Decentralized Resource Allocation-Based Multiagent Deep Learning in Vehicular Network

Armeline D. Mafuta, *Member, IEEE*, Bodhaswar T. J. Maharaj, *Senior Member, IEEE*, and Attahiru S. Alfa

***Abstract*—Resource allocation (RA) has a significant impact on vehicular network performance. With high mobility, RA is more challenging, as the number of vehicles in close proximity changes dynamically in the nonstationary environment. In this article, we propose a multiagent double deep Q-networks scheme to stabilize the system and maximize the sum-capacity of the vehicle-to-infrastructure (V2I) links, while satisfying the reliability and delay constraints for vehicle-to-vehicle (V2V) links. To avoid interference caused by unstable V2V links, a transmission mode selection is considered in the scheme design. In addition, we introduce a binarized weight algorithm to accelerate the deep neural network learning process and, therefore, improve the computational complexity of our scheme. Through extensive simulations and complexity analysis, we demonstrate that the proposed scheme yields excellent performance in terms of the sum-rate and probability rate of V2I and V2V communication modes. We also compare the proposed scheme with binarized weights with other algorithms in terms of accuracy evaluation.**

***Index Terms*—Binarized weights, deep reinforcement learning (DRL), Markovian decision process (MDP), multiagent scheme, resource allocation (RA), vehicular communication.**

I. Introduction

A VEHICULAR network is an enabling technology for autonomous driving and smart vehicles, capable of providing various on-board data services [1]. It is a key technology that enhances transportation by supporting cooperation among vehicles in the immediate vicinity, providing satisfactory quality of service (QoS). The intelligent transportation systems (ITS) [2] and the dedicated short-range communications (DSRC) [3], both based on IEEE 802.11p standard, have been studied to realize vehicular communications. A technique based on IEEE 802.11p intervehicle cooperation channel estimation has been proposed in [4] to obtain the accurate channel state information (CSI) in vehicle-to-everything (V2X) networks to improve the safety-critical data transmission. However, defective aspects of IEEE 802.11p, such as mobility management, scalability, and guaranteed QoS have been mentioned in [5] and [6]. This is basically due to its link and physical layers being designed for low mobility communications. This issue is addressed with the 3GPP standard that supports various QoS requirements of V2X networks and exploits the device-to-device (D2D) communication in long term evolution (LTE) and 5G cellular networks. We, therefore, focus on the resource allocation (RA) in vehicular network based on 3GPP standard, which comprises vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) links sharing frequency spectrum and using PC5 and Uu radio interfaces, respectively.

The most common communication modes in a vehicular network are the V2V, vehicle-to-pedestrian (V2P) and V2I links [7]. Usually, these links have different QoS constraints. For instance, V2I links focus primarily on the data sum-rate, whereas V2P and V2V links are more concerned with reliability and delay requirements. It is foreseen that in the future more traffic-related applications and entertainment will be undertaken by vehicles. Even currently, multiple vehicular applications are already delivered through the V2V and V2I communication links [8]. However, this requires frequent and unlimited Internet access provided through high-capacity V2I links and safety-critical messages being transmitted via V2V communications to neighboring vehicles instantaneously and in a reliable way.

In view of the number of inherent limitation factors, such as hostile wireless channels, a progressively congested spectrum, rapid growth of vehicular communication devices, and especially high mobility, it is very important to use and allocate the available resources in an extremely effective way. Conventional RA approaches cannot be used in V2V networks for D2D communication even with the assumption of full CSI, as it would be difficult to monitor the variation of the channel on a small time-scale. Rigorous mathematical methodologies for vehicular communication systems traditionally developed are mainly based on assumptions of low mobility or static environments. Generally, they are not modeled to deal with the different environmental conditions effectively [9]. It is, therefore, necessary to develop new schemes that can interact with a rapidly changing environment, in terms of RA, and obtain optimal decisions for high mobility vehicular systems.

Fortunately, deep reinforcement learning (DRL) models have been introduced into V2V networks for high efficiency in complex and big data problems, such as RA to handle decision-making challenges under uncertainty. However, DRL requires much memory and computation time to extract nonlinear feature

Manuscript received 9 August 2021; revised 30 November 2021 and 9 March 2022; accepted 16 March 2022. Date of publication 13 May 2022; date of current version 24 February 2023. *(Corresponding author: Armeline D. Mafuta.)*

Armeline D. Mafuta is with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: u20808781@tuks.co.za).

Bodhaswar T. J. Maharaj is with Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: sunil.Maharaj@up.ac.za).

Attahiru S. Alfa is with Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada, and also with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: attahiru.alfa@umanitoba.ca).

Digital Object Identifier 10.1109/JSYST.2022.3163235

vectors to predict outputs with high accuracy. Indeed, most of the V2V services-based DRL approaches for analyzing and processing the collected data from vehicles are performed in the cloud with high-performance resources. There are methods that could be applied to the training process to improve memory use and reduce the computation time of the DRL algorithms [10]. However, this may come at a price of accuracy loss. For instance, the BinaryConnect [11], BinaryNet [12], or XNOR-Net [13] and many others act like regulizers that use binary values for weights and activations instead of full precision values during training of deep neural networks (DNN) to reduce execution times. In this article, we address the computational complexity challenge of RA when considering DNN training in vehicular networks.

II. Related Works

Recently, several studies have addressed the problem of smart RA in different vehicular network environments. Overview studies on resource management for a range of vehicular networks schemes are presented in [1], where challenges and opportunities of research are examined. Existing literature on RA can mainly be categorized into two approaches: 1) centralized and 2) decentralized schemes. To obtain general network information, centralized algorithms would experience a large transmission overhead such that each vehicle would have to transmit interference information and the local channel state to the central controller [14]–[16]. However, it is challenging for these centralized schemes to satisfy various QoS constraints precisely with ultralow end-to-end delay and high reliability. The high mobility scenario in these systems also prevent accurate full CSI from being collected at central controllers. As for the decentralized RA schemes, every V2V communication link should make its own decision with either little or partial knowledge of the other V2V pairs [17], [18].

Wilhelmi *et al.* [19] concentrated on a completely decentralized scenario for the RA where no information on neighboring nodes was available to the learners. The reliability and latency requirements, which are key metrics to the algorithm performance, are not discussed. We, however, consider these requirements in the design of the proposed scheme. A decentralized resource and power allocation algorithm is proposed for a multiuser OFDMA network in [20] and compared with centralized approaches.

Note that it is important for an efficient RA scheme to support various QoS requirements in vehicular networks, particularly when considering the V2V and V2I communication links. Hence, Zhang *et al.* [21] considered the DRL method for resource management in vehicle communication to optimize the V2I sum-capacity, while satisfying the reliability and delay constraints of V2V links. However, the development communication of the scheme in [21] is conducted with a small number of vehicles. This makes the scheme not efficient enough for the nonstationary environment. A multiagent DRL algorithm that maximizes the delivery rate of the V2V safety message is proposed in [22] for an in-coverage scenario, where multiple independent deep Q Network (DQN) parameters are trained for every V2V link. This scheme, however, does not scale computationally. In this article, the scaling problem is resolved by grouping all the vehicle's V2V decisions into a single decision. Moreover, only a single model is trained, then shared among the vehicles, allowing the vehicles to learn from the other's experiences. The allocation of resources in vehicular networks is also investigated in [23], where a deep deterministic policy gradient (DDPG) approach to optimize the sum-rate of the V2I link is used, meanwhile satisfying the delivery probability of the V2V link. However, the scheme cannot guarantee robust performance owing to the estimation error that results in catastrophic agent forgetting [24]. Note that in this article, we consider the problem with continuous state spaces and discrete action space. We, therefore, apply the DQN-based DRL framework rather than the DDPG algorithm and consider its extension, the DDQN, to avoid the overestimation problem of the DQN. In addition, the critical difference between the DDPG and DDQN algorithms lies in their training processes. For the DDPG, both the actor and critic networks have to be trained, while it is not the case for the DDQN. Also, the number of parameters to deal with in the DDPG is much higher and requires more computation than a DDQN.

In addition, most of the abovementioned research considered only V2V communication links to distribute the safety-critical messages among different vehicles. Nevertheless, using the V2V links only when considering high-dimensional spaces or large data, causes blockage effects due to the lower reliability of V2V links. Therefore, it results in poor performance of V2V communications [25]. This issue is addressed in this article by considering a mode selection technique that is a V2I-based forwarding method. In fact, to improve the spectrum utilization while satisfying the QoS constraints in the vehicular networks, the RA and mode selection should be jointly optimized [21].

