Received October 7, 2019, accepted October 29, 2019, date of publication November 8, 2019, date of current version November 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952411

Distributed Deep Deterministic Policy Gradient for Power Allocation Control in D2D-Based V2V Communications

KHOI KHAC NGUYEN[1], **TRUNG Q. DUONG**[1], **(Senior Member, IEEE), NGO ANH VIEN**[1], **NHIEN-AN LE-KHAC**[2], **AND LONG D. NGUYEN**[3]

[1]School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, U.K.
[2]School of Computer Science, University College Dublin, Dublin, Ireland
[3]Duy Tan University, Da Nang 550000, Vietnam

Corresponding author: Trung Q. Duong (trung.q.duong@qub.ac.uk)

This work was supported in part by the U.K. Royal Academy of Engineering Research Fellowship under Grant RF1415\14\22, and in part by the Newton Fund Institutional Link through the Fly-by Flood Monitoring Project under Grant 428328486, which is delivered by the British Council.

ABSTRACT Device-to-device (D2D) communication is an emerging technology in the evolution of the 5G network enabled vehicle-to-vehicle (V2V) communications. It is a core technique for the next generation of many platforms and applications, e.g. real-time high-quality video streaming, virtual reality game, and smart city operation. However, the rapid proliferation of user devices and sensors leads to the need for more efficient resource allocation algorithms to enhance network performance while still capable of guaranteeing the quality-of-service. Currently, deep reinforcement learning is rising as a powerful tool to enable each node in the network to have a real-time self-organising ability. In this paper, we present two novel approaches based on deep deterministic policy gradient algorithm, namely "distributed deep deterministic policy gradient" and "sharing deep deterministic policy gradient", for the multi-agent power allocation problem in D2D-based V2V communications. Numerical results show that our proposed models outperform other deep reinforcement learning approaches in terms of the network's energy efficiency and flexibility.

INDEX TERMS Non-cooperative D2D communication, D2D-based V2V communications, power allocation, multi-agent deep reinforcement learning, and deep deterministic policy gradient (DDPG).

I. INTRODUCTION

Vehicle-to-vehicle (V2V) communication, which utilises intelligent vehicles in order to improve traffic safety and reduce energy consumption, has recently emerged as a promising technology. There have been researches on V2V communications that aim to make each vehicle more intelligent while ensuring safety [1], [2]. The V2V technology facilitates efficient supervision of possible pitfalls in the roadways by allowing vehicles to cooperate with the already existing transport management systems. Moreover, intelligent transport systems can exploit data from V2V communications to enhance traffic management and enable vehicles to communicate with road infrastructures in order to build more reliable self-driving cars.

In device-to-device (D2D) communications, end-users can interact with each other without having to connect directly to base stations (BS) or core networks. It enables the development of various platforms and applications. For example, D2D communication is a core technique in smart cities [3], high-quality video streaming [4], and disaster relief networks [5]. D2D communication can also support V2V communications as it has tremendous advantages such as spectral efficiency, energy efficiency, and fairness [6]–[9]. Firstly, the V2V communications under the D2D-enabled architecture are supported through localized D2D communication to inherit the benefits of D2D-based networks. Techniques that are used in D2D communication substantially reduce latency and power consumption; hence, they are suitable for tight delay V2V communications. Secondly, the requirement of time constraint in V2V links is strict as in D2D pairs due to the low latency is essential for critical safety services. In addition, the demand for high reliability in V2V communication is approximately similar in D2D communication. The V2V link reliability is guaranteed by ensuring the SINR is not lower than a small threshold. We identify and incorporate the

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

reliability QoS requirements for V2V links into the objective formulation. Therefore, the D2D communication represents an emerging solution to enable safe, efficient, and reliable V2V communications. However, the resource allocation problem is one of the challenges to enable D2D-based V2V communications due to rapid channel variations caused by V2V user mobility.

Resource allocation problems in D2D communication have received enormous attention from the research community [10]–[15]. In [10], the authors considered three scenarios, namely the perfect channel state information, partial channel state information, and imperfect channel between the users and the transmitters, to present a resource allocation algorithm to achieve optimal performance in terms of secrecy throughput and energy efficiency. In [11], the authors introduced an optimisation scheme based on the combination of coral reefs optimisation and quantum evolution to gain the optimal results for joint resource management and power allocation problem in cooperative D2D heterogeneous networks. The authors in [12] proposed an optimisation algorithm based on logarithm inequality to solve the joint energy-harvesting time and power allocation in D2D communications assisted by unmanned aerial vehicles. Meanwhile, in [13], in order to maximise the total average achievable rate from D2D transmitters to D2D receivers, the authors proposed an optimal solution to allocate the spectrum and power in cooperative D2D communications with multiple D2D pairs. In [14], a resource allocation approach was presented to improve energy-efficient D2D communication. In particular, the power allocation problem was solved by using the Lambert W function, and channel allocation was solved appropriately by Gale-Shapley matching algorithm. However, all the above approaches have a common drawback that requires the data of all D2D pairs to be collected and processed in a centralised manner at the BS. It causes delays in real-time scenarios. Furthermore, many previous algorithms typically only work on a small, static environment and all the data was analysed at one point. It is not realistic because environments are dynamic and centralised processing will inflict a bottleneck, congestion, and blockage at the BS or central processing unit.

Some recent works have studied to apply techniques in D2D communication to support V2V communications [6]–[9]. In [7], a cluster-based resource block sharing algorithm and in [9] a separate resource block algorithm were proposed to deal with the radio resource allocation problem in D2D-based vehicle-to-everything communications. Meanwhile, the authors in [8] proposed a grouping algorithm, channel selection, and power control strategies to maximise the performance of a network consisting of multiple D2D-based V2V links sharing the same channel. However, the major issue of D2D communication is that each D2D pair in the network typically has limited resources and power for transmitting information whilst the demand for efficient resource allocation such as spectrum and power allocation is rising rapidly. Furthermore, each pair in D2D networks cannot frequently transfer or store in their memory the information of its resource allocation scheme due to limitations in transmission power and memory storage. Besides, if we use BS as a central processing unit to find a resource allocation scheme for each pair, the delay incurred will make the system model unsuitable for real-time applications. Recently, efficient optimisation algorithms have been deployed to enhance both energy efficiency and processing time [12], [16], [17].

In [18], reinforcement learning algorithm (RL) was used to obtain the optimal policy for the power control problem in energy harvesting two-hop communication. The authors considered that each energy harvesting node only knows the harvested energy and channel coefficients. Thus, the problem can be transferred to two point-to-point problems, and to maximise the amount of data at the receiver, RL algorithm called SARSA is employed at each energy harvesting node to reach the optimal policy at a transmitter. Nevertheless, the RL based algorithm has some disadvantages such as instability and inefficiency when the number of nodes in the network is sufficiently large.

