FLOPS, addition, subtraction, multiplication, division, exponentiation, square root, and so on are usually counted as a single FLOPS. The computation is X with X inputs for rectified linear unit (ReLU) layers, $4X$ for sigmoid layers, and $6X$ for tanh layers (Qiu et al., 2019).

The input $\mathcal{S}_{it}$ is a state set of continuous values, not an image. Hence, there is no convolution layer in the channel selection net θ or the backoff adaptation net δ, and only fully connected layers are present. Assuming that each net in the dual Q-network has $N(Q)$ fully connected layers, and considering the bias of adding in fully connected layers and the target dual Q-network, the time complexity can be calculated as follows:

$$\begin{aligned}&4\left[v_{\mathrm{act}}\sum_{n=1}^{N(Q)-2}N(Q)_n+\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\right]\\&=O\left(\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\right),\end{aligned}\tag{8}$$

where $N(Q)_n$ represents the unit number in the n^{th} layer, $N(Q)_0$ equals the size of $\mathcal{S}_{it}$, and v_{act} indicates the corresponding parameter determined by the type of the activation layer.

For a fully connected layer, there is an $N(Q)_n \times N(Q)_{n+1}$ weight matrix and an $N(Q)_{n+1}$-dimensional bias vector. Hence, the memory of one fully connected layer is $[N(Q)_n + 1]N(Q)_{n+1}$. Because the activation does not need to save weights, the space complexity of the dual Q-network and its target network is formulated as follows:

$$4\sum_{n=0}^{N(Q)-1}[N(Q)_n+1]N(Q)_{n+1} = O\Bigg(\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\Bigg). \tag{9}$$

For each training step, the computation of the consensus reward is $N(r'_{itn})$, where $N(r'_{itn})$ is the number of individual rewards shared by neighboring nodes. Therefore, the overall time complexity of our training algorithm is represented as follows:

$$4\Bigg[v_{\text{act}}\sum_{n=1}^{N(Q)-2}N(Q)_n+\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\Bigg] + N(\mathcal{S}_{it}) + N(r'_{itn}) = O\Bigg(\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\Bigg) + O(N(\mathcal{S}_{it})) + O(N(r'_{itn})), \tag{10}$$

and the overall space complexity of our training algorithm is as follows:

$$4\sum_{n=0}^{N(Q)-1}[N(Q)_n+1]N(Q)_{n+1} + N(\mathcal{S}_{it}) + N(I_{\text{q}}) = O\Bigg(\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\Bigg) + O(N(\mathcal{S}_{it})) + O(N(I_q)). \tag{11}$$

4.2.2 Trained RL-CDCA

After training, there is no replay buffer, target dual Q-network, or consensus reward calculation in the trained algorithm. Only state normalization and dual Q-network approximators are needed. Therefore, the time complexity of the trained algorithm is as follows:

$$2\Bigg[v_{\text{act}}\sum_{n=1}^{N(Q)-2}N(Q)_n+\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\Bigg] + N(\mathcal{S}_{it}) = O\Bigg(\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\Bigg) + O(N(\mathcal{S}_{it})), \tag{12}$$

and the space complexity is as follows:

$$2\sum_{n=0}^{N(Q)-1}[N(Q)_n+1]N(Q)_{n+1} + N(\mathcal{S}_{it}) = O\Bigg(\sum_{n=0}^{N(Q)-1}N(Q)_nN(Q)_{n+1}\Bigg) + O(N(\mathcal{S}_{it})). \tag{13}$$

5 Performance evaluation

5.1 Simulation parameters

We perform extensive comparative simulations to show the performance of our proposed method. Specifically, the algorithms and simulations are implemented using a well-known VANET simulator Veins (http://veins.car2x.org/) and a C++ library MLPACK (http://www.mlpack.org/) for machine learning. The simulation scenario (Fig. 3) is a part of a scenario based on the city of Erlangen in Bavaria,