The computational complexity of the DNN training procedure is not investigated in most of the above literature. However, it is crucial to address the complexity of the DNN in vehicular networks, since it is mostly high. Simons and Lee [10] summarized the study on different algorithms to reduce the computational time of the DNN training by binarizing the weights and/or activations in image processing, computer vision, etc. Cerutti *et al.* [26] explored the binary neural networks on sound-event detection in tight power-constrained Internet of Things devices. In this article we introduce the same process of binarizing the weights during DNN training to reduce the complexity of our proposed algorithm.

A. Contributions, List of Symbols and Paper Organization

1) Main Contributions: In this article, we consider the RA problem in high mobility vehicle communication systems, where multiple V2V pairs attempt to share the V2I preallocated orthogonal subbands. We present an RA scheme-based decentralized DRL approach to resolve the consequent challenges. The scheme aims at maximizing the capacity of V2I pairs, while satisfying the strict reliability and delay constraints on the V2V links for a periodic safety critical message sharing. The main contributions are as follows.

1) Markovian decision process (MDP) models the RA problem, where the V2V links are the multiagents that make

TABLE I
LIST OF SOME SYMBOLS AND NOTATIONS USED IN THE ARTICLE

Symbols	Description
$h_{m,B}^{\text{V2IU}}, h_k^{\text{V2VU}}, h_{k,B}^{\text{V2VU}}$	Channel information from the m^{th} V2IU to the BS, between the V2VU pairs k and from the k^{th} V2VU transmitter to the BS, respectively.
$g_{k,B}, g_{k,j}, g_{m,k}$	Interference channel gains from the k^{th} V2VU transmitter to the BS, from the k^{th} V2VU transmitter to the j^{th} V2VU receiver and from the m^{th} V2IU to k^{th} V2VU receiver
$\gamma_m^{\text{V2IU}}, \gamma_k^{\text{V2VU}}$	SINR of the m^{th} V2IU and k^{th} V2VU, respectively
$P_m^{\text{V2IU}}, P_k^{\text{V2VU}}$	Transmission powers of the m^{th} V2IU and k^{th} V2VU, respectively
$C_m^{\text{V2IU}}, C_k^{\text{V2VU}}, C_{min}^{\text{V2IU}}$	Capacities of the m^{th} V2IU, the k^{th} V2VU and the V2IU minimum capacity requirement, respectively

adaptive decisions according to the local observations. The framework capitalizes on the current progress of this multiagent approach to design the DRL-based decentralized algorithm that concurrently enhances V2V and V2I pairs' performance. The blockage effect in the communication that may occur is controlled by introducing a mode selection technique that consists of a V2I-based forwarding solution.

2) This article proposes an MA-DDQN scheme to avoid the overestimation problem of the conventional DQN approach. Moreover, the proposed scheme adopts both distributed execution and centralized training processes to ease the implementation and improve the stability of the system. This approach tackles the nonstationary environment by trying different joint actions and also reinforcing patterns of actions yielding better results.
3) Furthermore, to improve the time complexity of the proposed scheme, we propose to binarize the parameter weights during the DNN training process of the algorithm by using binarized weight values to train the model. This makes the model sizes of the DNN much smaller than usual and removes $\approx 2/3$ of the matrix multiplication operations of the DNN, which consequently reduces the computation time of the proposed MA-DDQN scheme.

2) List of Symbols: As scientific articles in this field use different notation variables and symbols, Table I provides the conventions applied in this article.

3) Organization: The rest of this article is organized as follows. A description of the system model is presented in Section III. Section IV presents the MA-DDQN-based decentralized algorithm solutions for RA with some basis on the RL method. We present the binarized weights approach and complexity analysis in Section V. The simulation analysis is discussed in Section VI and Section VII concludes this article.

III. SYSTEM MODEL DESCRIPTION

A. Network Architecture

A vehicular network consisting of a base station (BS) and several vehicles' user equipment (VUEs) is considered. As illustrated in Fig. 1, the BS is placed at the center of crossroads,

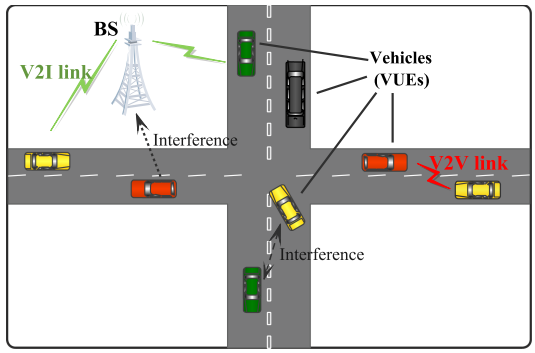

Fig. 1. Structure illustration of the vehicular network with V2I and V2V links.

whereas the VUEs are placed on the roads. The VUEs and the BS are all equipped with a single antenna. V2IU and V2VU denote the VUE communicating via V2I and V2V links, respectively. Let us assume that there are M V2IU and K V2VU in the network environment.

In particular, V2IU requires the V2I connection links through Uu interface to facilitate high-capacity communications with the BS, while V2VU pairs (that each has a V2V transmitter and a V2V receiver) demand V2V links for sharing their information, through PC5 interface, for efficient management of traffic safety. It is assumed that every V2I link employs preallocated orthogonal cellular uplink spectrum resources. The total bandwidth is divided into F subbands with a single subband being allocated to each V2IU for uplink transmission. In addition, the number of subbands is assumed to be larger than that of V2IU M. Spectrum utilization efficiency is ensured and improved when considering that the V2V links share the preallocated orthogonal uplink resource spectrum of the V2I links. This is reasonable because the interference at the BS can easily be controlled and the uplink resources are less intensively utilized. For the sake of simplicity, it assumed that not more than one uplink spectrum allocated to V2I links can be multiplexed by each V2V link at a time and that the spectrum of every V2I link is shared simultaneously by the V2V pairs maximum number [23]. Due to the vehicles' high mobility in the system, it is assumed that the large-scale channels are known at the vehicles and the BS. Thus, the channel information from the mth V2IU to the BS, that is between the V2VU pairs k and that from the kth V2VU transmitter to the BS are represented as $h_{m,B}^{\text{V2IU}}$, h_k^{V2VU} and $h_{k,B}^{\text{V2VU}}$, respectively. We also describe the interference channel gains from the kth V2VU transmitter to the BS, from the mth V2IU to kth V2VU receiver and from the kth V2VU transmitter to the jth V2VU receiver as $g_{k,B}, g_{m,k}$, and $g_{k,j}$, respectively.

B. V2IU and V2VU Communication Modes

In this article, the 3GPP standard is considered for the vehicular network, which provides two radio interfaces; the PC5 interface that supports the V2V communication modes and the Uu interface for the V2I communications. Also, the mode 4 of the V2X architecture is considered, in which vehicles have a set of radio resources from which they can select the V2V communication autonomously.

1) V2IUs: For the V2IU, we can only adopt the uplink V2I mode. Hence, the signal interference noise-to-ratio (SINR) of the mth V2IU, γ_m^{V2IU}, is expressed as [22]

$$\gamma_m^{\text{V2IU}} = \frac{P_m^{\text{V2IU}} h_{m,B}^{\text{V2IU}}}{\sum_{k\in K}\sum_{f\in F} \rho_{m,f}\rho_{k,f} P_k^{\text{V2VU}} g_{k,B} + \sigma^2} \tag{1}$$

where P_m^{V2IU} and P_k^{V2VU} represent the transmission power of the mth V2IU and kth V2VU, respectively. $\rho_{m,f} \in \{0,1\}$ is an indicator function for the subband allocation to the V2IU m, we have $m = f$ when $\rho_{m,f} = 1$ and $\rho_{m,f} = 0$, otherwise. $\rho_{k,f}$ denotes the indicator of the spectrum allocation with $\rho_{k,f} = 1$, when the V2VU k reuses the subband f of the mth V2IU or $\rho_{k,f} = 0$, otherwise. σ^2 denotes the noise power. Thus, the achievable capacity of the mth V2IU is formulated as

$$C_m^{\text{V2IU}} = W \log_2\left(1 + \gamma_m^{\text{V2IU}}\right) \tag{2}$$

where W represents the bandwidth. The communication links of V2I are mainly designed to enable smooth mobile broadband access for high data-rate mobile services, thus a suitable design aims at maximizing the sum capacity as $\sum_{m\in M} C_m^{\text{V2IU}}$. Meanwhile, the safety-critical messages are also managed by the V2I links for reliable transmission. These messages are generated periodically for advanced driving services at different frequencies according to the vehicle mobility.

