Multi-Agent Deep Reinforcement Learning Based Resource Allocation for Heterogeneous QoS Guarantees for Vehicular Networks

Jie Tian, *Member, IEEE,* Qianqian Liu, Haixia Zhang, *Senior Member, IEEE,* and Dalei Wu, *Senior Member, IEEE*

***Abstract*—Vehicle to vehicle communications can offer direct information interaction, including security-centered information and entertainment information. However, the rapid proliferation of vehicles and the diversity of communications services demand for a more intelligent and efficient resource allocation framework to enhance network performance. In this paper, a multi-agent deep reinforcement learning based resource allocation framework is developed to jointly optimize the channel allocation and power control to satisfy the heterogeneous Quality of Service (QoS) requirements in heterogeneous vehicular networks. In the proposed framework, the utility maximization problem is formulated by considering two types of traffics, i.e., the strict ultra-reliable and low latency requirements for safety-centric applications and the high-capacity requirements for entertainment applications. The utility of each vehicular users is formulated as a multi-criterion objective function by taking into account the heterogeneous traffic requirements. To overcome the drawbacks of the traditional totally centralized and distributed deep reinforcement learning based resource allocation approaches, we propose a multi-agent deep deterministic policy gradient algorithm with centralized learning and decentralized execution to solve the formulated optimization problem. The normalization of the input states and reward functions is introduced to speed up the training and learning progress of the proposed algorithm. Simulation results show the superiority of the proposed algorithm in terms of the convergence and system performance through the comparison with the other methods and schemes for the delay-sensitive applications and delay-tolerant applications.**

***Index Terms*—Resource allocation, deep reinforcement learning, multi-agent deep deterministic policy gradient (MADDPG), heterogeneous applications.**

I. Introduction

Vehicular communications networks have emerged as a promising technology to improve road safety, traffic efficiency, and entertainment experience in future 5G and intelligent transportation systems (ITS) [1]. Especially, heterogeneous vehicular communication networks have drawn much more attention from both industry and academia [2]. In heterogeneous vehicular networks, cellular networks are capable of providing wide coverage and quality of service (QoS) guarantee for vehicular users, which is essential to achieve reliable vehicle-to-vehicle (V2V) communication for fast-moving vehicles. On the other hand, V2V links provide direct communication with other neighbor vehicles by reusing the uplink radio resources of cellular user equipments. In addition, the V2V communications can improve the spectral efficiency and data transmission rate, while substantially decreasing latency and energy consumption. Therefore, it could support the heterogeneous services with different QoS requirements. In heterogeneous vehicular networks, the communication services can be broadly classified into two types, i.e., delay-sensitive applications service and delay-insensitive applications service. The delay-sensitive applications aim to guarantee the vehicular traffic safety by sharing safety-critical messages among VUs, and such safety messages often has strict high reliable and low latency requirements [3], [4]. The delay-insensitive applications usually provides VUs's general entertainment communication services, which involves amount of data transfer and thus demands higher data rate without strict reliability and latency demands [5]. Therefore, an effective resource allocation scheme is needed to support both types of services to the subscribers with limited bandwidth resources.

Up to now, there are numerous works on the resource management in wireless networks. Historically, the dominated approaches to resource allocation and optimization is through mathematical programming to optimize the design criteria of interest, for example, sum rate maximization or delay minimization while imposing some constraints on the remaining. However, most of the formulated optimization problems for wireless resource allocation are strongly non-convex or mixed integer nonlinear programming (MINLP) [6]–[11]. There have been no known algorithms reported that can solve the problems optimally with a polynomial time complexity, and thus most of the existing works pursue suboptimal solutions or some heuristic algorithms without any performance guarantee. Moreover, these suboptimal approaches are also computationally complex and hard to be executed in real time, and the performance of these methods also highly depends on the accuracy of the models. Nonetheless, as the wireless networks

J. Tian and Q. Liu are with School of Information Science and Engineering, Shandong Normal University, China. H. Zhang is with Shandong Provincial Key Laboratory of Wireless Communication Technologies, Shandong University, China and is also with School of Control Science and Engineering, Shandong University, China. D. Wu is with Department of Computer Science and Engineering, the university of Tennessee, USA. (e-mail: tianjie@sdnu.edu.cn; lqianstudent@gmail.com; haixia.zhang@sdu.edu.cn; dalei-wu@utc.edu. *Corresponding author: Haixia Zhang*).

This work was supported in part by the Project of International Cooperation and Exchanges NSFC (No. 61860206005), the National Natural Science Foundation of China (No. 61801278 and 61972237), the Key Laboratory of Cognitive Radio and Information Processing , Ministry of Education (Guilin University of Electronic Technology)(No. CRKL190205), and Shandong Provincial scientific research programs in colleges and universities (J18KA310).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089823, IEEE Internet of Things Journal

become increasingly diverse and complex, the aforementioned traditional design approaches face stringent challenges. For example, the vehicular networks environment is highly dynamic especially with the uncertainty in network model parameters, e.g., channel state information and network state information, leading to solutions with inferior performance. Especially, for such diverse QoS requirements in heterogeneous vehicular networks, the traditional optimization methods are hard to satisfy the service demands. Therefore, how to design a more intelligent and flexible resource allocation framework for vehicular communications has become an important issue in vehicular networks.

Recently, data-driven learning methods have been potential enabling techniques for future wireless communications, which have been proven to be effective in dealing with large-scale problems associated with wireless resource allocation [12]–[15]. Unlike the traditional mathematical methods, reinforcement learning (RL) can address the bottlenecks of modeling difficulties and high computational complexity. Moreover, in high mobility vehicular networks, RL can perceive these changes through interaction with the environment and make decisions. Therefore, it can be leveraged to solve complex optimization problems in vehicular networks. Nonetheless, the application of RL to solve the resource allocation in vehicular networks also has some limitations. Q-learning [16] [17] is the most common RL method, which can work well if the scale of the optimization problem is small. However, in the heterogeneous vehicular networks, the state and action spaces are very large and it is still challenging to find the optimal policy by looking up the Q-value table. With the help of the deep neural network (DNN), deep reinforcement learning (DRL) has shown impressive improvement in resource management field [18]–[20]. Moreover, centralized DRL is usually not feasible in vehicular network, since it is often impractical to obtain the complete channel state information (CSI) for a dynamic vehicular network, and the delay incurred by huge information interactions makes it unsuitable for real-time application. Moreover, the complexity of the centralized schemes increases with the increase of the number of vehicles, and thus results in enormous computational cost. In addition, a fully distributed DRL is likely to lead to a local optimum. Therefore, an effective resource allocation framework through combining the advantages of both the centralized and the distributed DRL methods is crucial for satisfying the heterogeneous QoS requirements in vehicular networks.

Motivated by the above analysis, this paper focuses on developing a more intelligent and efficient resource allocation framework for heterogeneous services in vehicular networks. To meet the heterogeneous services requirements of vehicular users, we develop a novel intelligent resource allocation scheme for vehicular communications networks based on the DRL theory. To improve the system performance and reduce the complexity, a multi-agent deep deterministic policy gradient method with centralized learning and decentralized execution is proposed to jointly optimize the channel allocation and power control.

The main contributions of the paper are summarized in the following aspects:

1) This paper focuses on heterogeneous QoS provisioning for V2V communications and two types of network traffic, i.e., from delay-sensitive applications (safety-related applications) and delay-insensitive applications (non-safety-related applications), respectively, have been taken into account. To satisfy the heterogeneous QoS requirements and meanwhile guaranteeing the QoS of all the cellular users, an intelligent resource management framework is developed to jointly optimize channel allocation and power control for heterogeneous cellular and vehicular communication networks.
2) In the proposed resource allocation framework, a distributed multi-agent DRL algorithm, i.e., multi-agent deep deterministic policy gradient (MADDPG), is proposed to satisfy both the ultra-reliable low latency requirements for safety-related applications and the high-capacity requirements for non-safety related applications.
3) The proposed multi-agent DRL resource allocation framework is based on centralized learning and decentralized execution, which combines the advantages of value function-based and policy search-based reinforcement learning methods to improve the speed of convergence and the effect of learning. Through centrally training the MADDPG model offline, the vehicles act as learning agents, then can rapidly make resource allocation decisions during the online execution stage. In addition, we also introduce a method to speed up the training and learning progress by normalizing the input states and reward functions.
4) Simulation results show the superiority of the proposed scheme in terms of the average throughput and the transmission rate performance through the comparison with existing reinforcement learning based algorithms and resource allocation schemes.

The reminder of this paper is structured as follows. In Section II, we present the related work. In Section III, the system model and the problem formulation are presented. In section IV, the joint resource allocation algorithm based on multi-agent deep deterministic policy gradient is described detailedly. In Section V, the simulation results are provided to evaluate the performance of the proposed algorithms. Finally, the conclusions are drawn in Section VI.