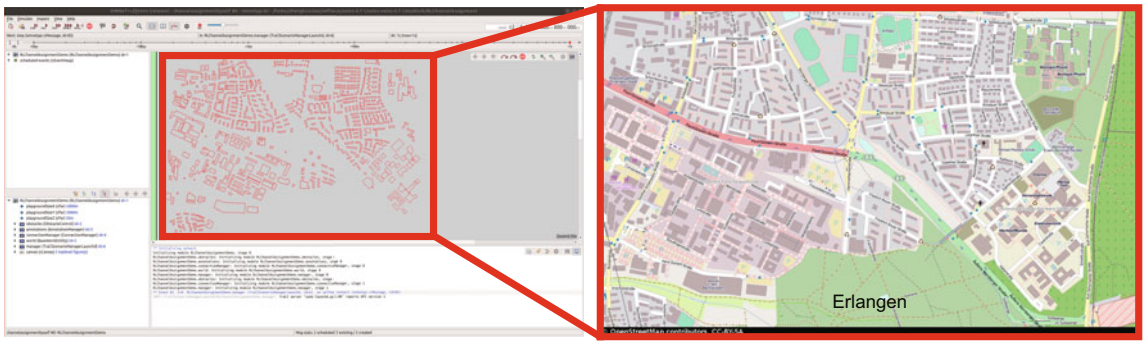

Fig. 3 Simulation scenario

Germany. Based on the observation of the simulation, assume $N(r'_{itn}) = 10$ when analyzing the complexity. The parameter setting and complexity analysis results are given in Table 1. The major procedures involved in the simulations are shown in Fig. 4. We compare our method with four existing schemes:

1. Random assignment scheme (Random). Each CNP randomly selects SCH in each SI.

2. Greedy selection scheme (Greedy). Each CNP selects SCH with the smallest number of subscribers.

3. Fixed channel assignment scheme (Fixed). Every CNP transmits data over an SCH during the whole simulation and does not change its target SCH.

4. Decentralized Q-learning (DQ). This is a Q-learning-based decentralized wireless communication resource allocation method for the V2V communication scenario. It is a DSRC-based variant of the method proposed by Ye et al. (2018).

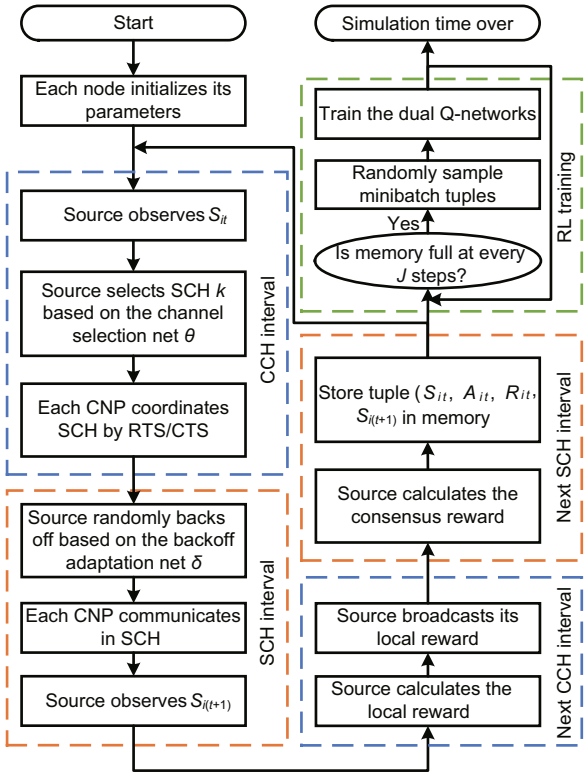

Fig. 4 Flowchart for simulations

The packet delivery ratio, one-hop packet delay, and fairness of packet delivery ratio are used as performance metrics. In this study, these three metrics are defined as follows:

1. Packet delivery ratio. This is the average transmit success rate of service data packets sent in an SCH interval. We calculate it as $\%\text{Ratio} = \text{Msg}_\text{r}/\text{Msg}_\text{s} \times 100$, where Msg_r and Msg_s are the service packets that are successfully received and sent, respectively, by all nodes during the entire simulation.