2) V2VUs: It is worth mentioning that for direct communication between V2VU pairs, the V2V mode is selected and for indirect communication through the BS, the V2VU pairs select the V2I communication mode. In the case of the V2V communication mode for the V2VU pairs, the interference occurs from the V2IU and the V2VU sharing the same subbands. Hence, the SINR of the kth V2VU in the V2V mode over the subband f, $\gamma_k^{\text{V2VU-(V)}}$, is formulated as

$$\gamma_{k,f}^{\text{V2VU-(V)}} = \frac{\rho_{k,f} P_k^{\text{V2VU}} h_k^{\text{V2VU}}}{\sum_{m=1}^{M} \rho_{m,f} P_m^{\text{V2IU}} g_{m,k} + \sum_{\substack{j\in K \\ j\neq k}}^{K} \rho_{j,f} P_j^{\text{V2VU}} g_{k,j} + \sigma^2} \tag{3}$$

where P_k^{V2VU} represents the transmission power of the kth V2VU. The first two components on the denominator in (3) capture the interference; the first one is the interference due to V2IU and the second due to V2VU. Thus, the achievable capacity of the kth V2VU in the V2V mode is given as

$$C_k^{\text{V2VU-(V)}} = \sum_{f=1}^{F} W \log_2\left(1 + \gamma_{k,f}^{\text{V2VU-(V)}}\right). \tag{4}$$

In the case of the V2I communication mode, the V2VU pairs firstly upload the safety-critical messages to the BS, then through a downlink, these messages are forwarded to the appropriate V2VU receivers. Note that in the V2I mode, only the unused subbands are assigned to the V2VU pairs and each unused subband can be assigned to a maximum of one V2VU pair. Here, the interference occurs from the V2V communication pairs that share the same subbands, while operating in the V2V communication mode. Thus, the uplink SINR of the kth V2VU pair in the V2I mode over subband f, $\gamma_{k,f}^{\text{V2VU-(I)}}$, is expressed as follows [21]:

$$\gamma_{k,f}^{\text{V2VU-(I)}} = \frac{\rho_{k,f} P_k^{\text{V2VU}} h_{k,B}^{\text{V2VU}}}{\sum_{\substack{j=1 \\ j\neq k}}^{K} \rho_{j,f} P_j^{\text{V2VU}} g_{j,B} + \sigma^2}. \tag{5}$$

According to [27], the achievable capacity of the kth V2VU in the V2I communication mode is expressed as

$$C_k^{\text{V2VU-(I)}} = \frac{1}{2} \sum_{f\in F} W \log_2\left(1 + \gamma_{k_f}^{\text{V2VU-(I)}}\right). \tag{6}$$

Therefore, the SINR γ_k^{V2VU} and the capacity C_k^{V2VU} V2V pair k are, respectively, written as follows:

$$\gamma_k^{\text{V2VU}} = \left(1 - s_k^{\text{V2VU}}\right) \sum_{f=1}^{F} \gamma_{k,f}^{\text{V2VU-(V)}} + s_k^{\text{V2VU}} \sum_{f=1}^{F} \gamma_{k,f}^{\text{V2VU-(I)}} \tag{7}$$

$$C_k^{\text{V2VU}} = \left(1 - s_k^{\text{V2VU}}\right) C_{k,f}^{\text{V2VU-(V)}} + s_k^{\text{V2VU}} C_{k,f}^{\text{V2VU-(I)}} \tag{8}$$

where $s_k^{\text{V2VU}} \in \{0,1\}$ is a mode selection indicator for the V2VU pair with $s_k^{\text{V2VU}} = 1$ when the kth V2VU pair chooses the V2I mode and otherwise, the V2V mode is selected.

C. QoS Requirement Formulation

The QoS requirements need to be taken into account when we want to assure efficient communication in vehicular networks. Multiple types of vehicular system applications exist, with various QoS criteria [1]. Since the V2IUs handle bandwidth-demanding traffic applications, their QoS criteria are presented as the minimum capacity constraints to satisfy a convenient experience. Thus, the V2IUs' capacity constraint is given as

$$C_m^{\text{V2IU}} \geq C_{\text{min}}^{\text{V2IU}} \quad \forall m \in M \tag{9}$$

where $C_{\text{min}}^{\text{V2IU}}$ represents the V2IUs minimum capacity requirement. It is assumed that all the V2IUs have the same capacity for simplicity. The QoS criteria of the V2VU pairs, on the other hand, are the delay requirements such that the V2VU pairs need to send safety-critical messages in real time. With an RA-based decentralized algorithm considered at the VUE side, the delay constraints between the V2VU pairs communication will include the transmission delay with no extra grant-based scheduling delay in the media access control (MAC) layer. The capacity constraint is given as

$$C_k^{\text{V2VU}} \geq \frac{L_k}{T_{\text{max}}}, \quad \forall k \in K \tag{10}$$

where L_k represents the size of the message in bits while T_{max} is the maximum tolerable delay.

Similar to [28], the reliability constraint is converted into an outage probability metric. It is also known as the delivery probability when both delay and reliability constraints are taken into account. However, for simplicity and the sake of convenience, we only consider the reliability constraint as the outage probability metric. Thus, the reliability constraint is formulated as

$$\text{Prb}\left\{\gamma_k^{\text{V2VU}} \leq \gamma_o\right\} \leq p_o \quad \forall k \in K \tag{11}$$

where γ_o and p_o are the SINR threshold for outage and the tolerable outage probability, respectively.

IV. Proposed Multiagent DRL Solutions-Based Decentralized Algorithm

To address the problem of RA in V2V networks, we first model the problem as an MDP and solve it by the proposed decentralized algorithm based on the DRL approach, which is a model-free technique and robust to unpredictable changes in the nonstationary vehicular networks [29]. Then, we present this DRL approach, which assists in finding the mapping between local observations of each vehicle, particularly the interference information and local CSI, as well as the RA scheme. We further improve the scheme by introducing the MA-DDQN-based decentralized scheme to solve the RA problem efficiently. The aim of this RA algorithm, given the resource management of a V2I link, is to guarantee that the reliability and delay constraints on V2V pairs are satisfied while the interference between V2V and V2I links is minimized. The V2VU pairs select the transmission power, selection indicator for the transmission mode and subband according to local observations in the decentralized RA scenario.

A. Markovian Decision Process Formulation

We consider an MDP that consists of sets of state spaces and available joint actions, immediate reward and discounting factor given as $\mathcal{S}, \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_k, r$, and γ, respectively [30]. In the environment considered, multi-V2V agents continuously learn by making decisions after interacting with the network environment in a time slot i. In each discrete time slot i, V2V agents observe the actual state as $s_i \in \mathcal{S}$, then execute an action $a_i \in \mathcal{A}$, by selecting the transmit power, selection indicator mode, and subband according to the π policy. The policy decision, π, is defined by a Q-function, $Q_i(s, a_i; \theta)$, with θ being the Q-function weight obtained by DL. Afterward, the multi-V2V agents receive a reward $r_i = r(s_i, a_i) \in R$ and end up in the following state, $s_{i+1} \in \mathcal{S}$, based on the state transition probability $p(s_{i+1}|s_i, a_i)$.

1) Agents: We consider an MA-DDQN technique with k V2VU pairs being agents in the vehicular communication network.

2) Action Spaces: The proposed model defines the action space $\mathcal{A} = \mathcal{A}_1, \ldots, \mathcal{A}_k$, $k \in K$ as a set of all joint actions that can be taken by the V2V communication links. Every V2V agent chooses an action $a_i \in \mathcal{A}$, during the time slot i, from the action spaces under the currently observed state s. The action includes the subband assignment ρ, the selection mode s^{V2VU}, and the transmission power control p^{V2VU}. It is assumed that the transmission power is discrete [31].