II. Related Work

There have been lots of resource allocation schemes for the wireless networks. For instance, in [13], [21]–[24], the coalition game and learning based algorithms were adopted to solve the channel allocation problem in heterogeneous cellular networks rather than heterogeneous vehicular networks. Authors in [25] developed a position-based joint cell association and resource block allocation to satisfy the ultra-reliable and low-latency requirements of the uplink for automated vehicles. In [26], to ensure the driving safety, a spectrum resource allocation scheme for alarm information delivery in V2V communication was proposed and the priority levels of vehicles were taken into account. In [27], a heterogeneous spectrum sharing algorithm was proposed to improve the capacity of V2V links by leveraging the millimeter wave and cellular spectrum. But all of the above work does not consider the power control for V2V communications. Authors in [28]–[31] introduced power allocation schemes and proved that the power allocation schemes outperformed the scheme with fixed power allocation obviously. A joint transmit power and resource allocation framework for safety-related service with constraints of ultra-reliable and low-latency in vehicular communication was proposed in [15], [20]. Considering the packet retransmission, the authors in [1] performed spectrum and power allocation for safety-critical information exchange in vehicular communication. However, the aforementioned work only focuses on one type of service and ignores the heterogeneity of QoS requirements for V2V communications services.

Deep reinforcement learning (DRL) as a subfield of machine learning has received significant attentions and particularly has been proposed as another powerful optimization tool to solve the resource management problems in vehicular networks. In fact, compared to traditional mathematical methods, the use of DRL allows to operate in more complex and challenging vehicular networks with higher performance and efficiency. The authors in [32] used DRL to solve the channel allocation in a centralized manner. In [18], authors investigated a decentralized resource allocation mechanism for both unicast and broadcast scenarios in V2V communication based on DQN. In [19], a joint problem of transmission mode selection, RB allocation and power control for cellular V2X communication was formulated to maximize the sum capacity of V2I users while ensuring the latency and reliability requirement of V2V pairs. Meanwhile, a spectrum and power allocation scheme for V2V communication has been developed in [33] based on DQN approach. Unfortunately, the traditional reinforcement learning approaches such as Q-learning or policy gradient are poorly suited to multi-agent environments, since each agent's policy is changing as training progresses, and the environment becomes non-stationary from the perspective of any individual agent [34]. In order to deal with the instable multi-agent environment, the authors in [35] designed a common reword for all agents. However, the above methods have another major limitation that they cannot handle the optimization of continuous variables well.

To address the above problem, the multi-agent deep deterministic policy gradient (MADDPG) algorithm adopting the framework of centralized training with decentralized execution is proposed for a discrete or continuous variable space in a multi-agent environment based on the Actor-Critic and Policy Gradient algorithms [34], [36], [37]. For the MADDP algorithm, every agent can achieve an individual performance according to its utility and a near-optimal solution within a limited number of iterations in a relatively stable environment. The authors in [38] investigated base station association and channel selection in vehicular network by using MADDPG. Based on the MADDPG algorithm, [39] and [40] investigated the cell association and resource allocation for underwater networks and UAV communication networks. However, to the best of our knowledge, the MADDPG based methods has not been studied for heterogeneous QoS guarantee in vehicular networks.

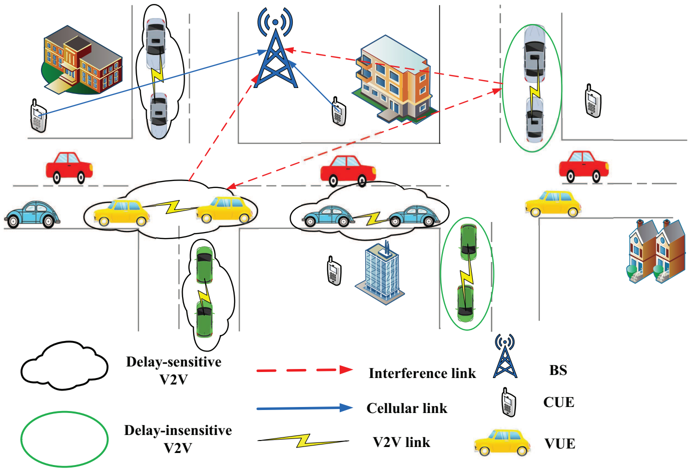


Fig. 1. An illustration of heterogeneous vehicular networks

III. System Model and Problem Formulation

In this section, we first introduce a heterogeneous vehicular network architecture and modeling the heterogeneous QoS requirement for V2V communications, and then present the problem formulation.

A. System Model

We consider a heterogeneous vehicular and cellular network in this paper as illustrated in Fig.1. The base station (BS) is located at the center of the area and there are $\mathcal{M} = \{1, 2, \cdots, M\}$ cellular users (CUs) connecting with the BS on M shared orthogonal channels, and $\mathcal{K} = \{1, 2, \cdots, K\}$ pairs of V2V users (VUs) and each pair VUs consists of one V2V receiver and one V2V transmitter directly communicating with each other. Moreover, we assume a single antenna for each CUs and VUs. The VUs have heterogeneous services with different communication requirements. Specifically, the safety-critical message exchange supported by V2V links requires strict ultra-reliable and low-latency communication, and the non-safety-related service such as the entertainment information sharing via V2V communication requires high rate transmission. To improve the spectrum utilization efficiently, it is assumed that VUs share the uplink channels of the CUs. We assume that M CUs operate over the orthogonal channels where $\mathcal{F} = \{1, 2, \cdots, F\}, |F| = |M|$. Thus there is no co-channel interference among CUs. Each VU can only occupy a single channel for uplink transmission at one slot. Denote the binary channel association vector for the kth VU as $c_k^f = \{c_k^1, c_k^2, \cdots, c_k^F\}, i \in \mathcal{K}, f \in \mathcal{F}$. When the kth VU utilizes the channel f, $c_k^f = 1$, otherwise $c_k^f = 0$. Moreover, each VU can only access to no more than one channel, i.e.,

$$\sum_{f=1}^{F} c_k^f \leq 1, \forall k \in \mathcal{K}, f \in \mathcal{F}. \tag{1}$$

Without loss of generality, the Rayleigh small scale fading is considered in the channel modeling and we also consider a nonsingular path loss model [1]. Let g_k^f denote the channel gain caused by the small scale fading for the kth VU on the channel f, and let $\widehat{g}_k = (1 + d_k^{-\alpha})^{-1}$ represent the path loss, where d_k denotes the distance between the transmitter and receiver for the kth VU, and α represents the path loss exponent. Then, the instantaneous signal-to-interference-plus-noise ratio (SINR) for the kth VU using channel f is give by

$$\Gamma_k^f = \frac{c_k^f p_k g_k^f \widehat{g}_k}{\sum_{m=1}^{M} c_m^f p_m g_m^f \widehat{g}_{mk} + \sum_{\acute{k}\neq k}^{K} c_{\acute{k}}^f p_{\acute{k}} g_{\acute{k}}^f \widehat{g}_{\acute{k}k} + \sigma^2}, \quad (2)$$

where p_k and p_m are the transmission powers of the kth VU and the mth CU, respectively. $\widehat{g}_{mk}$ denotes the path loss from CU m to the receiver of the VUs k. σ^2 is the noise power. The interference achieved by the kth VUs is from other VUs and CUs reusing the same channel f.

Similarly, the SINR of the mth CU using the channel f can be expressed as

$$\Gamma_m^f = \frac{c_m^f p_m g_m^f \widehat{g}_m}{\sum_{k=1}^{K} c_k^f p_k g_k^f \widehat{g}_{mk} + \sigma^2}, \quad (3)$$

where $\widehat{g}_m$ denotes the path loss for CU m, $\widehat{g}_{mk}$ denotes the path loss from the transmitter of the VUs k to CU m, and the interference for CUs only comes from the VUs sharing the same channel f.

B. Heterogeneous QoS requirements for V2V communications

In heterogeneous vehicular and cellular networks, there exists different types of vehicular services with different communication requirements for V2V communication links. The communication services can be broadly classified into two types, i.e., delay-sensitive applications service and delay-insensitive applications service [18]. The delay-sensitive applications aim to guarantee the vehicular traffic safety by sharing safety-critical messages among VUs, and such safety messages often has strict high-reliability and low-latency requirements. The delay-insensitive applications usually provide VU's general entertainment communication services, which involve high-volume data transfer and thus demands higher data rate without strict reliability and latency demands [6]. The utilities for the delay-sensitive applications and delay-insensitive applications are modeled respectively as below.

1) Utility of delay-sensitive applications: In light of the stringent constraints of ultra low latency and high reliability for the delay-sensitive applications, throughput is utilized to model the utility function for the VUs by jointly considering the delay violation probability caused by queueing delay and the transmission error probability incurred by the imperfect channel state.

The queuing theory is utilized to model the packet delay violation probability incurred by exceeding the maximization queueing delay. Similar to [42], we assume that the packet arrival rate of the kth VU follows a Poisson distribution with the parameter λ_k (packets per second), which has been widely used to model the traffic arrival process in wireless communication. In addition, the data packets serving time is assumed to follow the exponential distribution, which conforms to the general law. Therefore, the queueing process of the data packets arriving at the sender is modeled as M/M/1 queuing model. The maximum sojourn time in a queue allowed for the transmitter of VU k is denoted as D_k^{th}. The packet will be dropped when the average waiting time $T_{k,f}$ in a queue exceeds D_k^{th}. To reduce the delay, no retransmission is considered. The probability of a packet being discarded due to exceeding the average delay threshold, denoted by P_k^{dly} can be calculated as follows

$$P_{k,f}^{dly} = Pb(T_{k,f} > D_k^{th}) = \exp\{-(\frac{(R_{k,f} - \lambda_k l_{ave}) D_k^{th}}{l_{ave}})\}, \quad (4)$$

where l_{ave} represents average packet length (number of bits), $R_{k,f} = W \log_2(1 + \Gamma_k^f)$ is the achievable transmission rate of VU k on the channel f.