2. One-hop packet delay. This is the average delay of service packets transmitted between the source node and the application layer of the destination node. We calculate it as $\text{Delay} = T_\text{total}/\text{Msg}_\text{r}$, where T_total (s) is the total delay of all successfully received service packets during the entire simulation.

3. Fairness of packet delivery ratio. This is the fairness of a set of packet delivery ratios where there are n vehicular connections. We calculate it as $\text{Fairness}(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n\sum_{i=1}^{n} x_i^2}$, where x_i is the packet delivery ratio of the i^th connection (Jain et al., 1998). The fairness ranges from $1/n$ (the worst case) to 1 (the best case). The result is k/n when k connections equally share the channel resource and the other $n-k$ connections receive zero allocation. We can use this metric to identify underutilized channels.

Table 1 Simulation parameters

Parameter	Value
Channel size K	4
Density ρ	[50, 200, 400] (vehicles)
Discount factor γ	0.8
Learning rate α	0.01
Sampling threshold C	10
Training cycle F	100 ms
Number of update steps J	20
Memory size Ω_q	100
Minibatch tuple size I_q	10
ϵ-greedy policy	10%
ψ, β, $\forall\alpha_{itn}$	1
Number of fully connected layers, $N(Q)$	4
Number of units in each layer, $N(Q)_n$	10
Activation	ReLU
Training computation	1296
Training memory	1492
Trained computation	646
Trained memory	674

5.2 Results

Fig. 5 shows the convergence of the multiagent RL-CDCA. It can be seen that the Q-values of different actions converge to a steady state after about 1000 iterations. The average time of each

iteration is $\leq$ 10 ms. From Fig. 6a, it can be seen that increasing vehicle density can reduce the packet delivery ratio due to the fact that more data collisions occur in a denser traffic situation. Nevertheless, our method outperforms the other four methods even in a highly dense situation. For instance, when the number of vehicles is set to 400, our proposed method can achieve a higher packet delivery ratio of 13.83%, 21.98%, 16.97%, and 12.66% than the Random, Greedy, Fixed, and DQ methods, respectively. In the Random method, the channel selection is random and current network conditions are not considered; therefore, the Random approach is more likely to choose a congested channel with increasing vehicle density, resulting in significant reduction in the packet delivery ratio. For the Fixed model, each node's SCH is fixed; therefore, compared with the Random method, it is more difficult for the Fixed model to avoid congested channels. This result shows that the packet delivery ratio of the Fixed model is always lower than that of the Random model. Interestingly, although the Greedy model always selects the least-congested channel, its packet delivery ratio is the lowest. This can be explained by the fact that the Greedy algorithm obtains the congestion information of each channel by the CTS command, which hardly reflects the network change in time because of the delay in CTS transmission; therefore, a large number of Greedy nodes may access the same channel (the previous least-congested channel). Then, this least-congested channel becomes the most congested channel, and accordingly, the service packets of these Greedy nodes are more likely to collide. Now, for the DQ method, we believe that its packet delivery ratio is lower than that of RL-CDCA mainly because of the lack of multiagent cooperation; specifically, in the case of highly dense situations, the nodes' respective optimal selections are much more likely to conflict, which is similar to the Greedy method. Considering our multiagent RL-CDCA, it learns the channel selection policy from experience and achieves multiagent collaborative optimization by the consensus reward. Obviously, the performance of multiagent RL-CDCA is significantly better than those of the four existing mechanisms.