3) State Spaces: Let s_i denote the observed state for describing the network environment by each V2V pair agent, which consists of six parts and can be formulated as $s_i = [I_{i-1}, N_{i-1}, g_i, h_i, T_i, L_i]$, where I_{i-1} represents the received interference power to the V2V link at the precedent time slot i. N_{i-1} is the number of the subchannel of selected neighbors at the precedent time slot i. g_i and h_i are the channel gains of the V2I and V2V communication pairs, respectively. T_i is the remaining time to satisfy the delay requirement, while L_i is the VUE current load for transmission. The quality of every subband channel is demonstrated by the instant channel information and the interference obtained. The distribution of neighbors' selection relates the interference to the other V2IU. The rest of the time and extent of the message to be transmitted requires information to select the appropriate power level.

4) Reward Function: Note that flexibility in designing reward functions makes the DRL suitable for solving problems that are hard to optimize directly. The performance of the vehicular system can easily be improved once the designed reward function and the desired objective both correlate at each step. This reward function is defined by the V2V and V2I capacities, the reliability and delay requirements of the corresponding V2VU pairs. Therefore, the reward function is expressed as

$$r_i = \lambda_1 \sum_{m=1}^{M} C_m^{\text{V2IU}} + \lambda_2 \sum_{m=1}^{M} \left(C_m^{\text{V2IU}} - C_{\min}^{\text{V2IU}}\right) + \lambda_3 \sum_{k=1}^{K} C_k^{\text{V2VU}} + \lambda_4 \sum_{k=1}^{K} \left(\gamma_k^{\text{V2VU}} - \gamma_o\right) + \lambda_5 \sum_{k=1}^{K} \left(C_k^{\text{V2VU}} - \frac{L_k}{T_{k,i}}\right) \tag{12}$$

where $\lambda_2, \lambda_2, \lambda_3, \lambda_4$, and λ_5 are positive weights that balance the V2V and V2I objectives. The multi-V2VU agents aim at finding the optimal strategy π^* for the RA to maximize its received cumulative discounted reward function. The expected cumulative discounted reward is given as $R_i = \sum_{t=0}^{\infty} \gamma^{i-t} r_i$, with $\gamma \in (0, 1]$ being the discounting factor.

It is worth mentioning that interaction between each V2V agent and the unknown environment is mandatory for the agent to gain knowledge and more experience, which are later used in the design of its policy.

In addition, multi-V2V agents jointly investigate the unknown network environment, and improve the power control and spectrum allocation methods according to their own observations of the environmental state.

B. Deep Q-Learning

This article considers the Q-learning method to get the optimal decision policy π^* for the decentralized RA that maximizes the expected cumulative reward function in the vehicular network. In this article, the collaborative multiagents RL is considered with local states because of its distributed characteristic and simplicity. The multiagent refers to the decision making V2V agents that interact in the shared environment. The V2V agents select actions according to the stochastic policy decision π that is mapping the set of state $\mathcal{S}$ to the set of action $\mathcal{A}_i$. This is formulated as $a_i^* = \pi_i^*(s) \in \mathcal{A}_i$. It is important for each V2V agent to get the optimal decision policy to achieve the maximum cumulative discount reward while satisfying all the constraints. Moreover, in the Q-learning process, V2V agents can still take action while collecting rewards. The action selected in the following stage is a function of the current value learned; this is similar to the MDP operation. Thus, the MDP can be formulated as a Q-learning process. Using the Q-learning approach, the optimal Q-value function $Q^*(s, a_i)$ is obtained with Bellman's

equation as follows [32]:

$$Q^*(s,a_i) = r_i(s,a_i) + \gamma \max_{a_i' \in \mathcal{A}_i} Q_i^*(s', a_i'). \tag{13}$$

With the optimal Q-value function $Q_i^*(s,a_i)$ in (13), the optimal policy decision $\pi_i^*(s)$ can be evaluated as $\pi_i^*(s) = \max_{a_i' \in \mathcal{A}_i} Q_i^*(s,a_i')$. The updated Q-value function of the Q-learning for the $Q_i(s,a_i)$ can be obtained through the iterative process as

$$Q_i(s,a_i) = Q_i(s,a_i) + \alpha[r_i(s,a_i) + \gamma \max_{a_i' \in \mathcal{A}_i} Q^i(s', a_i') - Q_i(s,a_i)] \tag{14}$$

with α being the learning rate that determines the effect of new information on the existing Q-value $Q_i(s,a_i)$. The Q-learning can solve the problem of RA using random variables. However, when the high dimensional state-action spaces are addressed in actual complex problems, the efficiency of the Q-learning algorithm can be reduced. As a result, Q-learning might not be sufficient to achieve the optimal strategy within the acceptable time. To solve this problem, we, therefore, propose the multiagent DRL-based approach that uses the DNN to estimate the $Q_i(s,a_i)$ instead of computing a large Q-values table for every (s,a_i) pair in the Q-learning process. The combination of DNN and Q-learning results in the DQN, which is one of the most famous methods of DL. A DNN is defined as a neural network deep graph with many processing layers. In the DQN, DNN approximates the Q-values and the optimal decision policy π^*. Note that a neural network function approximator $Q_i(s,a_i;\theta) \approx Q^*(s,a_i)$ where the parameters θ are used as an on-line network. In this DQN approach, a target network with θ^- is used with the on-line network to stabilize the overall performance of the vehicular network. Thus, the optimal decision policy $\pi^*(s)$ after approximation, is presented as $\pi^*(s) = \arg\max_{a_i' \in \mathcal{A}_i} Q_i^*(s,' a_i'; \theta)$, with $Q_i^*(s,a_i)$ being the optimal Q-value obtained through the DNN. After the DQN has chosen the approximated action, the target Q-value $Q^{\text{DQN}}(s,a_i)$ is expressed as

$$Q_i^{\text{DQN}}(s,a_i) = r_i(s,a_i) + \gamma \max_{a_i' \in \mathcal{A}_i} Q_i(s,' a_i'; \theta^-). \tag{15}$$

The DNN parameter weight θ is continuously updated by minimizing the loss function $L_i(\theta)$ that is described by the difference between the two Q-functions as follows [33]:

$$L_i(\theta) = \mathbb{E}\left[\left(Q_i^{\text{DQN}} - Q_i(s,a_i;\theta) \right)^2 \right]. \tag{16}$$

The action a_i is selected from the on-line network $Q_i(s,a_i;\theta)$ using the greedy policy.

Although the target network Q^{DQN} is a duplication of the on-line network, target network parameters θ^- are set for a certain number of iterations, while the parameters are updated in the on-line network. The gradient descent technique is considered to change the parameter θ and it is computed as follows:

$$\frac{\partial L_i(\theta)}{\partial \theta} = \mathbb{E}\left[Q_i^{\text{DQN}} - \frac{Q_i(s,a_i;\theta) \partial Q_i(s,a_i;\theta)}{\partial \theta} \right]. \tag{17}$$

With the stochastic gradient descent applied, we can update the θ parameter as follows:

$$\theta \longrightarrow \theta + \alpha \left(Q_i^{\text{DQN}} - Q_i(s,a_i;\theta) \right) \nabla_\theta Q_i(s,a_i;\theta). \tag{18}$$

However, the operation "max" in the Q-learning and DQN approaches in (15) uses the same values for the selection and evaluation of actions. This makes overestimated Q-values more likely to be selected, resulting in overoptimistic Q-value function estimates. To prevent this, we consider the DDQN approach in this article [34].

C. Multiagent Double DQN Approach

In this subsection, we describe the proposed MA-DDQN-based decentralized algorithm that uses a joint action learning strategy. It is obvious that when considering a nonstationary environment in vehicular networks, it is hard to collect all the CSI over large networks immediately, resulting in an unstable system during the learning process. Hence, the need to address the problem with a multiagent learning system-based decentralized approach. The proposed framework has two parts, namely, the distributed execution and the centralized training part, both illustrated in Fig. 2.

1) Centralized Training Process: The Q-network is trained through running multiple episodes and all V2V agents explore the state-action spaces with soft policies at every training stage. Particularly during the learning process, the ϵ-greedy strategy is employed to manage the exploration and exploitation of action selection, whereby an action a_i is selected randomly with ϵ probability, otherwise, a greedy action with minimum Q-value is taken. The experience replay allows storage of the experience transition tuple $(s_i^k, a_i^k, r_i^k, s^{k'})$ at time i into a replay buffer B. Then, the DNN parameter weights θ are trained and updated with samples of mini-batch data randomly selected from the replay buffer using the variant of the stochastic gradient descent method referred to in (16)–(18). This allows experience data to be used repeatedly to improve sample effectiveness and then accelerate the convergence of the learning algorithm. Further improvement is achieved when the DDQN is adopted to stabilize the training process and improve the policy decision.