Recently, deep learning (DL), a subfield of machine learning, is a powerful optimisation tool to solve the resource management problems in modern wireless networks [19], [20]. An approach based on deep recurrent neural networks was presented in [19] to obtain the optimal policy for resource allocation in a non-orthogonal multiple access-based heterogeneous internet-of-things network. In [20], the authors proposed a deep learning-based resource management scheme to balance the energy and spectrum efficiency in cognitive radio networks. By utilising the neural networks, the convergence speed was significantly improved in terms of the lower computational complexity and learning cost while satisfying the network performance. DL has also been applied to solve the physical layer issues in wireless networks [21]–[25]. The authors in [21] proposed a convolutional neural network-based method to automatically recognise eight popular modulation models, which are used in advanced cognitive radio networks. The proposed network was trained by using the two datasets of in-phase and quadrature to extract features and efficiently classify modulated signals. Meanwhile, the authors in [22] introduced a fully-connected neural network-based framework for maximising the network throughput under the limited constraint of total transmit power. The data was generated without labels and put into the neural network for offline unsupervised training. The DL-based algorithms were also proposed to enable mmWave massive multiple-input multiple-output framework for hybrid precoding schemes [23] and to detect the channel characteristics automatically [24].

Deep reinforcement learning, a combination of RL and deep neural network, has been used widely in wireless communication thanks to its powerful features, impressive performance, and adequate processing time. The authors in [26] formulated a non-cooperative power allocation game in D2D communications and proposed three approaches based on deep Q-learning, double deep Q-learning, and dueling deep

Q-learning algorithm for multi-agent learning to find the optimal power level for each D2D pair in order to maximise the network performance. The authors in [27] used deep Q-learning algorithm to look for the optimal sub-band and transmission power level for each V2V user in V2V communications while satisfying the requirement of low latency. However, these algorithms can only work on the discrete action space; hence, human intervention is required to design the power level of each pair. With the finite set of action space, the performance of these algorithms cannot reach the optimal result, and the reward can become worse if we cannot divide the power level accurately.

Against this background, in this paper, we propose two novel models termed as distributed deep deterministic policy gradient (DDDPG) and sharing deep deterministic policy gradient (SDDPG) based on deep deterministic policy gradient (DDPG) algorithm [28]. Our proposed approaches can work on a continuous action space for the multi-agent power allocation problem in D2D-based V2V communications. Therefore, we can improve the algorithm convergence quality and sample efficiency significantly, especially when the number of V2V pairs in the network increases. From the numerical results, we show that our model outperform the approach based on the original DDPG algorithm in terms of energy efficiency (EE) performance, computational complexity, and network flexibility. Our main contributions are as follows:

- We provide two novel approaches based on DDPG algorithm to solve the multi-agent learning and non-cooperative power allocation problem in D2D-based V2V communications. Experiment results show promising results over other existing deep reinforcement learning approaches.
- By modifying the input of the neural network, all the agents in the multi-agent deep reinforcement learning algorithm can share one actor network and one critic network to reach higher performance and faster convergence while reducing the computational complexity and memory storage significantly.
- Finally, after training the policy neural network, the non-cooperative power allocation problem in D2D-based V2V communications can be solved in milliseconds. It becomes a promising technique for real-time scenarios.

The remainder of the paper is organised as follows. In Section II, we describe the system model and formulation of the multi-agent power allocation problem in D2D-based V2V communications. Section III describes the value functions, policy gradient concepts, and proposes distributed deep deterministic policy gradient algorithm-based method. In Section IV, we improve the model by using the embedding layer to solve the non-cooperative resource allocation problem in D2D-based V2V communications efficiently. In Section V, the simulation results are presented to demonstrate the efficiency of our proposed schemes. Finally, we conclude this paper and propose some future works in Section VI.

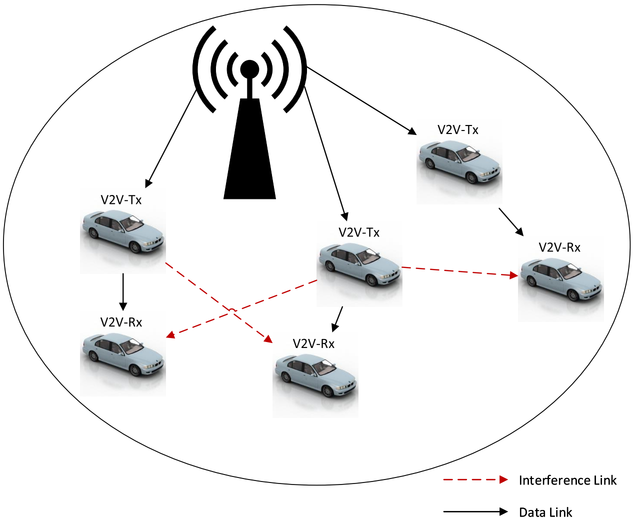


FIGURE 1. System model of D2D-based V2V communications.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we define the system model and formulation of the power allocation problem in D2D-based V2V communications. As depicted in Fig. 1, there are N V2V pairs are distributed randomly within the coverage of one BS. Each V2V pair consists of a single antenna V2V transmitter (V2V-Tx) and a single antenna V2V receiver (V2V-Rx). We define that β_0, f_i and α_h are the channel power gain at the reference distance, an exponentially distributed random variable with unit mean, and the path loss exponent for V2V links, respectively. The location of the ith V2V-Tx and jth V2V-Rx with $i, j \in \{1, \ldots, N\}$ are $(x^i_{\text{Tx}}, y^i_{\text{Tx}})$ and $(x^j_{\text{Rx}}, y^j_{\text{Rx}})$. Hence, the channel power gain h_{ij} between the ith V2V-Tx and jth V2V-Rx is written as

$$h_{ij} = \beta_0 f_i^2 R_{ij}^{-\alpha_h}, \tag{1}$$

where $R_{ij} = \sqrt{(x^i_{\text{Tx}} - x^j_{\text{Rx}})^2 + (y^i_{\text{Tx}} - y^j_{\text{Rx}})^2}$ is the Euclidean distance between the ith V2V-Tx and jth V2V-Rx.

The received signal-to-interference-plus-noise ratio (SINR) at the ith V2V user is defined as

$$\gamma_i = \frac{p_i h_{ii}}{\sum_{j \in N}^{j \neq i} p_j h_{ji} + \sigma^2}, \tag{2}$$

where $p_i \in (p_i^{min}, p_i^{max})$ and σ are the transmission power at ith V2V pairs and the AWGN power, respectively.