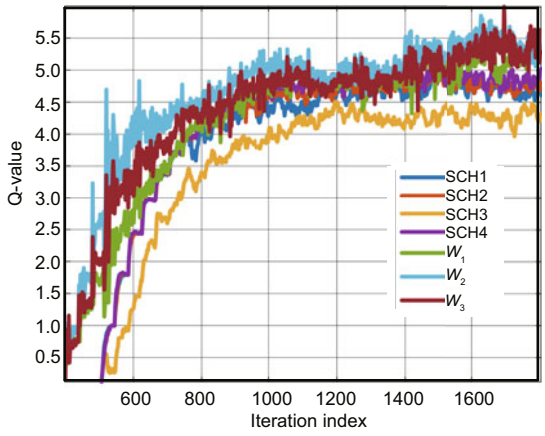

Fig. 5 Convergence performance of the multiagnet RL-CDCA

From Fig. 6b, it can be observed that the one-hop packet delay can be increased by increasing the vehicle density. This is because the busy duration of each SCH can be increased when more communication nodes exist. In addition, Fig. 6b illustrates that our method can achieve the lowest one-hop packet delay under each situation. Even when the vehicle density is high, the one-hop packet delay of our method is 73.73%, 73.65%, 76.00%, and 74.07% lower than those of the Random, Greedy, Fixed, and DQ methods, respectively. The main reason is that we jointly optimize channel selection and backoff

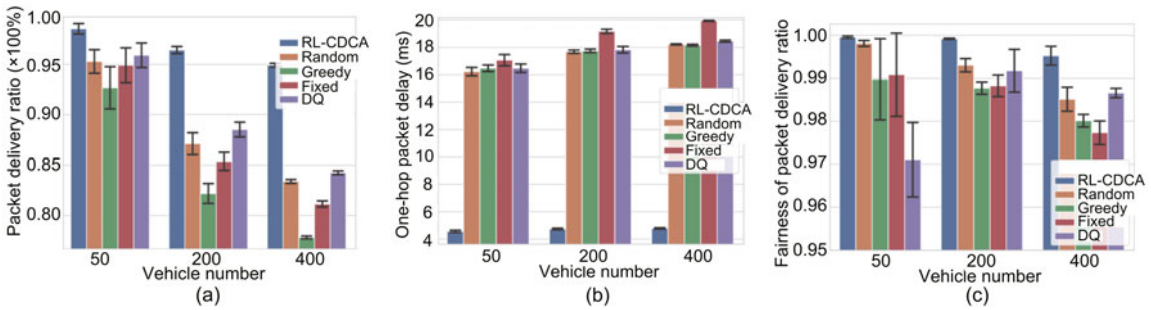

Fig. 6 Simulation results of different methods: (a) packet delivery ratio; (b) one-hop packet delay; (c) fairness of packet delivery ratio

adaptation, which enables the node to select the optimal channel and access it at the optimal time, thereby greatly reducing the channel access waiting time and thus the one-hop packet delay. Moreover, we compare the fairness of the global nodes' performances by introducing Jain's fairness index (Jain et al., 1998). From Fig. 6c, we find that our method can achieve the best fairness performance. Note that the Greedy, Fixed, and DQ models have large variance in fairness in the low-density scenario. Considering the Greedy and Fixed models, neither of them balances communication resource allocation well. Greedy suffers from its "misguided" channel selection, and Fixed suffers from its "limited" communication resources. Simulation results show that the smaller the base of resource allocation (the lower the vehicle density), the more obvious this unfairness. In the case of DQ, although it is a learning-based algorithm driven by rewards to learn the optimal resource allocation method, it does not learn a fair resource allocation policy in the low-density scenario, which proves the importance of multiagent collaboration in the low-density scenario. In the Random method, because of its random allocation of resources, its fairness in resource allocation is relatively high. RL-CDCA achieves regional collaborative optimization by means of the consensus reward, which further improves the fairness of resource allocation. To sum up, Fig. 6 confirms the advantage of our proposed method, which can achieve higher efficiency and better fairness performance.

6 Conclusions

In this study, we have dealt with the challenging issue of DCA in VANETs by proposing a multiagent RL framework combined with dual Q-network approximators. In this framework, each vehicular node can jointly adapt its channel selection decision and backoff window to the dynamic application demand and channel condition. To achieve collaborative optimization, we formulated a consensus reward for neighboring nodes. Simulation results showed that the proposed method significantly outperforms the four existing methods in terms of efficiency and fairness performance. To sum up, our main contributions are as follows: First, we have applied a dual RL framework to jointly optimize the decision-making behaviors of channel selection and backoff adaptation in DSRC-based vehicular communications. Second, we have used the consensus reward to replace the global reward, which is difficult to achieve because of the short CCH interval. The consensus reward is the key to multiagent collaborative optimization in DSRC-based scenarios. The results in this study can be used to develop optimal channel access control and next node selection strategies for efficient and reliable multihop routing applications. Another research direction is to extend the multiagent channel assignment optimization paradigm to more complex cooperation situations, where multi-hop routing should be carefully modeled and integrated into the policy learning and decision-making processes.