Considering the co-channel interference, we assume that when the SINR value Γ_k^f for VU k on channel f is lower than a given threshold Γ_k^{th}, the packet can not be correctly decoded at the receiver side and the transmission error occurs. Then the transmission error probability of the kth VU, denoted by $P_{k,f}^{err}$, is expressed as

$$P_{k,f}^{err} = Pb(\Gamma_k^f < \Gamma_k^{th}). \quad (5)$$

Therefore, the successful transmission probability for the kth VU on channel f is denoted as $P_{k,f}^{S,suc}$,

$$P_{k,f}^{S,suc} = 1 - (P_{k,f}^{dly} + (1 - P_{k,f}^{dly}) P_{k,f}^{err}). \quad (6)$$

The throughput, i.e., the number of successfully transmitted packets, is utilized to model the utility for the delay-sensitive applications, denoted by U_k^d, which is formulated as follows

$$U_k^S = \sum_{f \in \mathcal{F}} \lambda_k c_k^f P_{k,f}^{S,suc}. \quad (7)$$

2) Utility of delay-tolerate applications: Considering the high data rate requirement of the delay tolerance service, the utility function for this type of applications can be modeled by the transmission rate. Therefore, the transmission rate of the kth VU transmitting the delay-tolerate traffic in the channel f is expressed as

$$U_k^{T,f} = R_k^f = W \log_2(1 + \Gamma_k^f), \quad (8)$$

where W denotes the bandwidth for each channel. The utility of the kth VU is expressed as

$$U_k^T = \sum_{f \in \mathcal{F}} c_k^f U_k^{T,f}. \quad (9)$$

C. Problem Formulation

In this paper, our objective is to find the optimal joint channel allocation and power control scheme to maximize the utility of each vehicular user while guaranteeing the QoS requirements of all CUs in the heterogeneous vehicular network. We assume that each VU can possess heterogeneous

[1] In the nonsingular path loss model, $(1 + d_k^{-\alpha})^{-1}$, the path loss tends to be one as the propagation distance tends to zero, whereas in singular path model, $d^{-\alpha}$, the path loss tends to infinity [41].

applications which are either delay-sensitive traffics or delay-tolerant traffics. Hence, we define the utility function of the VUs as a multi-criterion objective function by considering the aforementioned two types of traffic requirements, i.e.,

$$U_k = \theta_k U_k^S + (1-\theta_k) U_k^T. \tag{10}$$

where θ_k is binary heterogeneous application selection factor for the kth VU. Without loss of generality, we assume that in a given slot, each VU only has one kinds of traffic type. If the kth VU deliveries the delay-sensitive applications, $\theta_k = 1$, otherwise $\theta_k = 0$. Moreover, in heterogeneous vehicular and cellular networks, the QoS requirements of all CUs can not be ignored and the minimum QoS constraints of the CUs should be satisfied. The power control and channel selection variables can be denoted by the vectors $\boldsymbol{P} = (P_k)_{k \in \mathcal{K}}$, and $\boldsymbol{C} = (c_k^f)_{k \in \mathcal{K}, f \in \mathcal{F}}$, respectively. Based on the above analysis, the optimization problem for any VU $k, k \in \mathcal{K}$ is formulated as

$$\max_{\boldsymbol{P},\boldsymbol{C}} \quad U_k(c_k^f, p_k) \tag{11}$$

$$\text{s.t.} \quad \sum_{f=1}^{F} c_m^f \Gamma_m^f \geq \Gamma_m^{th}, \forall m \in \mathcal{M}, f \in \mathcal{F}, \tag{12}$$

$$\sum_{f=1}^{F} c_k^f \leq 1, \forall k \in \mathcal{K}, f \in \mathcal{F}, \tag{13}$$

$$P_{k,f}^{dly} \leq P_{th}^{dly}, \forall k \in \mathcal{K}, f \in \mathcal{F}, \tag{14}$$

$$P_{k,f}^{err} \leq P_{th}^{dly}, \forall m \in \mathcal{K}, f \in \mathcal{F}. \tag{15}$$

where the constraint in (12) denotes the SINR constraint for cellular user which also represents the minimum QoS requirements of the CUs, the constraint in (13) indicates that each VU can only access to no more than one channel, and (14) and (15) denotes the constraints of the delay violation probability and the error transmission probability, respectively.

The formulated optimization problem in (11)-(15) is a nonlinear and non-convex problem, which is difficult to solve within a polynomial-time. Moreover, the channel selection and power control strategies of each VU are interrelated and influence each other due to the interference. Especially in the highly time-varying heterogeneous vehicular networks with heterogeneous traffics, the dynamic channel condition and network environment make it more complicated and challenging to solve the optimization problem using traditional optimization methods.

IV. Multi-Agent DRL For Resource Allocation

In this section, we demonstrate how to use the multi-agent deep reinforcement learning (MADRL) method within the framework of centralized learning and decentralized execution to solve the above formulated problem. In Subsection IV-A, we firstly model our proposed multi-agent joint optimization problem as a Markov decision process (MDP) by defining agent, state, action, and reward, respectively. Subsection IV-B presents the preliminaries of the proposed MADDPG algorithm. In subsection IV-C, the MADDPG algorithm for solving the above optimization problem is presented detailedly.

A. A Markov decision process for joint resource allocation

In the proposed dynamic and time-varying heterogeneous scenario, the VUs are defined as the agents, and each agent learns a policy that directs its best actions on the channel selection and power control to maximize its utility function. Since each agent's decision will be affected by other agents in the multi-agent environment, the extended MDP [43] is used to model multi-agent reinforcement learning. The MDP for K agents can be defined as a tuple $(\mathcal{S}; \mathcal{A}; \mathcal{R}; \mathcal{P}_{s\acute{s}})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ denotes the rewords and $\mathcal{P}_{s\acute{s}}$ is the transition probability. More specifically, at each time slot t, the VU as a agent, observes the current state s_t of the environment and then takes an action a_t according to a policy $\pi(a|s)$. As a benefit, the agent will obtain a reward r_t and transfer to a new state $\acute{s}$ (s_{t+1}). Three key elements of the MDP model, i.e., state, action and reward, are defined respectively as follows.

1) State Space: As mentioned in section III-B, the heterogeneous services have different utility functions. Therefore, the state should exhibit the VUs's binary heterogeneous application by incorporating selection factor, θ_k. According to Section III-C, the utility function (throughput or transmission rate) is related to the power and channel conditions of each agent. Thus, the binary channel-association vector c_k^f and the power p_k should be taken into consideration in the state. In addition, considering the QoS requirements of VUs and CUs, the achieved interference situation by the kth VU, I_k, and as well as the channel and power status information of both the remaining VUs and its neighbor CUs, denoted by the vector $\boldsymbol{C}_{k \in \mathcal{K}/k}, \boldsymbol{P}_{k \in \mathcal{K}/k}, \boldsymbol{N}_k$, respectively, are also taken into account in the state space. Hence the state of each VUs k can be expressed as

$$\mathbf{s_k} = \{\theta_k, I_k, \boldsymbol{C}_{k \in \mathcal{K}/k}, \boldsymbol{P}_{k \in \mathcal{K}/k}, \mathbf{N_k}\} \tag{16}$$

and the state space for the multi-agent environment can be formulated as $\mathcal{S} = \{\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_K}\}$.

Notice that the state includes discrete and continuous variables, and the dimensions of the vector $\mathbf{N_K}$ is different for different VU k. Considering that the neural networks are sensitive to the scales and the distribution of their inputs, proper normalization is critical for training [44]. In order to facilitate training, we have carried out min-max normalization before the state is fed into the learning model as below,

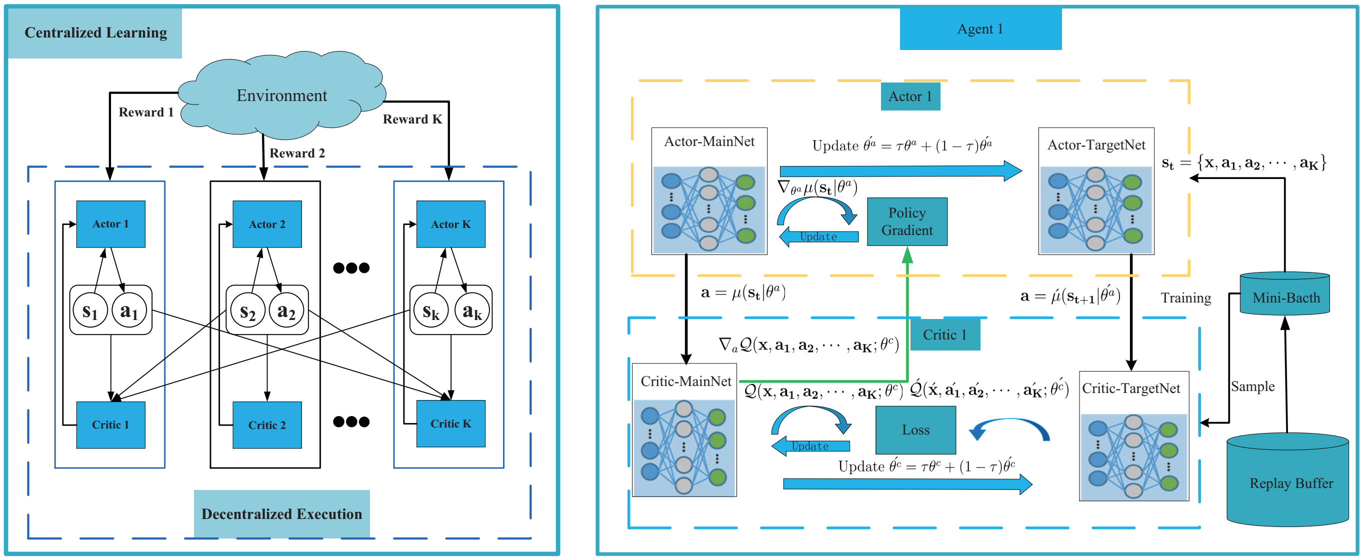
$$\mathbf{s_k} = \frac{\mathbf{s_k} - min_{\mathbf{s_k}}}{max_{\mathbf{s_k}} - min_{\mathbf{s_k}}}, \tag{17}$$

where $min_{\mathbf{s_k}} = \min(\mathbf{s_k})$, and $max_{\mathbf{s_k}} = \max(\mathbf{s_k})$.