In the power allocation problem in D2D-based V2V communications with N V2V pairs, our objective is to find an optimal policy to maximise the EE performance of our network. The information throughput at the ith V2V pair is defined as follows:

$$\begin{aligned}\psi_i =& W \ln(1 + \gamma_i) \\ =& W \ln\Big(1 + \frac{p_i h_{ii}}{\sum_{j \in N}^{j \neq i} p_j h_{ji} + \sigma^2}\Big)\end{aligned} \tag{3}$$

where W is a bandwidth. The total performance of the network is a joint function of all V2V pairs. We define the quality of service (QoS) constraints as

$$\gamma_i \geq \gamma_i^*, \forall i \in N. \tag{4}$$

In this work, we focus on maximising the total EE performance of the network while satisfying energy constraints and the QoS constraints for each V2V pair. Therefore, the EE optimisation problem can be defined as

$$\max \sum_{i}^{N} \frac{W}{p_i} \ln\left(1 + \frac{p_i h_{ii}}{\sum_{j\in N}^{j\neq i} p_j h_{ji} + \sigma^2}\right), \tag{5}$$

$$s.t\ \gamma_i \geq \gamma_i^*, \forall i \in N, \tag{6}$$

$$p_i^{min} \leq p_i \leq p_i^{max}. \tag{7}$$

In the D2D-based V2V communications, we have N V2V pairs in which each V2V pair can only have its environment information about power allocation strategy and current environment state. This makes the power allocation problem in D2D-based V2V communications become a multi-agent and non-cooperative game. Thus, we formulate the multi-agent power allocation game in D2D-based V2V communications and propose two deep reinforcement learning approaches based on the DDPG algorithm to enable each V2V user to have an optimal power allocation scheme.

In RL, an agent interacts with the environment to find the optimal policy through trial-and-error learning. We can formulate this task as a Markov decision process (MDP) [29]. In particularly, we define a 4-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}\rangle$, where $\mathcal{S}$ and $\mathcal{A}$ is the agent state space and action space, respectively. The reward function $r = \mathcal{R}(s, a, s')$ can be obtained at state $s \in \mathcal{S}$, action $a \in \mathcal{A}$, and next state $s' \in \mathcal{S}$. An agent has transition function $\mathcal{P}_{ss'}^{a}$ which is the probability of next states s' when taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$.

Regarding the multi-agent power allocation problem in D2D-based V2V communications, we define that each V2V transmitter is an agent, and the system consists of N agents. The ith V2V-Tx is defined as ith agent, which is represented as $\langle \mathcal{S}_i, \mathcal{A}_i, \mathcal{R}_i, \mathcal{P}_i\rangle$, where $\mathcal{S}_i$ is the environment state space, $\mathcal{A}_i$ is the action space, $\mathcal{R}_i$ is the reward function, and $\mathcal{P}_i$ is the state transition probability function. Generally, an agent corresponding to a V2V user at each time t observes a state, s^t from the state space, $\mathcal{S}$, then accordingly takes action of selecting power level, a^t, from the action space, $\mathcal{A}$ based on the policy, π. By taking the action a^t, the agent receives a reward, r^t and the environment transits to a new state s^{t+1}.

In the next step, we define the action spaces, state spaces and reward function of the multi-agent power allocation problem in D2D-based V2V communications as follows:

State spaces: At each time t, the state space of the ith V2V transmitter observed by the V2V link for characterising the environment is defined as

$$\mathcal{S}_i = \{i, \mathcal{I}_i\}, \tag{8}$$

where $\mathcal{I}_i \in (0, 1)$ is the level of interference as

$$\mathcal{I}_i = \begin{cases} 1 & \text{for } \gamma_i \geq \gamma_i^* \\ 0 & \text{for otherwise} \end{cases} \tag{9}$$

Action spaces: The agent i at time t takes an action a_i^t, which represents the agent selected power level, according to the current state, $s_i^t \in \mathcal{S}_i$ under the policy π_i. The action space of ith V2V-Tx is denoted as

$$\mathcal{A}_i = \{p_i\}, \tag{10}$$

where $p_i^{\min} \leq p_i \leq p_i^{\max}$.

Reward function: Our objective is to maximise the total performance of the network by interacting with the environments while satisfying the QoS constraints. Thus, we design a reward function $\mathcal{R}_i$ of the ith V2V user in state s_i by receiving the immediate return by executing action a_i as

$$\mathcal{R}_i = \begin{cases} \frac{W}{p_i}\ln(1+\gamma_i) \text{ if } \mathcal{I}_i = 1 \\ \qquad 0 \text{ if } \mathcal{I}_i = 0 \end{cases} \tag{11}$$

III. MULTI-AGENT POWER ALLOCATION PROBLEM IN D2D-BASED V2V COMMUNICATIONS: DISTRIBUTED DEEP DETERMINISTIC POLICY GRADIENT APPROACH

In RL, we have two main approaches and a hybrid model to solve the games. There are value function-based methods, policy search-based methods, and an actor-critic approach that employs both value functions and policy search [30]. In this section, we explain value function and policy search concepts which can learn on continuous domains. We further propose a solution based on the DDPG algorithm to solve the energy-efficient power allocation problem in D2D-based V2V communications.

A. VALUE FUNCTION

Value function, which is often denoted as $V^{\pi}(s)$, estimates the expected reward for an agent staring in state s and following the policy π subsequently. Value function represents how good for an agent to be in a given state

$$V^{\pi}(s) = \mathbb{E}\Big[\mathcal{R}|s_0 = s, \pi\Big], \tag{12}$$

where $\mathbb{E}(\cdot)$ stands for the expectation operation and $\mathcal{R}$ denotes the rewards gain from the initial state s while following the policy π. In all the possibility of the value function $V^{\pi}(s)$ there is an optimal value $V^*(s)$ corresponding to an optimal policy π^*; the optimal value function $V^*(s)$ can be defined as

$$V^*(s) = \max_{\pi} V^{\pi}(s), \quad s \in \mathcal{S}. \tag{13}$$

The optimal policy π^* is the policy that can be retrieved from optimal value function $V^*(s)$ by choosing the action a from the given state s to maximise the expected reward. We can rewrite (13) by using Bellman equation [31]

$$V^*(s) \ = V^{\pi^*}(s) = \max_{a\in\mathcal{A}} \Big[r(s,a) + \zeta \sum_{s'\in\mathcal{S}} p_{ss'}^{a} V^*(s')\Big], \tag{14}$$

where $r(s, a)$ is the expected reward obtain when taking action a from the state s, $p^a_{ss'}$ defines the probability of the next state s' if the agent at the state s takes action a, and $\zeta \in [0, 1]$ is the discounting factor.