Contributors

Yun-peng WANG designed the research. Kun-xian ZHENG processed the data and drafted the manuscript. Da-xin TIAN and Xu-ting DUAN helped organize the manuscript. Kun-xian ZHENG and Jian-shan ZHOU revised and finalized the paper.

Compliance with ethics guidelines

Yun-peng WANG, Kun-xian ZHENG, Da-xin TIAN, Xu-ting DUAN, and Jian-shan ZHOU declare that they have no conflict of interest.

References

Ahmed SAM, Ariffin SHS, Fisal N, 2013. Overview of wireless access in vehicular environment (wave) protocols and standards. *Ind J Sci Technol*, 7(6):4994-5001. https://doi.org/10.17485/ijst/2013/v6i7/34355

Ahmed T, Le Moullec Y, 2017. A QoS optimization approach in cognitive body area networks for healthcare applications. *Sensors*, 17(4):780. https://doi.org/10.3390/s17040780

Ahmed T, Ahmed F, Le Moullec Y, 2017. Optimization of channel allocation in wireless body area networks by means of reinforcement learning. IEEE Asia Pacific Conf on Wireless and Mobile, p.120-123. https://doi.org/10.1109/APWiMob.2016.7811445

Almohammedi AA, Noordin NK, Sali A, et al., 2017. An adaptive multi-channel assignment and coordination scheme for IEEE 802.11p/1609.4 in vehicular ad-hoc networks. *IEEE Access*, 6:2781-2802. https://doi.org/10.1109/ACCESS.2017.2785309

Arulkumaran K, Deisenroth MP, Brundage M, et al., 2017. A brief survey of deep reinforcement learning. *IEEE Signal Process Mag*, 34(6):26-38. https://doi.org/10.1109/MSP.2017.2743240

Atallah R, Assi C, Khabbaz M, 2017. Deep reinforcement learning-based scheduling for roadside communication networks. 15th Int Symp on Modeling and Optimization in Mobile, p.1-8. https://doi.org/10.23919/WIOPT.2017.7959912

Audhya GK, Sinha K, Ghosh SC, et al., 2011. A survey on the channel assignment problem in wireless networks. *Wirel Commun Mob Comput*, 11(5):583-609. https://doi.org/10.1002/wcm.898

Barto AG, Sutton RS, 1998. Reinforcement Learning: an Introduction. MIT Press, Cambridge, MA, USA.

Cheeneebash J, Lozano JA, Rughooputh HCS, 2012. A survey on the algorithms used to solve the channel assignment problem. *Rec Pat Telecommun*, 1(1):54-71. https://doi.org/10.2174/2211740711201010054

He Y, Zhao N, Yin HX, 2017. Integrated networking, caching, and computing for connected vehicles: a deep reinforcement learning approach. *IEEE Trans Veh Technol*, 67(1):44-55. https://doi.org/10.1109/TVT.2017.2760281

Jain RK, Chiu DMW, Hawe WR, 1998. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. CoRR. cs. NI/9809099, DEC, Hudson, Canada.