2) Action Space: The VU as an agent decides the action in every time slot, i.e., which channel to be utilized and how much power to be set for communication. Thus the action of each VU k can be defined as

$$\boldsymbol{a}_k = \{c_k^f, p_k\} \tag{18}$$

and the action space can be define as $\mathcal{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_K\}$. As we can see, the action includes discrete binary channel-association variable c_k^f and continuous power control variable


Centralized Learning
Environment
Reward 1
Reward 2
Reward K
Actor 1
Actor 2
Actor K
s_1
a_1
s_2
a_2
s_k
a_k
Critic 1
Critic 2
Critic K
Decentralized Execution
Agent 1
Actor 1
Actor-MainNet
Update $\acute{\theta^a} = \tau\theta^a + (1-\tau)\acute{\theta^a}$
Actor-TargetNet
$\mathbf{s_t} = \{\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}\}$
$\nabla_{\theta^a}\mu(\mathbf{s_t}|\theta^a)$
Policy Gradient
Update
$\mathbf{a} = \mu(\mathbf{s_t}|\theta^a)$
$\mathbf{a} = \acute{\mu}(\mathbf{s_{t+1}}|\acute{\theta^a})$
Training
Mini-Bacth
Critic 1
$\nabla_a\mathcal{Q}(\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}; \theta^c)$
Critic-MainNet
$\mathcal{Q}(\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}; \theta^c)$
$\acute{\mathcal{Q}}(\acute{\mathbf{x}}, \acute{\mathbf{a_1}}, \acute{\mathbf{a_2}}, \cdots, \acute{\mathbf{a_K}}; \acute{\theta^c})$
Critic-TargetNet
Sample
Loss
Update
Update $\acute{\theta^c} = \tau\theta^c + (1-\tau)\acute{\theta^c}$
Replay Buffer

(a)

(b)

Fig. 2. Framework of MADDPG algorithm for our multi-agent network. (a) the framework of MADDPG (b) the details of the neural network architecture for an agent

p_k. As the number of VU increases, the action space exponentially grows so that it is intractable to solve the joint resource allocation problem with traditional approaches.

3) Reward Design: The learning process is driven by the reward function in DRL framework, and each agent makes decision to maximize its own long-term accumulative reward by interacting with the environment. As mentioned in section III-B, for delay-sensitive services, the agent's goal is to maximize the throughput U_k^S, and for delay-tolerant services, the agent's goal is to maximize its transmission rate U_k^T. Therefore, the reward function should be designed based on the above goals, i.e., the objectives of the original formulated problems. In addition, the QoS constraint of CU in (12) is represented as a penalty. When it does not meet the QoS requirements of CUs, the reward function is set to be zero. Thus, the reward of each VU consists of two parts, i.e., utility function U_k and the QoS penalty, and can be expressed as

$$r_k = \begin{cases} \theta_k U_k^S + (1-\theta_k)U_k^T & \sum_{f=1}^{F} c_m^f \Gamma_m^f \geq \Gamma_m^{th}, \forall m \in \mathcal{M} \\ 0 & \text{else} \end{cases} \tag{19}$$

and the reward space is $\mathcal{R} = \{r_1, r_2, \cdots, r_K\}$.

B. Preliminaries of MADDPG

In this subsection, we first briefly introduce the single-agent DDPG method based on the actor-critic (AC) framework, which includes an actor network and a critic network. The critic network evaluates the action-value function under the actor policy. More specifically, for each agent, the input of its actor network is its own state $\mathbf{s}$, and the output is its deterministic action $\boldsymbol{a}$. On the other hand, the inputs of its critic network include both the state $\mathbf{s}$ and action $\boldsymbol{a}$ of all the agent generated by the actor network, and the output of its critic network is estimated by the Q-value function.

The critic network of DDPG is a value function evaluation approach similar to the evaluation network in the DQN. The implementation of the critic network is to use deep neural network (DNN) to estimate the action state value function $\mathcal{Q}^\pi(s,a) = E\{R(s,a)\}$ where $R(s,a) = \sum_{t=0}^{\infty}\gamma^t r_t$ is the accumulated reward, which can be computed recursively by using the Bellman equation,

$$\mathcal{Q}^\pi(s,a) = E_{s_{t+1}}\{r_t(s_t,a_t) + \gamma E_{a_{t+1}\sim\pi}\{\mathcal{Q}^\pi(s_{t+1},a_{t+1})\}\} \tag{20}$$

where π is the map policy of action and state, r_t is the immediate reward function, γ is the discount factor, which means current state $\mathbf{s_t}$ is affected by the later states.

Critic network learns the action-value function $\mathcal{Q}$ corresponding to the optimal policy by minimizing the loss

$$Loos(\theta^c) = E[\mathcal{Y} - \mathcal{Q}(s_t, a_t; \theta^c)^2], \tag{21}$$

and

$$\mathcal{Y} = r_t + \gamma\acute{\mathcal{Q}}(s_{t+1}, \acute{\mu}(s_{t+1}|\theta^a), \acute{\theta^c}) \tag{22}$$

where the target network $\acute{\mathcal{Q}}(s_{t+1}, a_{t+1}, \acute{\theta^c})$ is used to improve the stability of the DNN nonlinear estimator. The target network is used to calculate the updated target and has the same architecture as the evaluate network. It is worth noting that this value-based DRL (DQN) based on value function is not suitable for optimizing continuous variables, so it cannot be directly used to solve the joint optimization problem we formulated.

The actor network of DDPG is a policy search-based approach, and its objective is to choose an optimal action based on the deterministic policy $a = \mu(s, \theta^a)$. The main idea of the actor network of DDPG is to directly adjust the parameters θ^a of the policy in order to maximize the objective $J(\theta^a) = E\{R(s,a)\}$ by taking steps in the direction of $\nabla_{\theta^a} J(\theta^a)$.

$$\nabla_{\theta^a} J(\theta^a) \approx E[\nabla_a \mathcal{Q}(s_t, a_t = \mu(s_t), \theta^c)\nabla_{\theta^a}\mu(s_t, \theta^a)], \tag{23}$$

where the gradient consists of two parts: the gradient of action value function from critic ($\nabla_a \mathcal{Q}(s_t, a_t = \mu(s_t))$ and its own deterministic strategy action gradient ($\nabla_{\theta^a}\mu(s_t, \theta^a)$).

Note that the deterministic policy-based DRL can optimize continuous variables. In addition, when the number of VU increases, the action space becomes larger, and the search space of PG becomes larger, which will cause huge computational cost. Unlike the PG algorithm, DDPG will not fail with the explosion of the action space, so it is suitable to solve the optimal problem with high-dimensional actions in multi-agent scenarios.

After the above analysis, it seems feasible that the single DDPG can be directly applied to solve our multi-agent optimization problem by letting each agent independently learn its own Q-value function. However, the direct use of DDPG on each agent/user is a completely distributed. Each agent makes independent decisions without considering the influence of other agents, so it will lead to a local optimum, and the environment can appear to be non-stationary from the perspective of each agent. In order to address the above problems, we utilize the MADDPG that learns a Q-value function for each agent based on the global information to solve our joint optimization problem in a multi-agent scenario. For MADDPG, since all the action and state of agents are known during the training stage, the environment is stationary even the policy changed. That is

$$\mathcal{P}(\mathbf{s_{t+1}}|\mathbf{s}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}, \pi_1, \pi_2, \cdots, \pi_K) \quad (24)$$

$$= \mathcal{P}(\mathbf{s_{t+1}}|\mathbf{s}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}) \quad (25)$$

$$= \mathcal{P}(\mathbf{s_{t+1}}|\mathbf{s}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}, \acute{\pi_1}, \acute{\pi_2}, \cdots, \acute{\pi_K}) \quad (26)$$

for any $\pi_i \neq \acute{\pi_i}$. The details of MADDPG algorithm are described in the next section.

C. MADDPG for resource allocation

In this section, we first introduce the framework of MADDPG algorithm, and then describes in detail how to solve our joint optimization problem with the proposed MADDPG algorithm. Lastly, we analyze the computational complexity of the proposed algorithm.

1) MADDPG framework: As shown in Fig.2, the MADDPG framework is composed of the environment and K agents, where each agent has two phases: the centralized training and decentralized execution. It is worth noting that the training phase is offline, and the exploration is needed to search the optimal policy. While in the execution phase, only forwards propagation without random exploration process, which consumes much less resources than training. Besides, there is no need for exploration during the execution phase. Next, we take an agent as an example to explain how to centrally train the MADDPG model and execute the learned model in a decentralized way.