The action-value function $Q^\pi(s, a)$ is the total reward which represents how good for an agent to pick an action a in state s when following the policy π

$$Q^\pi(s, a) = \mathbb{E}\Big[r(s, a) + \zeta\mathbb{E}[V^\pi(s')]\Big]. \tag{15}$$

The optimal action-value function $Q^*(s, a)$ can be written as

$$Q^*(s, a) = \mathbb{E}\Big[r(s, a) + \zeta \sum_{s'\in\mathcal{S}} p^a_{ss'} V^*(s')\Big]. \tag{16}$$

Thus, we have

$$V^*(s) = \max_{a\in\mathcal{A}} Q^*(s, a). \tag{17}$$

$$Q^*(s, a) = \mathbb{E}\Big[r(s, a) + \zeta \max_{a'\in\mathcal{A}} Q^*(s', a')\Big]. \tag{18}$$

Q-learning [32], an off-policy algorithm, regularly uses the greedy policy $\pi = \arg\max_{a\in\mathcal{A}} Q(s, a)$ to choose the action. The agent can achieve the optimal results by adjusting Q value according to the updated rule

$$Q(s, a) \leftarrow (1-\alpha)Q(s, a)+\alpha\Big[r(s, a)+\zeta \max_{a'\in\mathcal{A}} Q(s', a')\Big], \tag{19}$$

where $\alpha \in [0, 1]$ is the learning rate.

B. POLICY SEARCH

The policy gradient, which is one of policy search techniques, is a gradient-based optimisation algorithm. It aims to model and optimise the policy to directly search for an optimal behaviour strategy π^* for the agent. The policy gradient method is in popularity because of the efficient sampling ability when the number of policy parameters is large. Let π and θ_π denote the policy and vector of policy parameters, respectively; and J is the performance of the corresponding policy. The value of the reward function depends on this policy, and then the various algorithms can be applied to optimise parameter θ_π to achieve the optimal performance.

The average reward function on MDPs can be written as

$$J(\theta) = \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \pi_\theta(s, a; \theta_\pi)\mathcal{R}(s, a) \tag{20}$$

where $d(s)$ is the stationary distribution of Markov chain for policy π_θ. Using gradient ascent, we can adjust the parameter θ_π suggested by $\nabla_\theta J(\theta_\pi)$ to find the optimal θ^*_π that produces the highest reward. The policy gradient can be computed like in [33] as follows

$$\begin{aligned}\nabla_\theta J &= \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \pi_\theta(s, a; \theta_\pi)\nabla_\theta \log \pi_\theta(s, a; \theta_\pi)Q^\pi(s, a)\\ &= \mathbb{E}_{\pi_\theta}\Big[\nabla_{\theta_\pi} \log \pi_\theta(s, a; \theta_\pi)Q^\pi(s, a)\Big] \end{aligned}\tag{21}$$

The REINFORCE algorithm is devised as a Monte-Carlo policy gradient learning algorithm that relies on an estimated return by Monte-Carlo simulations where episode samples are used to update the policy parameter θ_π. The objective of REINFORCE algorithm is to maximise expected rewards under policy π

$$\theta^*_\pi = \arg\max_{\theta_\pi} J(\theta). \tag{22}$$

Thus, the gradient is presented as

$$\nabla\theta_\pi = \mathbb{E}_{\pi_\theta}\Big[\nabla_{\theta_\pi} \log \pi_\theta(s, a; \theta_\pi)Q^\pi(s, a)\Big], \tag{23}$$

Then, parameters are updated along positive gradient direction

$$\theta_\pi \leftarrow \theta_\pi + \alpha\nabla\theta_\pi \tag{24}$$

A drawback of the REINFORCE algorithm is the slow speed of convergence due to the high variance of the policy gradients.

C. DISTRIBUTED DEEP DETERMINISTIC POLICY GRADIENT

By utilising the advantages of both policy search-based methods and value function-based methods, a hybrid model called the actor-critic algorithm has grown as an effective approach [30]. In policy gradient-based methods, the policy function $\pi(a|s)$ is always modelled as a probability distribution over actions space $\mathcal{A}$ in the current state, and thus it is stochastic. Very recently, deterministic policy gradient (DPG) is deployed as an actor-critic algorithm in which the policy gradient theorem is extended from stochastic policy to deterministic policy. Inspired by the success of deep Q-learning [26], which uses neural network function approximation to learn value functions for a very large state and action space online, the combination of DPG and deep learning called deep deterministic policy gradient enables learning in continuous spaces.

An existing drawback of most optimisation algorithms is that the samples are assumed to be independently and identically distributed. It leads to the destabilisation and divergence of RL algorithms if we use a non-linear approximate function. To overcome that challenge, we use two major techniques as follows:

- *Experience replay buffer:* agent i has a replay buffer $\mathcal{D}_i$ to store the samples and take mini-batches for training. Transitions are sampled from the environment following the exploration policy and the tuple $(s^t_i, a^t_i, r^t_i, s^{t+1}_i)$ will be stored in $\mathcal{D}_i$. When the replay buffer $\mathcal{D}_i$ is big enough, a mini-batch K_i of transitions is sampled randomly from the buffer $\mathcal{D}_i$ to train the actor and critic network. By setting the finite size of replay buffer $\mathcal{D}$, the oldest samples are removed to retrieve space for the new samples, and the buffers are always up to date.
- *Target network:* At each step of training, the Q value is shifted. Thus, if we use a constantly shifting set of values to estimate the target value, the value estimations are easy out of control, and it makes the network unstable. To address this issue, we use a copy of the actor

and critic networks, $Q'_i(s_i, a_i; \theta_{q'_i})$ and $\mu'_i(s_i; \theta_{\mu'_i})$, respectively, to calculate the target values. The parameter $\theta_{q'_i}$ and $\theta_{\mu'_i}$ in actor and critic network are then updated using soft target updates with $\tau \ll 1$

$$\theta_{q'_i} \leftarrow \tau\theta_{q_i} + (1-\tau)\theta_{q'_i} \tag{25}$$

$$\theta_{\mu'_i} \leftarrow \tau\theta_{\mu_i} + (1-\tau)\theta_{\mu'_i} \tag{26}$$

By using the target networks, the target values are constrained to change slowly, significantly learning the action-value function closer to supervised learning. However, both target μ'_i and Q'_i are required to process a stable target in order to train the critic consistently without divergence. Herein, this may slow training since the target network delays the propagation of value estimations.

A notable challenge of learning in continuous action spaces is exploration [28]. In order to do better exploration, we add a small white noise $\mathcal{N}_i(0, 1)$ to our actor policy to construct a Gaussian exploration policy μ'_i [28]

$$\mu'_i(s_i^t) = \mu_i(s_i^t; \theta_{\mu_i}^t) + \epsilon\mathcal{N}_i(0, 1) \tag{27}$$

where ϵ is a small positive constant. The details of our proposed algorithm, distributed deep deterministic policy gradient based on DDPG algorithm, to deal with the multi-agent power allocation problem in D2D-based V2V communications are described in Algorithm 1.