Kaelbling LP, Littman ML, Moore AW, 1996. Reinforcement learning: a survey. *J Artif Intell Res*, 4(1):237-285. https://doi.org/10.1613/jair.301

Li L, Lv YS, Wang FY, 2016. Traffic signal timing via deep reinforcement learning. *IEEE/CAA J Autom Sin*, 3(3):247-254. https://doi.org/10.1109/JAS.2016.7508798

Li XH, Hu BJ, Chen HB, et al., 2015. An RSU-coordinated synchronous multi-channel MAC scheme for vehicular ad hoc networks. *IEEE Access*, 3:2794-2802. https://doi.org/10.1109/ACCESS.2015.2509458

Liu N, Li Z, Xu JL, et al., 2017. A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. IEEE 37th Int Conf on Distributed Computing Systems, p.372-382. https://doi.org/10.1109/ICDCS.2017.123

Liu SJ, Hu X, Wang WD, 2018. Deep reinforcement learning based dynamic channel allocation algorithm in multibeam satellite systems. *IEEE Access*, 6:15733-15742. https://doi.org/10.1109/ACCESS.2018.2809581

Louta M, Sarigiannidis P, Misra S, et al., 2014. RLAM: a dynamic and efficient reinforcement learning-based adaptive mapping scheme in mobile WiMAX networks. *Mob Inform Syst*, 10(2):173-196. https://doi.org/10.1155/2014/213056

Maddison CJ, Huang A, Sutskever I, et al., 2014. Move evaluation in go using deep convolutional neural networks. https://arxiv.org/abs/1412.6564

Mao HZ, Alizadeh M, Menache I, et al., 2016. Resource management with deep reinforcement learning. Proc 15th ACM Workshop on Hot Topics in Networks, p.50-56. https://doi.org/10.1145/3005745.3005750

Mnih V, Kavukcuoglu K, Silver D, et al., 2013. Playing Atari with deep reinforcement learning. https://arxiv.org/abs/1312.5602

Mnih V, Kavukcuoglu K, Silver D, et al., 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529-533. https://doi.org/10.1038/nature14236

Nie JH, Haykin S, 1999. A dynamic channel assignment policy through Q-learning. *IEEE Trans Neur Netw*, 10(6): 1443-1455. https://doi.org/10.1109/72.809089

Ouyous M, Zytoune O, Aboutajdine D, 2017. Multi-channel coordination based MAC protocols in vehicular ad hoc networks (VANETs): a survey. In: El-Azouzi R, Menasche D, Sabir E, et al. (Eds.), Advances in Ubiquitous Networking 2. Springer, Singapore. https://doi.org/10.1007/978-981-10-1627-1_7

Qiu CR, Hu Y, Chen Y, et al., 2019. Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. *IEEE Int Things J*, 6(5):8577-8588. https://doi.org/10.1109/JIOT.2019.2921159

Seah MWM, Tham CK, Srinivasan V, et al., 2007. Achieving coverage through distributed reinforcement learning in wireless sensor networks. 3rd Int Conf on Intelligent Sensors, Sensor Networks and Information, p.425-430. https://doi.org/10.1109/ISSNIP.2007.4496881

Silver D, Schrittwieser J, Simonyan K, et al., 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676):354-350. https://doi.org/10.1038/nature24270

Wang Q, Leng S, Fu HR, et al., 2012. An IEEE 802.11p-based multichannel MAC scheme with channel coordination for vehicular ad hoc networks. *IEEE Trans Intell Trans Syst*, 13(2):449-458. https://doi.org/10.1109/tits.2011.2171951

Wang W, Kwasinski A, Niyato D, et al., 2017. A survey on applications of model-free strategy learning in cognitive wireless networks. *IEEE Commun Surv Tutor*, 18(3):1717-1757. https://doi.org/10.1109/COMST.2016.2539923

Xu ZY, Wang YZ, Tang J, et al., 2017. A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs. IEEE Int Conf on Communications, p.1-6. https://doi.org/10.1109/ICC.2017.7997286

Yau KLA, Komisarczuk P, Paul DT, 2010. Enhancing network performance in distributed cognitive radio networks using single-agent and multi-agent reinforcement learning. IEEE Local Computer Network Conf, p.152-159. https://doi.org/10.1109/LCN.2010.5735689

Ye H, Li GY, and Juang BHF, 2018. Deep reinforcement learning based resource allocation for V2V communications. IEEE Int Conf on Communications, p.1-6. https://doi.org/10.1109/ICC.2018.8422586