In the centralized offline training phase, the critic network calculates the centralized action-value function $\mathcal{Q}(\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}; \theta^c)$, based on global state information including all agents' actions and observations. The centralized $\mathcal{Q}$ function evaluates the actor's actions from a global perspective, and uses this to guide the actor to choose better action. Then, the critic network update the parameters θ^c by minimizing the loss:

$$Loos(\theta^c) = E[\mathcal{Q}(\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}; \theta^c) - \mathcal{Y}^{\mathcal{MADDPG}})^2] \quad (27)$$

where

$$\mathcal{Y}^{MADDPG} = r_t + \gamma \acute{\mathcal{Q}}(\acute{\mathbf{x}}, \acute{\mathbf{a_1}}, \acute{\mathbf{a_2}}, \cdots, \acute{\mathbf{a_K}}; \acute{\theta^c}) \quad (28)$$

where $\mathbf{x} = (\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_K})$, θ^c is the parameter of evaluation network. While $\acute{\mathbf{x}} = (\acute{\mathbf{s_1}}, \acute{\mathbf{s_2}}, \cdots, \acute{\mathbf{s_K}})$ indicates the updated states for the target network, and $\acute{\theta^{\mathbf{c}}}$ is the parameter of evaluation network.

At the same time, the actor network updates network parameters θ^a and outputs actions $\mathbf{a}$ based on the centralized $\mathcal{Q}$ function calculated by the critic and its own observation information. Specifically, the actor network is directly adjusting the network parameters θ^a at the direction of $\nabla_{\theta^a} J(\theta^a)$, which is given by

$$\nabla_{\theta^a} J(\theta^a) \approx E[\nabla_a \mathcal{Q}(\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}; \theta^c) \nabla_{\theta^a}\mu(\mathbf{s}, \theta^a)] \quad (29)$$

where θ^a is the parameter of actor network.

In the stage of decentralized execution, the critic network is not involved, only the trained actor network works online. The actor network outputs actions according to its own state. During this execution, there is only a forwards propagation process and no random exploration process, which greatly reduces computing resources and time compared to the training phase. In the actor with well-trained parameters, each agent can obtain an action close to the global optimum without aware of other agents' information.

$$\mathbf{a_k} = \mu(\mathbf{s_k}, \theta^a) \quad (30)$$

where s_k is the observation of the agent k, and the θ^a is the well trained network parameter. The detailed description of the decentralized execution is shown in Algorithm 2.

2) MADDPG algorithm design: According to the above discussion and the framework as shown in Figure 2, the offline centralized training algorithm of the MADDPG for the proposed multi-agent environment is described in Algorithm 1 and the online decentralized execution algorithm is summarized in Algorithm 2.

In Algorithm 1, to train the network better, the centralized offline training algorithm sets the S episodes and resets the environment before the start of each episode. For example, the locations of VUs and the type of heterogeneous services requested, and the locations and channel occupancy of CUs, etc. In each episode, there are L loop and in each step each agent selects an action $\mathbf{a}$ (channel and power) according to the observed state $\mathbf{s}$, then executes the action, gets a reward, and then enters the next state. Then transition $(\mathbf{s}, \mathbf{a}, r, \acute{\mathbf{s}})$ is stored into replay buffer D. When the number of experiences is greater than the set mini-batch sampling size N, start training the critic and actor networks and soft updating parameters of the target network in both critic and actor networks according to the following procedure,

$$\acute{\theta^a} \leftarrow \tau\theta^a + (1-\tau)\acute{\theta^a} \quad (31)$$

$$\acute{\theta^c} \leftarrow \tau\theta^c + (1-\tau)\acute{\theta^c} \tag{32}$$

where τ is the soft update factor improving the stability of learning, and $\{\theta^a, \theta^c, \acute{\theta^a}, \acute{\theta^c}\}$ is the set of the network parameters of an agent. Note that each agent owns different network parameter set $\{\theta^a, \theta^c, \acute{\theta^a}, \acute{\theta^c}\}$. Finally, the parameters $\{\theta^a_1, \theta^a_2, \cdots, \theta^a_K\}$ of the trained actor network for all agents are obtained as the output.

Algorithm 2 is the detailed description of the online decentralized execution. It is an online algorithm that requires very little computational cost and latency. After loading the trained actor network parameters, each agent only needs to input the current states, and then outputs the optimal action $\mathbf{a}$.

3) Computational Complexity of MADDPG algorithm: The computational complexity of the proposed Algorithm 1 and Algorithm 2 are analyzed in the following, respectively.

For easy of decription, we let N^a_l denote the number of the neurons in the lth layer of the actor network. Since the actor network is fully connected, the computational complexity of the lth layer can be written as $O(N^a_{l-1}N^a_l + N^a_l N^a_{l+1})$. The computational complexity of the actor network is $O(\sum_{l=2}^{L-1}(N^a_{l-1}N^a_l + N^a_l N^a_{l+1}))$, where L is the number of layers in the actor network. The critic network is also fully connected. Define the number of neurons the hth layer in the critic network as N^c_h, the computational complexity of the hth layer is $O(N^c_{h-1}N^c_h + N^c_h N^c_{h+1})$. The computational complexity of the whole critic network is $O(\sum_{h=2}^{H-1}(N^c_{h-1}N^c_h + N^c_h N^c_{h+1}))$, where H is the number of layers in the critic network.

In Algorithm 1, the actor and critic networks of all agents are being trained at the same time and they are all extracting N experiences from the replay buffer for back-propagation training. Therefore, the complexity of Algorithm 1 is $O(N_u * L * S * N * (\sum_{l=2}^{L-1}(N^a_{l-1}N^a_l + N^a_l N^a_{l+1})) + \sum_{h=2}^{H-1}(N^c_{h-1}N^c_h + N^c_h N^c_{h+1}))$, where N_u is the number of vehicle agents, L is the max training steps of each episode, S is the number of episodes, and N is mini-batch sampling size. In Algorithm 2, for each agent, only the actor network works online, so the complexity of Algorithm 2 is $O(N_u * L * \sum_{l=2}^{L-1}(N^a_{l-1}N^a_l + N^a_l N^a_{l+1}))$.

V. Simulation Results

In this section, we provide the simulation results to demonstrate the performance of the proposed MADDPG resource allocation method.

A. Simulation Settings

We consider a single cell during the simulation where the vehicle and cellular users are distributed following the spatial Poisson process and a BS is located at the center with the radius $500m$. The radio resource is organized in a number of orthogonal uplink channels with $180kHz$ per channel, and each V2V transmitter communicates with its intended receiver that is located in an arbitrary direction with maximum distance $10\sqrt{2}m$. The latency threshold for delay sensitive traffic is set to be 60ms and the outage threshold is set to be be 10 dB, while the minimum SINR threshold of all the cellular users are set to be 3 dB. The average incoming packet rate λ for the VUs is varied from 20 to 180, and represents the low, medium and high load conditions of the network, respectively. In the proposed MADDPG, we design the actor and critic networks with one input layer, one output layer, one hidden layer of 100 units and Adam optimization algorithm for training. We assume that there are 50 steps in each episode, and the environment is updated at the beginning of each episode. We update the actor and critic networks with the learning rate of 0.001, and 0.002, respectively. The detailed simulation parameters are summarized in Table I.

Algorithm 1 Centralized Traning Algorithm using MADDPG for Joint Resource Allocation

```
Input :
Training episode numbers $S$, training steps $L$;
discount factor $\gamma$, the soft update factor $\tau$;
replay buffer $D$, mini-batch sampling size $N$;
critic net learning rate $\alpha^c$, actor net learning rate $\alpha^a$;
Gaussian distributed behavior $noise$ with the average value $n_0$, behavior noise decay factor $k$;
VUES numbers $K$, CUES numbers $M$.
Output : Actor networks' weights for each agent
Initialize : Actor network $\mu$, critic network $\mathcal{Q}$ with weights $\theta^a$ and $\theta^c$ for each agent
for episode=1 to S do
  Receive an initial state space $\mathcal{S}_0$ from the reset environment
  for step=1 to L do
    each VU selects an action $\mathbf{a} = \mu(\mathbf{s}, \theta^a)$ by observing the state $\mathbf{s}$, executes action, then gets a reward $r$, transforms to new state $\acute{\mathbf{s}}$. Store transition $(\mathbf{s}, \mathbf{a}, r, \acute{\mathbf{s}})$ into $D$
    if step $\geq$ N then
      for VUES k=1 to K do
        Sample a random minibatch of $N$ transition samples $(\mathbf{s}, \mathbf{a}, r, \acute{\mathbf{s}})$ from $D$
        set $\mathcal{Y}^{MADDPG}$
        $= r_t + \gamma\acute{\mathcal{Q}}(\acute{\mathbf{x}}, \acute{\mathbf{a_1}}, \acute{\mathbf{a_2}}, \cdots, \acute{\mathbf{a_K}}; \acute{\theta^c_k})$
        Update critic network by minimizing the loss function
        $Loos(\theta^c_k) = E[\mathcal{Q}(\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}; \theta^c_k) - \mathcal{Y}^{\mathcal{MADDPG}})^2]$
        Update the actor network by using the policy gradient $\nabla_{\theta^a} J(\theta^a_k)$
        $\approx E[\nabla_a \mathcal{Q}(\mathbf{x}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K}; \theta^c_k)\nabla_{\theta^a}\mu(\mathbf{s}, \theta^a_k)]$
        Soft updates parameters of the target network in both critic and actor network
        $\acute{\theta^a_k} \leftarrow \tau\theta^a_k + (1-\tau)\acute{\theta^a_k}$
        $\acute{\theta^c_k} \leftarrow \tau\theta^c_k + (1-\tau)\theta^c_k$
      end for
    end if
  end for
end for
```