Algorithm 1 Distributed Deep Deterministic Policy Gradient Algorithm for Multi-Agent Power Allocation Problem in D2D-Based V2V Communications

```
Initialisation:
for all V2V $i$, $i \in N$ do
    Randomly initialise critic $Q_i(s_i, a_i; \theta_{q_i})$ and actor $\mu_i(s_i; \theta_{\mu_i})$
    Randomly initialise targets $Q'_i$ and $\mu'_i$ with parameter $\theta_{q'_i} \leftarrow \theta_{q_i}, \theta_{\mu'_i} \leftarrow \theta_{\mu_i}$
    Initialise replay buffer $D_i$
end for
for all V2V $i$, $i \in N$ do
    for episode = $1, \ldots, M$ do
        Initialise the action exploration to a Gaussian $\mathcal{N}_i$
        Receive initial observation state $s_i^1$
        for iteration = $1, \ldots, T$ do
            Obtain the action $a_i^t$ at state $s_i^t$ according to the current policy and action exploration noise
            Measure the achieved SINR at the receiver according to (2)
            Update the reward $r_i^t$ according to (11)
            Observe the new state $s_i^{t+1}$
            Store transition $(s_i^t, a_i^t, r_i^t, s_i^{t+1})$ into replay buffer $D_i$
            Sample randomly a mini-batch of $K_i$ transitions $(s_i^k, a_i^k, r_i^k, s_i^{k+1})$ from buffer $D_i$
            Update critic by minimising the loss:
                $L_i = \frac{1}{K_i}\sum\left(y_i^k - Q_i(s_i^k, a_i^k; \theta_{qi})\right)^2,$    (28)
            where
                $y_i^k = r^k(s_i^k, a_i^k)$
                    $+\zeta Q'_i(s_i^{k+1}, a_i^{k+1}; \theta_{q'i})|_{a_i^{k+1}=\mu'(s_i^{k+1};\theta_{\mu'})}$    (29)
            Update the actor policy using the sampled policy gradient: $\nabla_{\theta_{\mu_i}} J_i \approx$
                $\frac{1}{K_i}\sum \nabla_{a_i^k} Q_i(s_i^k, a_i^k; \theta_{qi})|_{a_i^k=\mu_i(s_i^k)} \nabla_{\theta_{\mu_i}} \mu(s_i^k; \theta_{\mu_i})$    (30)
            Update the target networks:
                $\theta_{q'_i} \leftarrow \tau\theta_{q_i} + (1-\tau)\theta_{q'_i}$    (31)
                $\theta_{\mu'_i} \leftarrow \tau\theta_{\mu_i} + (1-\tau)\theta_{\mu'_i}$    (32)
            Update the state $s_i^t = s_i^{t+1}$
        end for
    end for
end for
```

IV. SHARING DEEP DETERMINISTIC POLICY GRADIENT FOR MULTI-AGENT POWER ALLOCATION PROBLEM IN D2D-BASED V2V COMMUNICATIONS

In this section, we present a simple improvement of the DDPG algorithm with the parameter sharing technique in multi-agent learning problems. In this algorithm, we can reach more effective policies for all the V2V pairs in the network by sharing the parameters of a single policy due to the homogeneous quality of all agents. Therein, each agent can be trained with the experiences of all agents simultaneously [34].

With the DDDPG algorithm in Algorithm 1, each agent has an actor network and a critic network for their own. It makes the systems shift significantly when the number of V2V pairs increases. In addition, the computational complexity, memory storage, and processing time are also unmanageable. Inspired by the impressive results of the paper [34], to overcome that problem, we propose a novel model based on DDPG called SDDPG algorithm in which a large number of agents can use sharing networks. By adding the embedding layer to build a new input layer of neural networks, we can use one actor and one critic network for the multiple agents in deep reinforcement learning. Consequently, it reduces the overall computational processing significantly in our model while ensuring the performance. The speed of convergence is also better than standard approaches.

The simplest way to represent an input layer with a node for every pair is "one-hot" encoding that is a vector of zeros with one at a single position. However when the number of V2V pairs in the network increases, the "one-hot" encoding vector becomes more sparse with relatively few non-zero values.

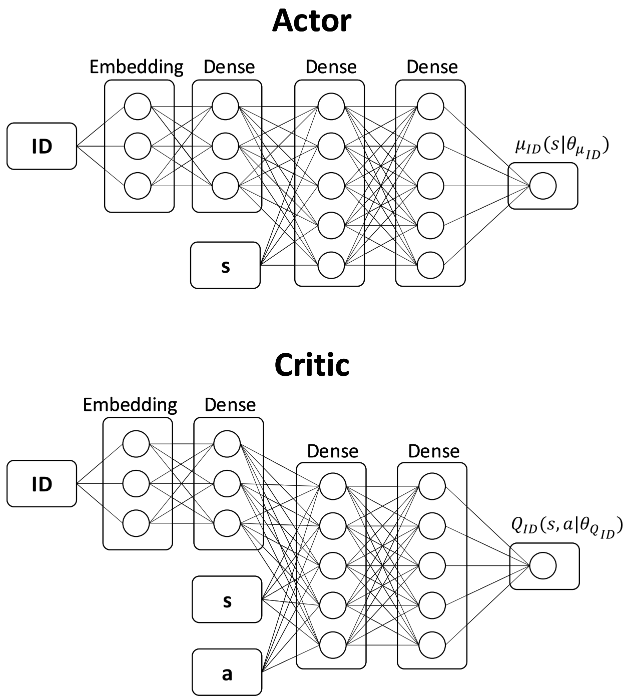


FIGURE 2. **Proposed model with sharing actor and critic network for multi-agent deep reinforcement learning problem.**

Thus, "one-hot" encoding has some issues such as the more data is needed to train the model effectively and the more parameters, the more computation is required to train and use the model; herein, it turns out that making a model more difficult to learn effectively and it is easy to exceed the capabilities of the hardware.

Embedding is rising as a potential technique in which a lower-dimensional space can be achieved by translating from a large sparse vector while preserving semantic relationships [35] to deal with these above problems. To apply the embedding layer efficiently in our problem, we divide the input of the ith V2V pair into two parts, ID i and QoS constraint $\mathcal{I}_i$. Depending on the number of V2V pairs in the network, the output dimension of embedding layers can be chosen flexibly to reduce the memory storage and processing time while ensuring the performance of the network. The ID i of the V2V pair is put into the embedding layer and a fully-connected layer before being concatenated with the level interference of the ith V2V pair, $\mathcal{I}_i$. We assume that the concatenated layer is the input of neural networks in the DDPG algorithm. The actor and critic network of our proposed model are described in Fig. 2.

The details of the SDDPG algorithm-based approach for multi-agent power allocation problem in D2D-based V2V communications are described in Algorithm 2.