Unlike the DQN, the DDQN uses the double Q-learning process to minimize the overestimation by decomposing the operation "max" in the target network into selection and evaluation of the actions. Specifically, the selection is decoupled from the evaluation. Thus, the target Q-value Q_i^{DQN} is replaced by the target Q_i^{DDQN}, which is expressed as

$$Q_i^{\text{DDQN}}(s,a_i) = r_i(s,a_i) + \gamma Q_i\left(s, \arg\max_{a_i' \in \mathcal{A}_i} Q_i(s,' a_i'; \theta); \theta^-\right). \tag{19}$$

In the DDQN approach, two Q-functions are learned by allocating every experience randomly to update one of the two value functions, knowing that there are two sets of parameters, θ and θ^-. For each update, the on-line network weights θ is employed to determine the greedy strategy, while the target network weights θ^- determine its value. Also, with the experience replay, the DDQN approach is leveraged to train the

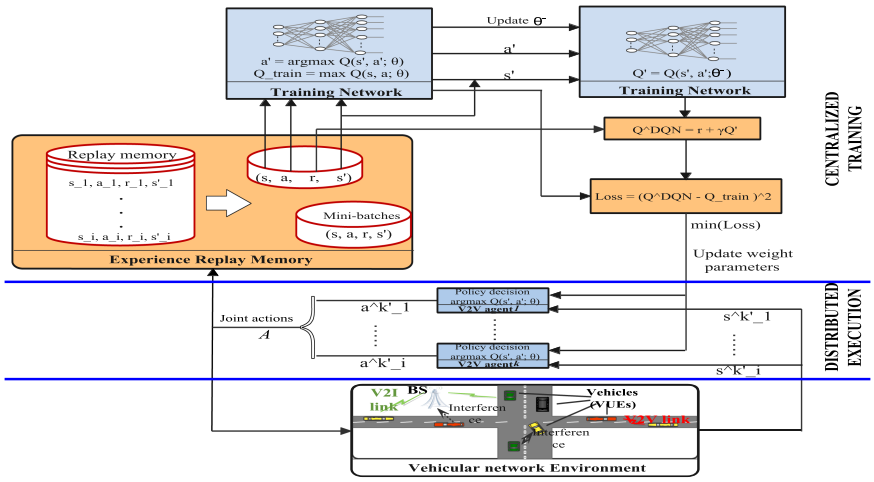

Fig. 2. Illustration of the MA-DDQN scheme with the distributed execution and centralized training processes.

V2V agents and clear out of the divergence owing to the strong correlation between consecutive transitions in the approximation of Q-function for efficient learning of RA.

This stage facilitates the weight-sharing method knowing that the weights are shared with all the V2V agents. This significantly reduces the number of weights to be trained making, the learning process computationally favorable and scalable. Moreover, since all the V2V agents cooperate to optimize the global objectives with the reward designed in (12), it is necessary for each V2V agent to benefit from all the others' experience. However, even when the agents share the same parameter weights during the training process, there is still different behavior among them because the same parameter weight is executed with different local states as input. However, another issue occurs when DDQN is combined with Q-learning, because each V2V agent might face a nonstationary environment, while other neighboring agents also learn to adapt their behavior. This challenge is much more severe when considering the experience replay since it is an important factor in the success of DDQN. This destabilizes the learning process. An efficient way of addressing this issue is that while an agent's action-value function is nonstationary with other V2V agents that change their behavior every time; it can be made stationary based on other agents' policies. So every V2V agent's observation space can be augmented with an estimation of other agents' policies and, therefore, mitigate nonstationarity.

2) Distributed Execution Process: In this part, the V2V agents need to learn the behaviour of the vehicular environment cooperatively in a distributed manner to achieve their objectives. During the time slot i, each V2V agent k selects the action a_i^k with the highest action value based on its trained Q-networks. After that, all the V2V agents begin the transmission with the subband, transmit power level, and selection mode determined by their selected action.

D. Details of the MA-DDQN Scheme

We can see in Algorithm 1, in lines 1–2, that the main structures are defined and initialized, where the limited amount of replay memory is reserved. The on-line Q-function and target Q-function with their respective weights are randomly initialized for the learning process. The learning process starts at the *for loop* between lines 3 and 23 and it adjusts the weights of the on-line Q-function and target Q-function, with the episodes being each iteration. During every training episode, all the V2V agents observe their current states then coordinate their actions based on the ϵ-greedy policy simultaneously. During the learning process, a mini-batch of experience is randomly selected in every time slot from the replay buffer B where the oldest experience is replaced by the latest experience in the queue. The replay buffer B is a first-in–first-out (FIFO) queue with the length being proportional to the number of multi-V2V agents. Once the mini-batch has been selected, the parameters θ are updated to minimize the loss function in (16), with the stochastic gradient descent technique in (17) and (18). As soon as the parameters converge, the training process can end.

V. Binarized Weights and Computational Complexity Analysis

The major concern of deep learning in general and the proposed MA-DDQN scheme, in particular, is computational time complexity. The efficiency and effectiveness of the proposed scheme are presented in this section; we also analyze its complexity and propose binarized weights as a solution to reduce the computational time complexity of our scheme.

A. Efficiency and Effectiveness of the Algorithm

The MA-DDQN scheme faced a speed-related problem due mainly to overdemanding of the hardware (laptop) with some heavy computation. The RA in vehicular networks, particularly using the DRL approach is a complex problem when it comes to computation time. The most intensive task that causes large computational time is the training process of the DNN model. DNN are more difficult to train but yield better performance compared to traditional ML methods requiring more effort on the feature design.

This is the reason we consider a high-performance computer to train and generate the results a little faster. The proposed

Algorithm 1: MA-DDQN-Based Decentralized Algorithm for RA With Joint Actions Learning.

1: **Input**: - Learning rate α, discounting factor γ, replay memory B with capacity C
2: **Initialization:** The Q-value function $Q_i(s, a_i; \theta)$ with random weight θ, target Q-value $Q_i(s,' a_i'; \theta^-)$ with θ^-, where the initial parameters $\theta^- = \theta$
3: **for** *episode* $= 1, \ldots, E$ **do**
4: **for** *agent* $k = 1, \ldots, K$ **do**
5: Initializing the vehicular networks state s
6: **end for**
7: **for**: $i = 1, \ldots, I$ **do**
8: **for** *agent* $k = 1, \ldots, K$ **do**
9: Randomly select an action a_i at the state s using ϵ-greedy probability strategy, otherwise best action as $a_i = \arg\max_{a_i} Q(s, a_i; \theta)$
10: Execute a_i selected
11: Obtain the reward function r_i and next state s'
12: **end for**
13: Store the transition experience (s_i, a_i, r_i, s') in B.
14: Get the random sample mini-batch of transition from B.
15: **if** *episode* ends at step i **then**
16: $Q_i^{\text{DDQN}} = r_i(s, a_i)$,
17: **else**
18: $Q_i^{\text{DDQN}}(s, a_i) = r_i(s, a_i) + \gamma Q_i(s, \arg\max_{a_i' \in \mathcal{A}_i} Q_i(s,' a_i'; \theta); \theta^-)$
19: **end if**
20: Compute the gradient descent step on $L_i(\theta)$ in respect of θ
21: Update the target Q^{DDQN} parameters $\theta^- = \theta$ after every N-steps of iteration
22: **end for**
23: **end for**

scheme yields better results with slightly more computational time when compared to other algorithms, as illustrated in Section VI. Note that to obtain effective results, the proposed model needs to be trained for a long time. One can also consider using GPU in the laptop instead of using a server to run this algorithm. However, further work is being done to improve the computational efficiency of the proposed algorithm to obtain slightly better results in less computational time. Therefore, we consider the binarized weights method to solve this issue [11].

B. Training DNN With Binarized Weights

Since the main reason for the high complexity of this method resides in the training process, solutions to reduce the trained data through advanced training techniques may also reduce the complexity of this algorithm. The reduction of the DNN computational complexity has previously been studied through several schemes. In this article, we propose to binarize the weights during the DNN training process. This method is very efficient in terms of memory and computation time [10]. When considering Algorithm 1, this process fits between line 15 and line 21.