The proposed method (denoted as MADDPG) is compared with the following reinforcement learning based baselines:

1) DQN: The ordinary reinforcement learning algorithm,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089823, IEEE Internet of Things Journal

Algorithm 2 Decentralized Execution Algorithm using MADDPG for Joint Resource Allocation

```
Input : Test steps $L$, current state $\mathbf{s}$ for each agent, online actor network structure $\mu_1, \mu_2, \cdots, \mu_K$
Output : Action $\mathbf{a}$ for each agent
Initialize : Load the actor networks' weights $\theta_1^a, \theta_2^a, \cdots, \theta_K^a$
for step=1 to L do
    Each agent selects an action $\mathbf{a} = \mu(\mathbf{s}, \theta^a)$ based on the current state $\mathbf{s}$
    Update the environment based on the actions selected $(\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_K})$
    Each agent gets reward $r$, transforms to new state $\acute{\mathbf{s}}$
end for
```

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Bandwidth	180kHz
Power of CU	10dBm
Power of VUs	[15,20,25]dBm
Maximum distance of V2V link	$10\sqrt{2}m$
Path loss exponent α	3
Noise power σ^2	-114dBm
Average packet length	4000 bits
Packet arrival rate	50ps
Latency constraints for VUs	60ms
SINR threshold of VUs Γ_k^{th}	3dB
SINR threshold of VUs γ_m^{th}	10dB
The average incoming packet rate λ	[20 180]
Actor network learning rate l_a	0.001
Critic network learning rate l_c	0.002
Discount factor γ	0.95
Memory pool capacity D	10000
Mini-batch size K	32

which is suitable for optimizing discrete variables and is an independently trained single-agent method [33] .

2) DuDQN: Comparing with DQN algorithm, Dueling DQN is an improved reinforcement learning algorithm by spliting the state action-value function $Q^\pi(s,a)$ into a state-value function $V^\pi(s)$ and an action advantage function $A^\pi(s,a)$, i.e., $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$. This neural network structure can use the state-value function $V^\pi(s)$ to evaluate the quality of the state, and use the action advantage function $A^\pi(s,a)$ to evaluate the quality of an action. However, it is still an independently trained single-agent algorithm.

3) Fix Power scheme: In this scheme, the power of the VUs are fixed and only optimize the channel selection variable [31]. The proposed MADDPG resource allocation algorithm is still used to verify the effect of continuous power optimization.

To further evaluate the performance of the proposed algorithm, we simulate two common scenarios: one is that the VUs generate the delay-sensitive traffic. Another is that the VUs demand for delay-insensitive entertainment information. In the above two scenarios, we have surveyed the convergence performance and the network performance in terms of their utility, i.e., the average throughput and the average rate, respectively.

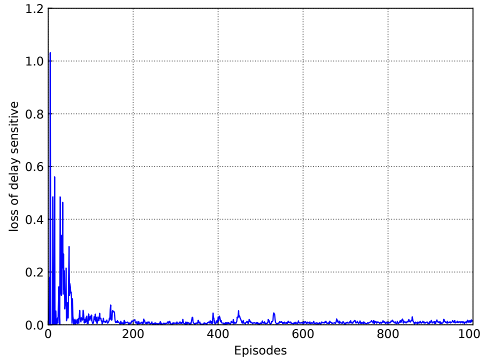


Fig. 3. The learning process of MADDPG in delay sensitive scenarios

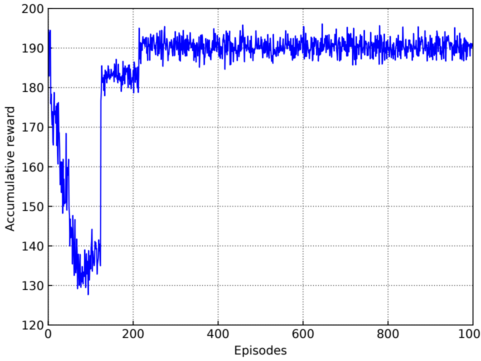


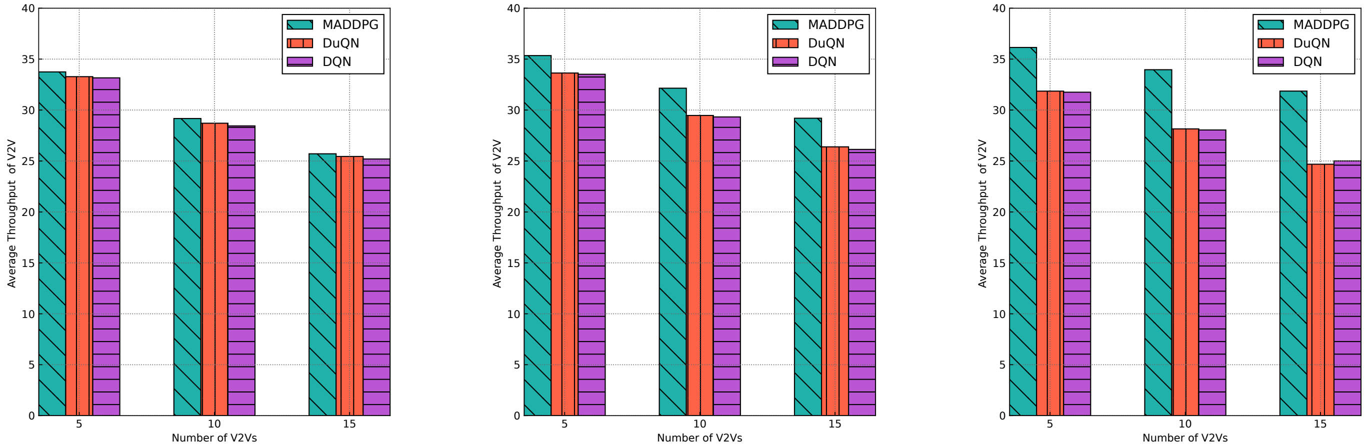
Fig. 4. Convergence performance of accumulative reward for all VUs in delay sensitive scenarios

B. Delay sensitive traffic

In this scenario, all the VUs transmit the delay sensitive information such as the security-centric information, resulting in communication requirements for ultra-high reliability and ultra-low latency.

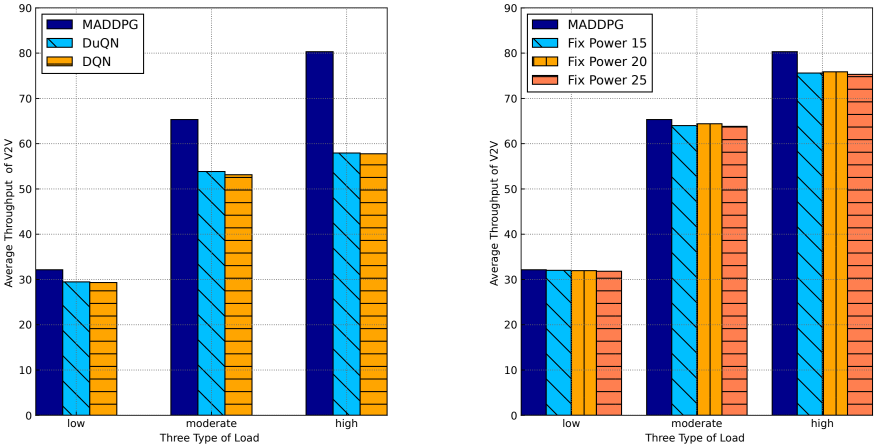
1) Convergence performance: The convergence performance of our proposed MADDPG algorithm is shown in Fig.3 and Fig.4, in the case of 8 VUs and 5 CUs.

Fig.3 shows the learning process of one randomly selected VU in terms of the loss function performance. It can be observed that the value of the loss function is high at the beginning of learning process. When training to about 200 episodes, the loss value declines to a very small and relatively stable value. This means that our approach achieves a good performance with fast convergence speed and stable learning process with less fluctuations. This is due to the normalization of the input states (including continuous and discrete variables) and reward functions. There is no doubt that this normalization is conducive to speed up the training and learning progress, so in the next simulation, we train only 200 epsides (10000 steps).



(a) (b) (c)

Fig. 5. Average throughput performance comparison among three algorithm versus different number of VUs with low level offered traffic load in the network. (a) Three frequency channels (b) Five frequency channels (c) Ten frequency channels.