Algorithm 2 Sharing Deep Deterministic Policy Gradient for Multi-Agent Power Allocation Problem in D2D-Based V2V Communications

```
Initialisation:
Initialise the critic network $Q(s, a; \theta_q)$ and actor network $\mu(s; \theta_\mu)$ with random parameter $\theta_q$ and $\theta_\mu$
Initialise the target networks $Q'$ and $\mu'$ with parameter $\theta_{q'} \leftarrow \theta_q, \theta_{\mu'} \leftarrow \theta_\mu$
Initialise replay buffer $D$
for all V2V $i, i \in N$ do
  for episode $= 1, \ldots, M$ do
    Initialise the embedding layer
    Initialise a random process $\mathcal{N}_i$ for action exploration
    Receive initial observation state $s_i^1$ by concatenating the output of the embedding layers and $\mathcal{I}_i$
    for iteration $= 1, \ldots, T$ do
      Obtain the action $a_i^t$ at state $s_i^t$ according to the current policy and exploration noise
      Measure the achieved SINR at the receiver according to (2)
      Update the reward $r_i^t$ according to (11)
      Observe the new state $s_i^{t+1}$
      Store transition $(s_i^t, a_i^t, r_i^t, s_i^{t+1})$ into replay buffer $D$
      Sample randomly a mini-batch of $K$ transitions $(s_i^k, a_i^k, r_i^k, s_i^{k+1})$ from buffer $D$
      Update critic by minimising the loss:
        $$L = \frac{1}{K} \sum \left( y^k - Q(s_i^k, a_i^k; \theta_q) \right)^2 \quad (33)$$
      where
        $$y^k = r(s_i^k, a_i^k) + \zeta Q'(s_i^{k+1}, a_i^{k+1}; \theta_{q'})|_{a_i^{k+1} = \mu'(s_i^{k+1}; \theta_{\mu'})} \quad (34)$$
      Update the actor policy using the sampled policy gradient: $\nabla_{\theta_\mu} J \approx$
        $$\frac{1}{K} \sum \nabla_{a_i^k} Q(s_i^k, a_i^k; \theta_q)|_{a_i^k = \mu_i(s_i^k)} \nabla_{\theta_{\mu_i}} \mu(s_i^k; \theta_{\mu_i}) \quad (35)$$
      Update the target networks:
        $$\theta_{q'} \leftarrow \tau\theta_q + (1 - \tau)\theta_{q'} \quad (36)$$
        $$\theta_{\mu'} \leftarrow \tau\theta_\mu + (1 - \tau)\theta_{\mu'} \quad (37)$$
      Update the state $s_i^t = s_i^{t+1}$
    end for
  end for
end for
```

V. SIMULATION RESULTS

In this section, we perform the simulation results on PC Intel(R) Core(TM) i7-8700 CPU @ 3.20Ghz to demonstrate the effectiveness of our proposed methods in solving the power control problem in D2D-based V2V communications. Tensorflow version 1.13.1 [36] is used to implement all

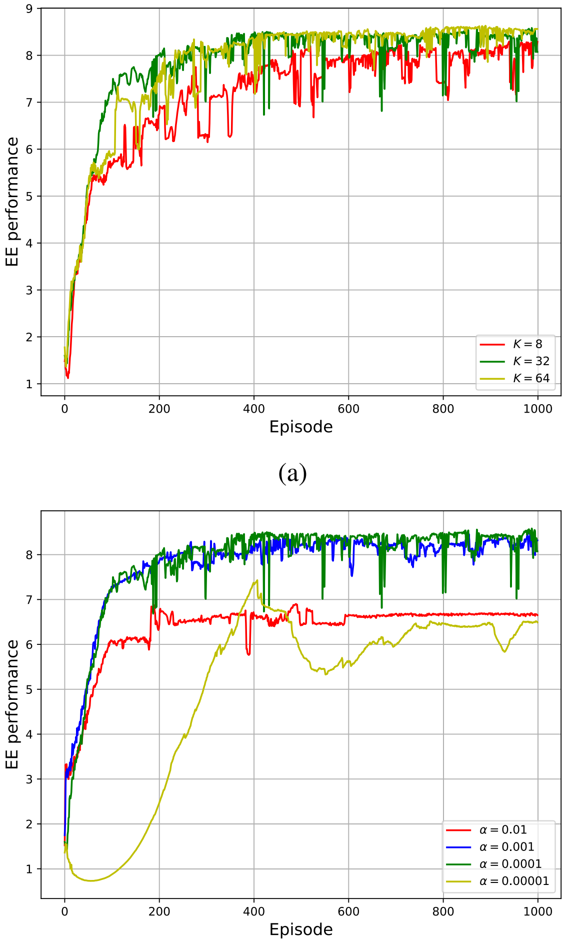
TABLE 1. **Simulation parameters.**

Parameters	Value
Bandwidth (W)	1 MHz
Path-loss exponent	$\alpha_h = 3$
Maximum V2V transmit power	$p_{max} = 23$ dBm
Minimum V2V transmit power	$p_{min} = 0$ dBm
Channel power gain at the reference	$\beta_0 = -30dB$
Noise power density	$\eta = 0.5$
V2V connection SINR QoS constraint	$\gamma^* = 0$ dB
Actor network learning rate	$\alpha_A = 0.0001$
Critic network learning rate	$\alpha_C = 0.0001$
Soft replacement	$\tau = 0.01$
Discount factor	$\zeta = 0.9$
Memory pool capacity	$D = 10000$
Mini-batch size	$K = 32$

algorithms. We design the actor and critic networks with one input layer, one output layer, one hidden layer of 100 units and Adam optimisation algorithm [37] for training. The parameters of neural networks are initialised with small random values with a zero-mean Gaussian distribution. The other simulation parameters are given in Table 1.

Fig. 3 illustrates the EE performance of the network using the DDDPG algorithm while considering different values of mini-batch size K and the learning rate of actor and critic network, α_A and α_C, respectively. From Fig. 3 (a), we can see that with a small batch size, our proposed algorithms can be needed to take a long time to reach the optimal policy. On the other hand, there is a possibility that the learning process can be trapped in local optimum and cannot escape to reach the best performance if a batch size is too large, although the calculated gradient is more accurate than the ones with a small batch size; hence, it may lead to a slower convergence. Meanwhile, the parameters of neural networks are updated according to the value of the learning rate. The learning rate decides the speed of convergence and stability of our proposed algorithms. In Fig. 3 (b) with the small values of the learning rate, results are at a slower speed of convergence. On the contrary, if we choose a high learning rate, the algorithms can diverge from the optimal solution. Clearly, our proposed algorithms can achieve the best performance with the learning rate, $\alpha_A = 0.0001$ and $\alpha_C = 0.0001$. Based on the result shown in Fig. 3, we choose the batch size to be $K = 32$ and the initial learning rate $\alpha = 0.0001$ for actor and critic networks.