Note that each iteration of training a neural network involves three steps, namely, forward pass, backward pass, and the parameter updates in the back-propagation algorithm. Applying DNN involves convolutions and matrix multiplications. Hence, the computation performed during the training process of the proposed MA-DDQN scheme relates to the multiplication of a real-valued weight θ^R and a real-valued activation η in the forward pass or gradient in the backward pass of the back-propagation algorithm. Eliminating the need for these multiplication operations reduces the training time. Thus, we constrain the algorithm to use the binary weights $\theta_l^B \in \{-1, 1\}, l = 1, \ldots, N$ during the forward and backward pass to eliminate these multiplications. N represents the number of DNN layers. As a result, many multiplication operations are replaced with simple additions and subtractions in this process. The binary values are used during the training process instead of using these at the end of the training. This provides a complete representation loss function to train against. In fact, it is not a problem to compute the gradient of the loss function L with the binary weights through back propagation. However, the parameter updates using the gradients descent method are very hard with the binary weights. To solve this problem, a set of real-valued weight θ^R, which is generated in the continuous interval of $[-1, 1]$ and binarized later to obtain binary weights θ^B, is kept. Later, the θ^R are then updated through back propagation and the incremental updates gradient descent. If a weight update brings the real-valued weight θ^R outside the interval $[-1, 1]$, it is clipped with the $clip()$ function. This is done because otherwise; the real-valued weight θ^R will grow large without having any impact on the binary weight θ^B. We consider the sign function sign() method to binarize the real-valued weights as

$$\theta^B(x) = \begin{cases} -1, & x < 0 \\ +1, & \text{otherwise.} \end{cases} \tag{20}$$

The gradient of this function, however, is not continuous, presenting a challenge to the DNN training's back propagation. Hence, the straight-through estimator (STE) is adopted in the backward pass. Sometimes, the binarized weight method for DNN training can take more training time than the traditional DNN because of the STE heuristic required to approximate the gradient of the real-valued weight θ^R [10]. Thus, the batch normalization layer and ADAM optimizer are used to speed up the training process by internal covariate shifting all the bits, updating the real-valued weights and reducing the overall effect of the weights scale. The DNN learning process of the proposed MA-DDQN with the binarized weights (MA-DDQN-BW) is illustrated in Algorithm 2.

1) In the *forward pass*, in the training stage, with the given DNN inputs (s_i, a_i), the unit activations are computed layer by layer from $l = 1, \ldots, N$, which leads to the last layer, the output layer. This is where the real-valued weights θ_l^R are binarized with the sign() function, then

Algorithm 2: Training Process of DNN with Binarized Weights.

Input: - A minibacth of (inputs, targets), loss function L, previous weight θ_{l-1}^R, learning rate α.
Forward pass:
for $l = 1, \ldots, N$ **do**
  if in the training stage **then**
    Use the sign*()* to obtain $\theta_l^B = \text{sign}(\theta_l^R)$
    Calculate the activation η_l knowing η_{l-1}, θ_l^B
  end if
end for
Backward pass:
Initialize the output layer's activation gradient $\frac{\partial L}{\partial \eta_N}$
for $l = N$ to 2 **do**
  Compute $\frac{\partial L}{\partial \eta_{l-1}}$ knowing $\frac{\partial L}{\partial \eta_l}$
end for
Parameter update:
for $l = 1, \ldots, N$ **do**
  Compute $\frac{\partial L}{\partial \theta_l^B}$ knowing $\frac{\partial L}{\partial \eta_l}$ and a_{l-1}
  Update the weight $\theta_l^R \leftarrow clip(\theta_{l-1}^R - \alpha \frac{\partial L}{\partial \theta_l^B})$
end for
Output: Updated weight parameters θ_l^R

the activations are computed knowing the binary values θ_l^B and the previous activation η_{l-1}.

2) In the *backward pass*, we consider the DNN targets that are the expected return r_i, the training of the objective gradient for every layer's activation, starting from the output layer and going down layer by layer to the first hidden layer. We compute $\frac{\partial L}{\partial \eta_{l-1}}$ knowing $\frac{\partial L}{\partial \eta_l}$.
3) In the *parameter updates*, the gradient descent is computed for each layer's parameter weight. Then, the parameter weight is updated using their computed gradients and their previous values.

C. Complexity Analysis of the Algorithms

For the simulation, we use a server to run the MA-DDQN scheme. The computational complexity at each V2V agent is dominated by the training of the DNN with and without the binarized weights method. The complexity analysis of all the algorithms is as follows:

1) With the MA-DDQN, we need to consider the matrix multiplications. Assuming that the number of neurons in the l^{th} layer is n, note that the number of multiplications required to compute the activation of all neurons in the l^{th} layer, such that we have five layers (including input and output layers), is $n(l) * n(l-1)$. The output neurons are simple to compute; thus, for the vehicular network with n neurons, the output step is in $\mathcal{O}(n)$. The computational complexity of training the DNN of the proposed scheme is $\mathcal{O}(n^2)$.
2) When calculating the time complexity for training the DNN back propagation for the MA-DDQN-BW approach, we need to consider the different factors contained in the DNN training, such as the number of layers, iterations, training episodes, number of neurons in each layer, etc. Note that when using the binarized weight method, the matrix multiplication operations have been reduced. Therefore, after calculations, the complexity of this method is $\mathcal{O}(nt * (N_1 + N_2 + N_3 + N_4 + N_5))$, with t being the training episodes.
3) The complexity of the random algorithm for RA usually increases linearly with the increase in the number of V2V agents, which is $\mathcal{O}(K)$, with K being the number of the V2V agent [35].
4) The complexity of the DRL algorithm with mode selection (DRL-MS) proposed in [21] is similar to the MA-DDQN without binarized weight approach. It is $\mathcal{O}(n^2)$.
5) The DRL algorithm needs to satisfy the constraint while training the DNN, which makes the algorithm have a high complexity of $\mathcal{O}(n^2)$ [17].

We notice that the complexity of both the MA-DDQN and the DRL algorithms, has at least the same high computational complexity. However the proposed MA-DDQN-BW scheme has lower complexity when compared to the precedents. The random algorithm, on the other hand, has the lowest complexity but it is an unstable algorithm for the RA in the vehicular networks.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Number of V2V pairs	[20, 40, 60, 80, 100]
Number of V2I and Vehicle speed	5 and 36 Km/h
Carrier frequency and Bandwidth	2 GHz and 10 MHz
Number of sub-bands	10
Delay constraints for V2V links	100 ms
Vehicle and BS receiver noises	9 dB and 5dB
Vehicle antenna gain and height	3dBi and 1.5 m
BS antenna gain and height	8 dBi and 25 m
V2V transmit power level and Noise power σ^2	23dBm and -114dBm

VI. SIMULATION RESULTS

The simulations that were conducted are presented to evaluate the performance of the proposed schemes-based decentralized algorithm to allocate resources effectively in the vehicular communication network. The performance metrics are evaluated in terms of the sum-rate and delivery probabilities of the V2I and V2V communication links, respectively. We also evaluate the accuracy and error rate when training the MA-DDQN, MA-DDQN-BW, DRL-MS, and DRL algorithms.

A. Simulation Setting

A scenario considering a cellular communication network with a BS placed at the center of the network configuration is presented in the simulation. The following setup is similar to the Manhattan case detailed in 3GPP TR 36.885 [36], where vehicles are randomly placed according to the spatial Poisson process with a mobility speed of 36 Km/h. This ensures that all subbands are entirely reused by the V2V communication link and also verifies the algorithm's robustness. The simulation parameters are summarized in Table II. The DQN contains five

TABLE III
DDQN PARAMETERS

Parameter	Value
Learning rate α and discounting factor γ	0.01 and 0.5
Size of experience replay memory B	1000000
Size of mini-batch and Number of episode	2000 and 2500
Number of steps in each episode	100
Weight in the reward	[0.1, 0.9, 1, 1, 1]
Activation function	RELU, tanh
Neurons for hidden layers	500, 250, 120
Optimizer	Adam

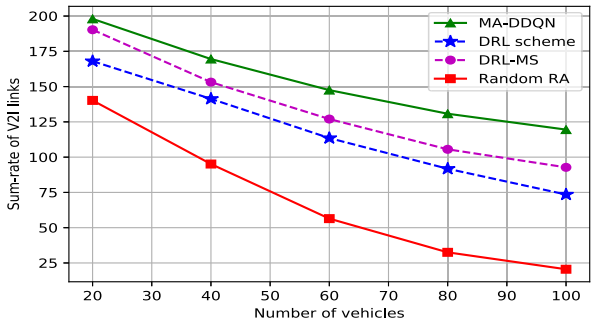

Fig. 3. Sum-Rate of the V2I communication links versus the number of vehicles.