(a) (b)

Fig. 6. Average throughput performance comparison among three algorithms and four schemes in the case of three kinds of load levels, low : $\lambda \in [20\ 60]$, moderate: $\lambda \in [60\ 140]$ /s, and high: $\lambda \in [140\ 180]$ /s. (a) Three algorithms (b) Four schemes

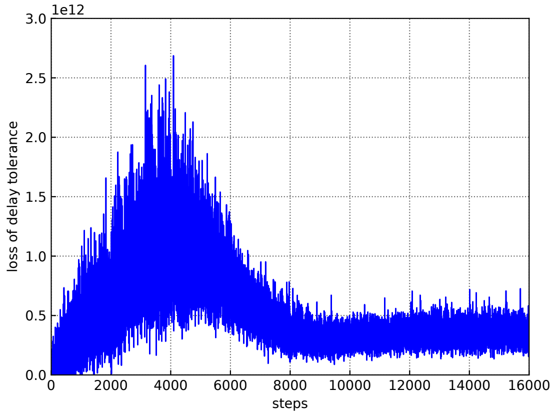
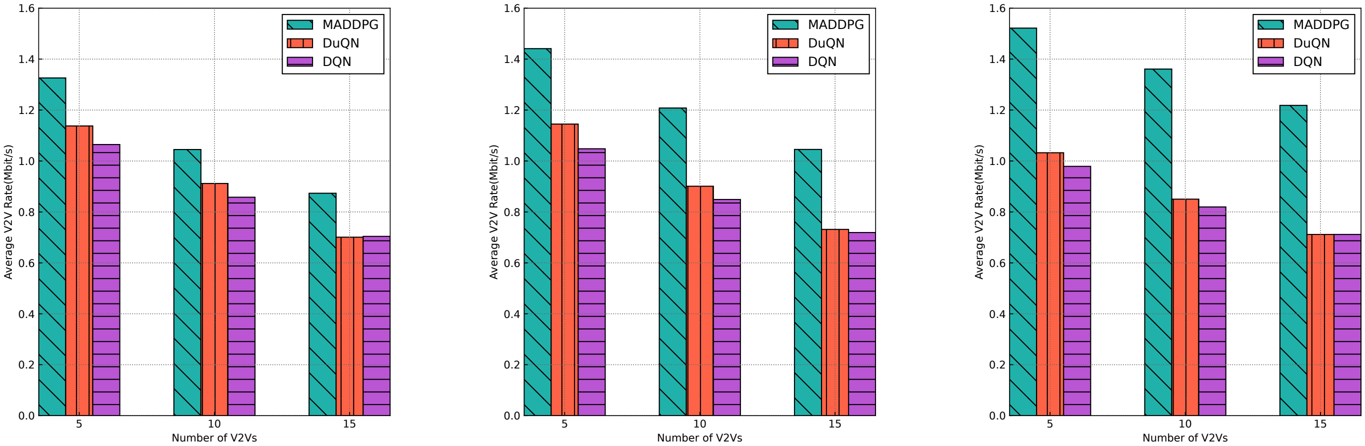


Fig. 7. Convergence performance of loss function in delay-tolerant scenarios

In Fig.4, we plot cumulative reward of all VUs. The average incoming packet rate λ is randomly selected from the interval $[20, 60]$, which represents the low load traffics in the considered heterogeneous vehicular network. Due to the existence of the exploration strategy, the curve has oscillated, but according to Fig.3, it converges at about 200 episodes, so the cumulative reward begins to stabilize after 200 episodes.

2) System performance: In the following, the system performance in terms of the average throughput of all VUs for delay-tolerant traffic is investigated.

Fig.5 represents average throughput performance comparison among three algorithms versus different number of VUs. During the simulation, the average packet rate of each VU is selected randomly from $[20\ 60]$, which results in a network with low traffic load. We can see that our proposed MADDPG algorithm achieves the best average throughput performance.

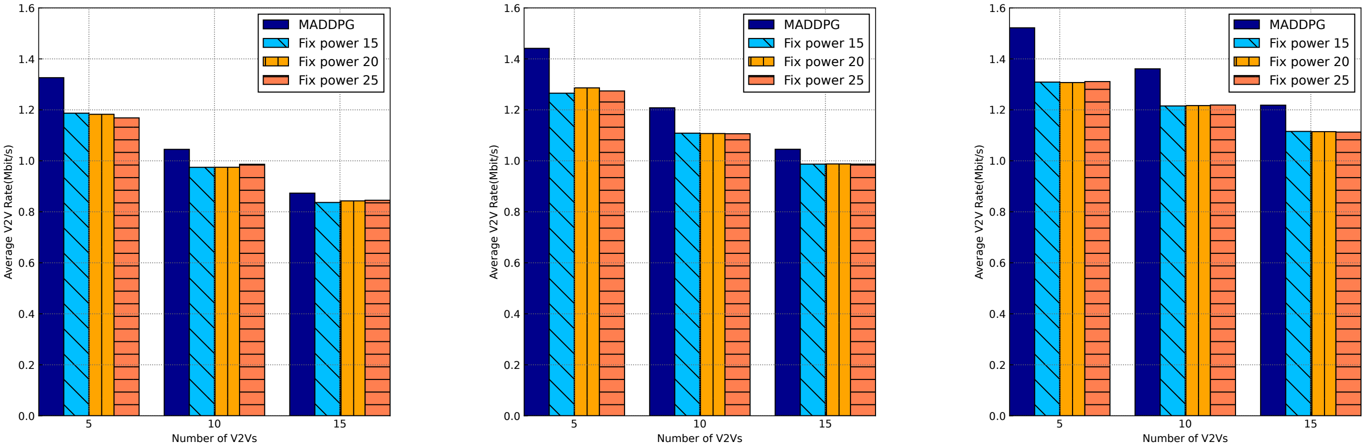


(a)

(b)

(c)

Fig. 8. Average rate performance comparison among three algorithms in the case of different number of VUs in the network. (a) Three frequency channels (b) Five frequency channels (c) Ten frequency channels.



(a)

(b)

(c)

Fig. 9. Average rate performance comparison among four schemes in the case of different number of VUs in the network. (a) Three frequency channels (b) Five frequency channels (c) Ten frequency channels.

For example, Fig.5 (b) shows that the average throughput decreases as the number of the VUs increases, because under the moderate limited radio resource, the large number of VUs will incur greater interference among each other. Additionally, Fig.5 shows that along with the increase in channel number, the user can achieve a higher average throughput. This is because more channels represents more wireless spectrum resources, so each VU can better meet the communication needs of low latency and high reliability, thus achieving higher throughput. Especially, as shown in Fig.5 (a), where resources are most limited and the interference is greatest (15 VUs on 3 channels), all approaches fail to achieves a satisfying throughput. However, the proposed MADDPG algorithm still achieves a much larger average throughput through effectively resource management policy. This is because agents can cooperate with each other in our proposed multi-agent DDPG algorithm, while other single-agent DRL is trained independently without cooperation.

Fig.6 shows the average throughput performance comparison among three algorithms and four schemes with different load levels, 10 VUs and 5 channels. As shown in Fig.6 (a), compared with the other two algorithms, the proposed MADDPG algorithm can achieve better performance under any kind of load level. Especially, for the case of high load level, our algorithm has a higher average throughput. Because our proposed MADDPG algorithm has the characteristics of centralized training and distributed execution, where in the training phase, the resource allocation strategy can be achieved by learning the policy of other users and coordinating with each other. Fig.6 (b) presents the impact of power control methods. Compared with the fixed power scheme, our scheme can achieve better system performance.

C. Delay tolerant traffic

For the vehicle to vehicle communication, all the VUs could share delay-tolerant entertainment information over the V2V link with high rate transmission. In such case, we also verify the performance of our proposed algorithm in terms of two aspects, i.e., convergence performance and system performance.

1) Convergence performance: Fig.7 demonstrates the convergence performance through the loss function for delay tolerant traffic and it shows the average transmission rate varies along with the training steps. The large amount of loss change is because that the delay tolerant VUs directly maximize the transmission rate without normalization processing. As the training steps increases, the loss starts to decrease and then converges to a relative stable value. Moreover, it can be seen that the proposed algorithm converges at approximately 10000 steps (i.e., 200 episodes), which also shows that our proposed approach has good convergence performance.

2) System performance: Fig.8 illustrates the average V2V transmission rate versus the number of VUs in case of three type channels (i.e.,3, 5, 10), respectively. From Fig.8, it can be seen that along with the increase of channels, the average rate of the VUs also rises, while as the increase of the number of the VUs, the average rate of the VUs decreases. In addition, compared with other algorithms and schemes, the proposed algorithm still has much better performance through properly wireless resource allocation to mitigate the interference.

To further evaluate the performance of our proposed joint optimization scheme, we also compare our proposed scheme with a channel-only optimization scheme with fixed power. The results are shown in Fig.9. During the simulation, the power of the fixed power resource allocation schemes are set to be 15dBm, 20dBm, and 20dBm. To fairly comparison, resource allocation are conducted by the proposed MADDPG algorithm under the cases of different number of vehicles in case of three type channels (i.e., 3, 5, 10). From Fig.9, it can been observed that the performance of our proposed scheme always outperforms that of the channel-only optimization scheme under different network conditions.

VI. Conclusions

In this paper, we have investigated the joint channel allocation and power control problem for heterogeneous QoS guarantee in the heterogeneous vehicular and cellular networks. A resource allocation framework based on multi-agent deep reinforcement learning has been developed to solve the above resource allocation problem for V2V communication link. The utility of the vehicular users has been formulated as a multi-criterion objective function by taking into account the heterogeneous traffic requirements. In order to support high reliable delay-sensitive and high rate delay-tolerance vehicle applications, we proposed a multi-agent deep deterministic policy gradient algorithm with centralized learning and decentralized execution to maximize the utilities of vehicular users while guaranteeing the QoS requirements of all CUs in the network. To speed up the training and learning procedure of the proposed algorithm, the input states and reward functions has been normalized. Extensive simulation has been conducted and the results show the superiority of the proposed algorithm in terms of the convergence and system performance through the comparison with the other reinforcement learning methods and resource allocation schemes for both the delay-sensitive applications and delay-tolerant applications.