Fig. 4 compares the performance of our two proposed approaches based on the DDDPG and SDDPG algorithm with the output dimension of the embedding layer set to 5, $|Dims| = 5$. The comparison is against the standard DDPG algorithm for a multi-agent power allocation problem in D2D-based V2V communications. The EE performance of the network when using the DDDPG and SDDPG algorithm are almost identical and better than the standard DDPG in multi-agent learning. In convergence, the speed of convergence with the SDDPG algorithm is faster than ones with the DDDPG algorithm and ones with the standard DDPG algorithm. The reason is that when we use sharing networks for N agents, these networks are trained many times and the next



(a)

(b)

FIGURE 3. **The EE performance of the network by using the DDDPG algorithm in multi-agent power allocation problem in D2D-based V2V communications with different values of batch size K and learning rate α, the number of V2V pairs, $N = 30$.**

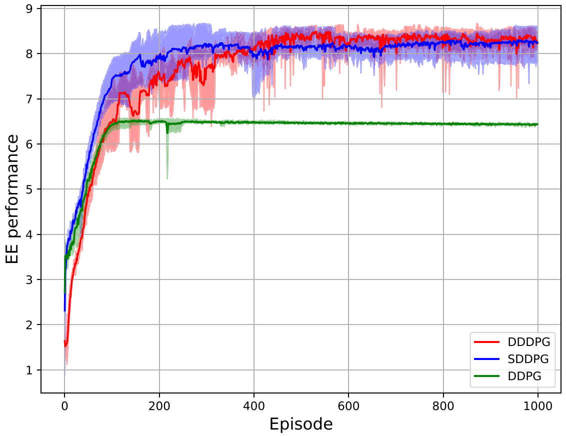


FIGURE 4. **The EE performance of the network by using the DDDPG, SDDPG and DDPG algorithm in multi-agent power allocation problem in D2D-based V2V communications with the number of V2V pairs, $N = 30$.**

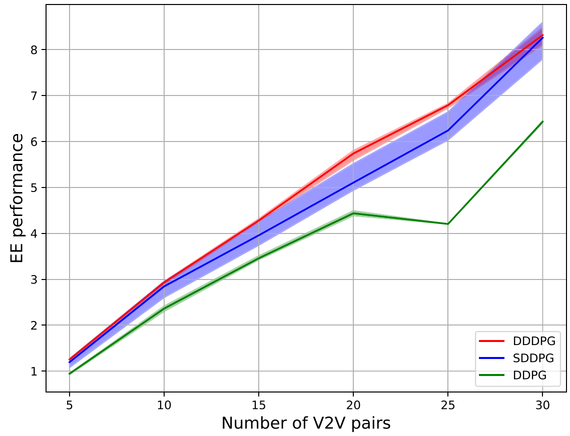


FIGURE 5. Performance results of the DDDPG, SDDPG and DDPG algorithm-based approaches with different number of V2V pairs in the network.

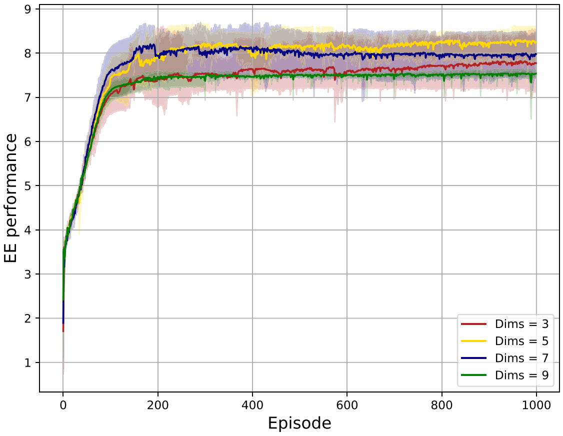


FIGURE 6. The EE performance of the network by using the SDDPG algorithm with different output dimensions of embedding layer in multi-agent power allocation problem in D2D-based V2V communications with the number of V2V pairs, $N = 30$.

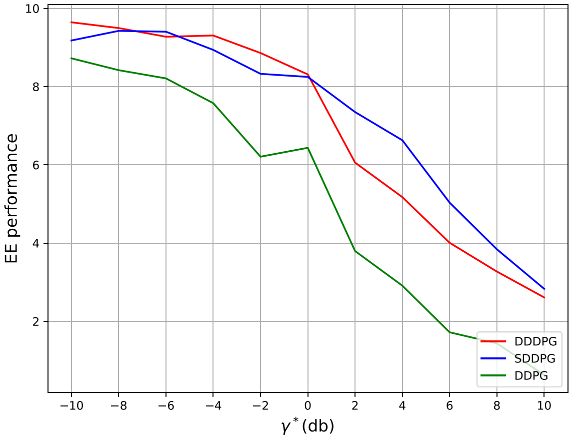


FIGURE 7. The EE performance of the network by using the DDDPG, SDDPG and DDPG algorithm in multi-agent power allocation problem in D2D-based V2V communications while considering different values of SINR requirement, γ^*.

agents can use the previous pre-trained networks to achieve an optimal policy faster than the DDDPG algorithm-based approach. These results advise that using the combination of multi-agent learning and the DDPG algorithm significantly helps to find an optimal policy for non-cooperative energy-efficient power allocation problem in D2D-based V2V communications.

Fig. 5 plots the EE results of the network with different number of V2V pairs by using the DDDPG, SDDPG, and standard DDPG algorithm. The output dimensions of the embedding layer in the SDDPG algorithm for $N = 5$, $N = 10$, $N = 15$, $N = 20$, $N = 25$, $N = 30$ are $Dims = 2$, $Dims = 3$, $Dims = 3$, $Dims = 4$, $Dims = 4$ and $Dims = 5$, respectively. The performance of the network by using the DDDPG and SDDPG algorithm-based approaches outperform with ones based on the classical DDPG algorithm in different number of V2V pairs. The simulation result difference between models based on the DDDPG and SDDPG algorithm is small even when the number of V2V pairs increases. With $N = 30$, the average performances of the DDDPG and SDDPG algorithm-based approaches are almost identical. The performance of the scheme based on the SDDPG algorithm is better than the DDDPG algorithm in some cases. However, the DDDPG algorithm uses N neural networks for actor function and N neural networks for critic function. Meanwhile, in the SDDPG algorithm, we share one actor network and one critic network for all the agents. Therefore, the computational processing and memory storage used for the DDDPG algorithm-based approach is many times higher than the SDDPG algorithm when the number of V2V pairs increases.

Next, we compare EE performance results of the network using the SDDPG algorithm-based approach in different output dimensions of the embedding layer in Fig. 6. With the number of V2V pairs in the network $N = 30$, we can achieve the best performance while setting the output dimension of the embedding layer to $Dims = 5$. The higher output dimensions in the embedding layer do not guarantee better performance. However, the variance of models with the higher output dimensions of the embedding layer is lower.