NN layers that are fully connected with three hidden layers. The ReLU and tanh activation functions are adopted and expressed as f(b) = max(0, b) and tanh = $\frac{2}{1+\exp(-2b)} - 1$, respectively. The learning rate has an initial value that decreases exponentially. We use the ϵ-greedy strategy to manage the exploration and exploitation with the ϵ value. Table III shows the details of all the DDQN parameters.

B. Evaluation Results and Analysis

Three other algorithms are selected and simulated to demonstrate the effectiveness of the MA-DDQN algorithm.

1) The DRL algorithm [17] considered the latency and reliability requirements. However, the V2V pairs adopted only the V2V communication (selection) mode and every V2V pair independently selects its subband and its transmission power according to the local DRL framework.
2) The DRL-MS algorithm in [21] also used the mode selection approach for the V2I and V2V communications as considered in the proposed scheme. However, the DRL framework and DNN training process of the DRL-MS algorithm is similar to [17].
3) The random RA scheme, a subband for transmission with the lower interference, is randomly selected by the V2V agent from a pool of candidate subbands consisting of 5 RB. The transmission power is fixed to a maximum transmission power. Note that the V2V mode is the only communication mode adopted.

Fig. 3 illustrates the sum-rate performance of the V2I links versus the number of vehicles. It is observed that as the number of vehicles increases, the sum-rate performance of the V2I links decrease for all the algorithms. This is because when increasing the number of vehicles, the use of V2V communication links

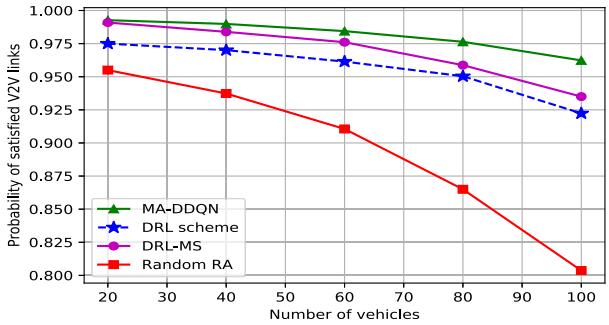

Fig. 4. Probability of satisfied V2V communication links versus the number of vehicles.

increases as well, which results in the growth of interference coming from the V2V to V2I links and, therefore, causes a drop of summation-rate of the V2I links. Nevertheless, the MA-DDQN scheme performs better with the highest sum-rate of V2I links to mitigate the interference. This is because when the number of vehicles increases and causes more interference to nearby vehicles, the MA-DDQN can select the best transmission mode at the time, according to the local observations, while others fail to do so. This is the purpose of the mode selection technique considered in our scheme. However, the DRL-MS scheme also uses the mode selection approach, but, even while using this method, the proposed scheme performs better. This is due to the DDQN framework considered in our scheme. The performance of the DRL-MS algorithm seems to decrease more when a larger number of vehicles is considered compared to the proposed scheme. The random algorithm, contrarily, has the worst sum-rate performance due to the random allocation of subbands and transmit power, resulting in catastrophic interference between the V2I and V2V links. As for the DRL algorithm, the performance is somewhat better than the random scheme but not as good as the MA-DDQN scheme so no selection mode is considered.

Fig. 4 presents the performance of probability of satisfied V2V links in terms of delay and reliability requirements versus the number of vehicles. Similar to Fig. 3, it is observed that when the number of vehicles increases, the average probability of satisfied V2V links decreases for all the algorithms. However, the MA-DDQN scheme still gives better performance to satisfy the delay and reliability constraints compared to the other algorithms. It guarantees the probability of satisfied V2V links at between 0.99% and 0.95% even in the worst case of 100 vehicles being deployed. Furthermore, in the proposed scheme, when the V2V links in the crossroads choose the V2I communication mode, lower transmission power is required to satisfy the reliability requirement when the number of V2V links is very large, while other algorithms fail.

Fig. 5 presents the MA-DDQN effectiveness by illustrating the learning process of the cumulative reward with the number of vehicles set to 20. We can observe that at the start of the training process, the average cumulative reward is lower and increases as the episodes increase, demonstrating the effectiveness of the MA-DDQN learning scheme. The average cumulative reward, however, becomes stable when it reaches stable values. In our

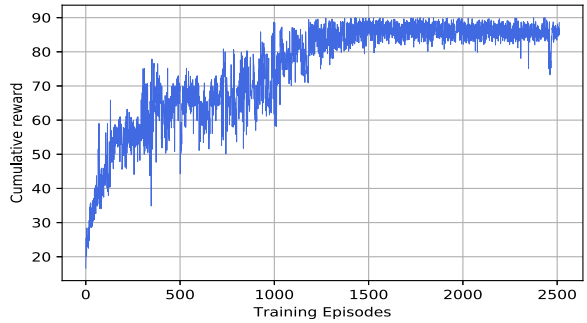

Fig. 5. Cumulative reward versus the training episodes.

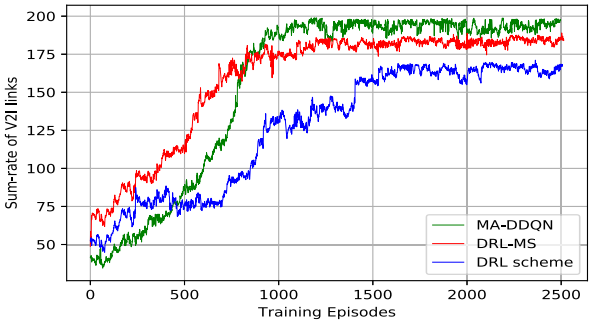

Fig. 6. Training process of the sum-rate of the V2I communication links with the number of vehicles being 20.

case, despite some variations due to mobility-induced channel fading in the environment, it is when the training episodes approximate 1200 episodes that the performance of the average reward progressively converges. According to this observation, we, therefore, trained V2V agents for 2500 episodes when evaluating the V2V and V2I links performances in Figs. 3 and 4 to guarantee safe convergences.

Fig. 6 illustrates the optimization comparison of the sum-rate training process of the V2V communication links between the MA-DDQN, DRL-MS, and the DRL schemes, respectively. For this simulation, the number of vehicles is set to 20. Although the MA-DDQN is initially unstable when compared to the DRL scheme, it is obvious that the MA-DDQN generally outperforms the DRL-MS and DRL schemes. However, before 500 episodes the DRL algorithm performs a little better than our proposed scheme. However, after 500 episodes, the MA-DDQN scheme begins to outperform the DRL algorithm owing to the implementation of the distributed execution and centralized training methods proposed in our scheme. The sum-rate learning process of the DRL-MS scheme also starts low and increases as the training episodes increase. It becomes stable around 1000 episodes but still the proposed scheme outperforms the convergence of the DRL-MS. The higher the increase in the training episodes, the stable the performance of the MA-DDQN scheme becomes. It can be observed that after approximately 1100 episodes, the performance of the proposed scheme becomes more stable, while the DRL scheme stabilizes after approximately 1500 episodes. This shows that the convergence of the MA-DDQN scheme is achieved faster compared to the DRL scheme.

TABLE IV
ACCURACY AND EXECUTION TIME

	MA-DDQN	MA-DDQN-BW	DRL	DRL-MS
-Time	3.72×10^{-4}	5.21×10^{-5}	2.4×10^{-4}	2.9×10^{-4}
-Accuracy	94.3%	**82.1**%	92.6%	93.1%

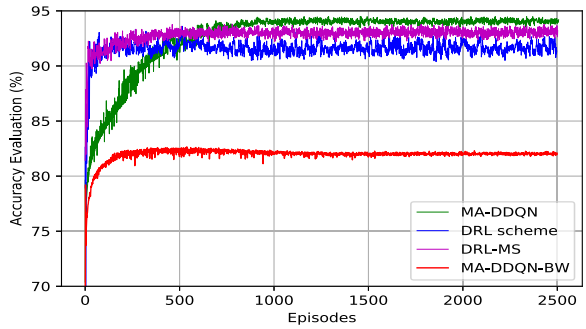

Fig. 7. Accuracy evaluation versus the training episode.