References

[1] C. Guo, L. Liang, and G. Y. Li, “Resource allocation for high-reliability low-latency vehicular communications with packet retransmission,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6219–6230, Jul. 2019.

[2] Z. Xiao, X. Shen, F. Zeng, V. Havyarimana, D. Wang, W. Chen, and K. Li, “Spectrum resource sharing in heterogeneous vehicular networks: A noncooperative game-theoretic approach with correlated equilibrium,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9449–9458, Oct. 2018.

[3] J. Mei, K. Zheng, L. Zhao, Y. Teng, and X. Wang, “A Latency and Reliability Guaranteed Resource Allocation Scheme for LTE V2V Communication Systems,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3850–3860, Jun. 2018.

[4] L. Wang, R. F. Iida, and A. M. Wyglinski, “Performance analysis of multichannel edca-based V2V communications via discrete event system,” in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Sep. 2019, pp. 1–5.

[5] L. Liang, H. Ye, and G. Y. Li, “Toward intelligent vehicular networks: A machine learning framework,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 124–135, Feb. 2019.

[6] L. Liang, H. Ye, G. Yu, and G. Y. Li, “Deep-learning-based wireless resource allocation with application to vehicular networks,” *Proceedings of the IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[7] N. Wang and S. Xu, “Resource Allocation for LTE-Based Heterogeneous Vehicular Network in Unlicensed Bands,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6.

[8] H. Yang, L. Zhao, L. Lei, and K. Zheng, “A two-stage allocation scheme for delay-sensitive services in dense vehicular networks,” in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 1358–1363.

[9] W. Huang, L. Ding, D. Meng, J. Hwang, Y. Xu, and W. Zhang, “QoE-based resource allocation for heterogeneous multi-radio communication in software-defined vehicle networks,” *IEEE Access*, vol. 6, pp. 3387–3399, 2018.

[10] L. Liang, S. Xie, G. Y. Li, Z. Ding, and X. Yu, “Graph-based resource sharing in vehicular communication,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4579–4592, Jul. 2018.

[11] C. He, Q. Chen, C. Pan, X. Li, and F. Zheng, “Resource allocation schemes based on coalition games for vehicular communications,” *IEEE Communications Letters*, vol. 23, no. 12, pp. 2340–2343, Dec. 2019.

[12] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, “Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches,” *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, 2020.

[13] N. Zhao, Y. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, “Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.

[14] H. Ye and G. Y. Li, “Deep reinforcement learning for resource allocation in v2v communications,” in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[15] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, “Distributed federated learning for ultra-reliable low-latency vehicular communications,” *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.

[16] Y. Luo, Z. Shi, X. Zhou, Q. Liu, and Q. Yi, “Dynamic resource allocations based on Q-learning for d2d communication in cellular networks,” in *2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing(ICCWAMTIP)*, Dec. 2014, pp. 385–388.

[17] K. Zia, N. Javed, M. N. Sial, S. Ahmed, A. A. Pirzada, and F. Pervez, “A Distributed Multi-Agent RL-Based Autonomous Spectrum Allocation Scheme in D2D Enabled Multi-Tier HetNets,” *IEEE Access*, vol. 7, pp. 6733–6745, 2019.

[18] H. Ye, G. Y. Li, and B. F. Juang, “Deep Reinforcement Learning Based Resource Allocation for V2V Communications,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089823, IEEE Internet of Things Journal

[19] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep Reinforcement Learning Based Mode Selection and Resource Allocation for Cellular V2X Communications," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6380–6391, Jul. 2020.

[20] M. Chen, J. Chen, X. Chen, S. Zhang, and S. Xu, "A deep learning based resource allocation scheme in vehicular communication systems," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2019, pp. 1–6.

[21] Y. Chen, B. Ai, Y. Niu, K. Guan, and Z. Han, "Resource allocation for device-to-device communications underlaying heterogeneous cellular networks using coalitional games," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4163–4176, Jun. 2018.

[22] Y. Liu, Y. Wang, R. Sun, and Z. Miao, "Distributed resource allocation for D2D-assisted small cell networks with heterogeneous spectrum," *IEEE Access*, vol. 7, pp. 83 900–83 914, 2019.

[23] A. H. Arani, A. Mehbodniya, M. J. Omidi, and F. Adachi, "Learning-based joint power and channel assignment for hyper dense 5G networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[24] Z. Lu and M. C. Gursoy, "Dynamic channel access and power control via deep reinforcement learning," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Sep. 2019, pp. 1–5.

[25] L. Ding, Y. Wang, P. Wu, and J. Zhang, "Position-Based User-Centric Radio Resource Management in 5G UDN for Ultra-Reliable and Low-Latency Vehicular Communications," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[26] B. Li, D. He, Y. Feng, Y. Xu, and H. Zheng, "Spectrum Resource Allocation Scheme for Alarm Information Delivery in V2V Communication," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug 2018, pp. 1–5.

[27] Y. Xuan, C. Guo, C. Feng, and Z. Li, "Multi-graph based spectrum sharing scheme in vehicular network with integration of heterogenous spectrum," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[28] G. Liu, Z. Wang, J. Hu, Z. Ding, and P. Fan, "Cooperative NOMA Broadcasting/Multicasting for Low-Latency and High-Reliability 5G Cellular V2X Communications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7828–7838, Oct. 2019.

[29] K. K. Nguyen, T. Q. Duong, N. A. Vien, N. Le-Khac, and L. D. Nguyen, "Distributed Deep Deterministic Policy Gradient for Power Allocation Control in D2D-Based V2V Communications," *IEEE Access*, vol. 7, pp. 164 533–164 543, 2019.

[30] C. Liu and M. Bennis, "Ultra-Reliable and Low-Latency Vehicular Transmission: An Extreme Value Theory Approach," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1292–1295, Jun. 2018.

[31] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, Aug. 2019.

[32] S. Gyawali, Y. Qian, and R. Q. Hu, "Resource allocation in vehicular communications using graph and deep reinforcement learning," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6.

[33] L. Liang, H. Ye, and G. Y. Li, "Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.

[34] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 2017, pp. 6379–6390.

[35] L. Liang, H. Ye, and G. Y. Li, "Multi-Agent Reinforcement Learning for Spectrum Sharing in Vehicular Networks," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jul. 2019, pp. 1–5.

[36] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31th International Conference on Machine Learning*, Beijing, China, 2014.

[37] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[38] D. Kwon and J. Kim, "Multi-agent deep reinforcement learning for cooperative connected vehicles," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6.

[39] Y. Zhang, Z. Zhuang, F. Gao, J. Wang, and Z. Han, "Multi-agent deep reinforcement learning for secure UAV communications," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–5.

[40] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multi-agent DDPG-Based Deep Learning for Smart Ocean Federated Learning IoT Networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9895–9903, Oct. 2020.

[41] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.

[42] J. Tian, H. Zhang, D. Wu, and D. Yuan, "QoS-constrained medium access probability optimization in wireless interference-limited networks," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1064–1077, 2018.

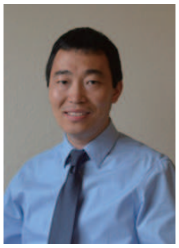
[43] M. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *in Proceedings of the Eleventh International Conference*, New Brunswick, Jul. 1994.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

Jie Tian (S'12-M' 16)) received the B.E. and M.E. degrees from Shandong Normal University, China, in 2008 and 2011, respectively, and the Ph.D. degree in communication and information systems from the School of Information Science and Engineering, Shandong University, China, in 2016. She is currently an Associate Professor with the School of Information Science and Engineering, Shandong Normal University, Jinan, China. Her current research interests include intelligent radio resource management, industrial Internet of Things (IIoT), and mobile edge computing.She is an Associate Editor for the *International Journal of Communication Systems*.

Qianqian Liu is currently working toward the B.E. degree in Communication Engineering, Shandong Normal University, Jinan, China. She has won the first class scholarship of the university three times. Her research interests include machine learning for wireless resource management in Heterogeneous Networks.

Haixia Zhang (M'08-SM'11) received the B.E. degree from the Department of Communication and Information Engineering, Guilin University of Electronic Technology, China, in 2001, and the M.Eng. and Ph.D. degrees in communication and information systems from the School of Information Science and Engineering, Shandong University, China, in 2004 and 2008, respectively. From 2006 to 2008, she was with the Institute for Circuit and Signal Processing, Munich University of Technology, as an Academic Assistant. From 2016 to 2017, she was a Visiting Professor with the University of Florida, USA. She is currently a Full Professor with Shandong University. Her current research interests include cognitive radio systems, cooperative (relay) communications, resource management, spacetime process techniques, mobile edge computing, and smart communication technologies. Dr. Zhang serves on editorial boards of the IEEE Internet of Things Journal, IEEE Wireless Communication Letters, and China Communication, She has been actively participating in many professional services. She has been serving as TPC member, session chair and invited speaker for conferences.



Dalei Wu (M' 11) received the B.S. and M.Eng. degrees in electrical engineering from Shandong University, Jinan, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer engineering from the University of NebraskaLincoln, Lincoln, NE, USA, in 2010.

He is an Associate Professor with the Department of Computer Science and Engineering at the University of Tennessee at Chattanooga (UTC), Chattanooga, TN, USA. Prior to joining UTC, he was a Postdoctoral Research Associate with the Mechatronics Research Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include data-driven intelligent systems, cyber-physical systems, and complex dynamic system modeling and optimization.