C. Performance Evaluation for Training With Binarized Weights

In this subsection, we present the performance evaluation for the DNN training when considering the binarized weights. This method reduces the computational time cost of the MA-DDQN by binarizing the weights during training, allowing faster computation results. We can see from Table IV, that the training time of the proposed MA-DDQN scheme is $\approx 6\times$ improved when training the DNN using the binarized weight compared to the other schemes. However, this affects the accuracy of the proposed scheme, which is 94.3%. We can notice it in form of $\approx 12\%$ accuracy degradation, which becomes 82.1%. Even if; the accuracy of the MA-DDQN scheme is still acceptable, we observe that the DRL-MS and DRL algorithms have better accuracy than the MA-DDQN-BW scheme.

We also notice that the speed training of the DRL and DRL-MS algorithms may be acceptable (2.4×10^{-4} and 2.9×10^{-4} s, respectively), because the constraints considered in the network are not very stringent. The training time of the MA-DDQN is, however, the highest among the other algorithms at 3.72×10^{-4} s owing to all the operations required when training the DNN. Interestingly, the MA-DDQN-BW scheme used to train the DNN; gives a much better training time (5.21×10^{-5} s); it removes almost $2/3$ of multiplication operations in the network. It is also worth noting that the agent's computational capacity in a practical network is stronger than that of a computer used in a simulation. As a result, in a practical network, the average time required to train a DNN to achieve efficient RA in V2V networks can be reduced even further.

Fig. 7 illustrates the accuracy performance with a mini-batch size of 2000. It can be seen that the accuracies of all the schemes increase as the training episodes increase. For the MA-DDQN-BW and DRL, we observe that the accuracy increase more quickly with the training of episodes when compared to the MA-DDQN. However, the accuracy performance of the MA-DDQN-BW is lower than that of the other schemes. This is

due to the binarization of the weights, which lowers the accuracy but achieves a stable accuracy value faster than the proposed scheme (MA-DDQN).

VII. Conclusion

In this article, we proposed a decentralized MA-DDQN scheme consisting of centralized learning and distributed implementation processes to allocate resources for the V2V communication networks efficiently. A selection transmission mode for the V2V has been considered to avoid interference caused to nearby vehicles. The simulation results demonstrated that the MA-DDQN algorithm maximized the sum-rate of V2I communications, while guaranteeing the delay and reliability for the V2V communications. To improve the complexity of the proposed scheme further, we proposed to binarize the weights during the DNN training process, where weights became binary values. The architecture of this type of vehicular network makes use of memory computational time more effectively with an execution time of 5.21×10^{-5} s. However, reducing the computational time of our DNN training process affected the accuracy of the proposed scheme, yielding $\approx 12\%$ less accurate performance. Future works will consist of developing other techniques to maintain accuracy, while reducing computational time. In addition, we will investigate both discrete and continuous actions spaces and consider an extension to the DDPG algorithm. This will include the in-depth analysis and comparison of the robustness of this proposed scheme and the extended DDPG scheme.

References

[1] M. Noor-A-Rahim, Z. Liu, H. Lee, G. M. N. Ali, D. Pesch, and P. Xiao, "A survey on resource allocation in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 701–721, Feb. 2022.

[2] E. E. TR, "101 607: Intelligent transport systems (ITS); cooperative ITS (C-ITS)," ESTI, Sophia Antipolis Cedex, FRANCE, Release 1. Technical Report Ver 1.1. 1, Eur. Telecommun, 2013.

[3] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[4] Y. Yang, D. Fei, and S. Dang, "Inter-vehicle cooperation channel estimation for IEEE 802.11 P V2I communications," *J. Commun. Netw.*, vol. 19, no. 3, pp. 227–238, 2017.

[5] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: A survey," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, May 2013.

[6] L. Liang, H. Peng, G. Y. Li, and X. Shen, "Vehicular communications: A physical layer perspective," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10647–10659, Dec. 2017.

[7] C. Guo, L. Liang, and G. Y. Li, "Resource allocation for low-latency vehicular communications: An effective capacity perspective," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 905–917, Apr. 2019.

[8] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

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

[10] T. Simons and D.-J Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, 2019, Art. no. 661.

[11] M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Proc. Adv. Neural Inform. Process. Syst.*, 2015, pp. 3123–3131.

[12] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to 1 or-1," 2016, *arXiv:1602.02830*.

[13] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: Imagenet classification using binary convolutional neural networks," in *Proc. Euro. Conf. Comput. Vision*, 2016, pp. 525–542.

[14] M. Zhang, S. Wang, and Q. Gao, "A joint optimization scheme of content caching and resource allocation for internet of vehicles in mobile edge computing," *J. Cloud Comput.*, vol. 9, 2020, Art. no. 33.

[15] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for D2D-enabled vehicular communications," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 3186–3197, Jul. 2017.

[16] M. Chen, J. Chen, X. Chen, S. Zhang, and S. Xu, "A deep learning based resource allocation scheme in vehicular communication systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2019, pp. 1–6.

[17] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[18] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Reinforcement learning for resource provisioning in the vehicular cloud," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 128–135, Aug. 2016.

[19] F. Wilhelmi, B. Bellalta, C. Cano, and A. Jonsson, "Implications of decentralized Q-learning resource allocation in wireless networks," in *Proc. IEEE 28th Annu. Int. Symp. Personal, Indoor, Mobile Radio Commun.*, 2017, pp. 1–5.

[20] M. Yassin, S. Lahoud, K. Khawam, M. Ibrahim, D. Mezher, and B. Cousin, "Centralized versus decentralized multi-cell resource and power allocation for multiuser OFDMA networks," *Comput. Commun.*, vol. 107, pp. 112–124, 2017.

[21] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6380–6391, Jul. 2020.

[22] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.

[23] Y.-H. Xu, C.-C. Yang, M. Hua, and W. Zhou, "Deep deterministic policy gradient (DDPG)-based resource allocation scheme for NOMA vehicular communications," *IEEE Access*, vol. 8, pp. 18797–18807, 2020.

[24] F. Rezazadeh, H. Chergui, and C. Verikoukis, "Zero-touch continuous network slicing control via scalable actor-critic learning," 2021, *arXiv:2101.06654*.

[25] S. Chen, J. Hu, Y. Shi, and L. Zhao, "LTE-V: A TD-LTE-based V2X solution for future vehicular network," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 997–1005, Dec. 2016.

[26] G. Cerutti, R. Andri, L. Cavigelli, E. Farella, M. Magno, and L. Benini, "Sound event detection with binary neural networks on tightly power-constrained IoT devices," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Des.*, 2020, pp. 19–24.

[27] N. Sawyer and D. B. Smith, "A Nash stable cross-layer coalitional game for resource utilization in device-to-device communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8608–8622, Sep. 2018.

[28] L. Liang, S. Xie, G. Y. Li, Z. Ding, and X. Yu, "Graph-based resource sharing in vehicular communication," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4579–4592, Jul. 2018.

[29] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *ACM Comput. Surveys (CSUR)*, New York, NY, USA: ACM, vol. 54, no. 6, pp. 1–25, 2021.

[30] O. Sigaud and O. Buffet, *Markov Decision Processes in Artificial Intelligence*. Hoboken, NJ, USA: Wiley, 2013.

[31] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5 G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications," *IEEE Veh. Technol. Mag.*, vol. 12, no. 4, pp. 30–39, Dec. 2017.

[32] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for interference control in OFDMA-based femtocell networks," in *Proc. IEEE 71st Veh. Techn. Conf.*, 2010, pp. 1–5.

[33] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[34] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.

[35] O. A. Saraereh, S. L. Mohammed, I. Khan, K. Rabie, and S. Affess, "An efficient resource allocation algorithm for device-to-device communications," *Appl. Sci.*, vol. 9, no. 18, 2019, Art. no. 3816.

[36] "Study on evaluation methodology of new vehicle-to-everything (V2X) use cases for LTE and NR," 3GPP Sophia Antipolis, France, Tech. Rep. TR 37.885, release 16, 2018.