Moreover, in Fig. 7, we present the performance of the network with different values of SINR requirements γ^* in models using the DDDPG, SDDPG, and classical DDPG algorithm. As we can see from Fig. 7, when the value of SINR requirement γ^* is too high, the EE result degrades due to the decease in the number of V2V links that satisfy QoS requirements. The performance of the SDDPG algorithm-based approach is better than the ones using the DDDPG algorithm when we choose the γ^* high. In addition, the effectiveness of our proposed algorithms, SDDPG and DDDPG, is superior to the classical DDPG algorithm for multi-agent power control problem in D2D-based V2V communications.

Finally, we evaluate the processing time of our proposed models during test time after neural networks being trained in comparison with other approaches. Table 2 presents the average processing time in different scenarios. By using our proposed models, each V2V user can choose the power level to

TABLE 2. The running time of our proposed models in comparison with other approaches.

	DDDPG	SDDPG	DDPG	OPA [12]
$N = 5$	4.16ms	5.78ms	4.12ms	121.1ms
$N = 15$	8.05ms	11.61ms	7.36ms	130.8ms
$N = 30$	21.1ms	28.07ms	19.89ms	145.6ms

maximise the EE performance of the network in milliseconds while satisfying QoS requirements. Particularly, we only need 21.1ms and 28.07ms to solve the power allocation problem in D2D-based V2V communications with the number of V2V pairs, $N = 30$, by using the DDDPG and SDDPG algorithm, respectively. On the other hand, the method based on the logarithmic inequality algorithm in [12] needs 145.6ms to solve a similar problem with the same environment parameters. Therefore, the results suggest that our proposed models are promising techniques for real-time scenarios.

VI. CONCLUSION

In this paper, we proposed two models based on the DDDPG and SDDPG algorithm to solve the multi-agent energy-efficient power allocation problem in D2D-based V2V communications. By utilising the advantage of neural networks and the embedding layer, our proposed models can overcome the limitations of existing approaches. The simulation results outperformed other base-line algorithms in terms of the EE performance of the network, computational complexity, and memory storage. The computational complexity and memory storage of the solution can be significantly reduced by using the SDDPG algorithm when the number of V2V pairs increases. In the future, we will investigate more efficient multi-agent learning approaches and more advanced deep learning models in order to improve the learning convergence, reduce the training variance, and reduce the algorithm's computational complexity.

REFERENCES

[1] X. Huang, R. Yu, J. Liu, and L. Shu, "Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications," *IEEE Access*, vol. 6, pp. 66649–66663, 2018.

[2] X. Huang, R. Yu, M. Pan, and L. Shu, "Secure roadside unit hotspot against eavesdropping based traffic analysis in edge computing based Internet of vehicles," *IEEE Access*, vol. 6, pp. 62371–62383, 2018.

[3] C. Kai, H. Li, L. Xu, Y. Li, and T. Jiang, "Energy-efficient device-to-device communications for green smart cities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1542–1551, Apr. 2018.

[4] N.-S. Vo, L. D. Nguyen, T. Q. Duong, and M.-N. Nguyen, "Optimal video streaming in dense 5G networks with D2D communications," *IEEE Access*, vol. 6, pp. 209–223, 2017.

[5] A. Masaracchia, L. D. Nguyen, T. Q. Duong, and M.-N. Nguyen, "An energy-efficient clustering and routing framework for disaster relief network," *IEEE Access*, vol. 7, pp. 56520–56532, 2019.

[6] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, Nov. 2014.

[7] W. Sun, D. Yuan, E. Ström, and F. Brännström, "Cluster-based radio resource management for D2D-supported safety-critical V2X communications," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2756–2769, Apr. 2016.

[8] Y. Ren, F. Liu, Z. Liu, C. Wang, and Y. Ji, "Power control in D2D-based vehicular communication networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5547–5562, Dec. 2015.

[9] W. Sun, E. G. Ström, F. Brännström, K. C. Sou, and Y. Sui, "Radio resource management for D2D-based V2V communication," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6636–6650, Aug. 2016.

[10] Z. Sheng, H. D. Tuan, A. A. Nasir, T. Q. Duong, and H. V. Poor, "Power allocation for energy efficiency and secrecy of wireless interference networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3737–3751, Jun. 2018.

[11] H. Gao, S. Zhang, Y. Su, and M. Diao, "Joint resource allocation and power control algorithm for cooperative D2D heterogeneous networks," *IEEE Access*, vol. 7, pp. 20632–20643, 2019.

[12] M.-N. Nguyen, L. D. Nguyen, T. Q. Duong, and H. D. Tuan, "Real-time optimal resource allocation for embedded UAV communication systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 225–228, Feb. 2019.

[13] J. Lee and J. H. Lee, "Performance analysis and resource allocation for cooperative D2D communication in cellular networks with multiple D2D pairs," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 909–912, May 2019.

[14] S. Liu, Y. Wu, L. Li, X. Liu, and W. Xu, "A two-stage energy-efficient approach for joint power control and channel allocation in D2D communication," *IEEE Access*, vol. 7, pp. 16940–16951, 2019.

[15] P. Zhang, A. Y. Gao, and O. Theel, "Bandit learning with concurrent transmissions for energy-efficient flooding in sensor networks," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 4, no. 13, pp. 1–14, Mar. 2018.

[16] L. D. Nguyen, A. Kortun, and T. Q. Duong, "An introduction of real-time embedded optimisation programming for uav systems under disaster communication," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 5, no. 17, pp. 1–8, Dec. 2018.

[17] M. T. Nguyen, H. M. Nguyen, A. Masaracchia, and C. V. Nguyen, "Stochastic-based power consumption analysis for data transmission in wireless sensor networks," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 6, no. 19, pp. 1–11, Jun. 2019.

[18] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting decode-and-forward two-hop communications," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 3, pp. 309–319, Sep. 2017.

[19] M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for NOMA based heterogeneous IoT with imperfect SIC," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2885–2894, Apr. 2019.

[20] M. Liu, T. Song, J. Hu, J. Yang, and G. Gui, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 641–653, Jan. 2019.

[21] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.

[22] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink MIMO," *IEEE Access*, vol. 7, pp. 7599–7605, 2019.

[23] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.

[24] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.

[25] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions," Apr. 2019, *arXiv:1904.09673*. [Online]. Available: https://arxiv.org/abs/1904.09673

[26] K. K. Nguyen, T. Q. Duong, N. A. Vien, N.-A. Le-Khac, and N. M. Nguyen, "Non-cooperative energy efficient power allocation game in D2D communication: A multi-agent deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 100480–100490, 2019.

[27] 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.

[28] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," Sep. 2015, *arXiv:1509.02971*. [Online]. Available: https://arxiv.org/abs/1509.02971

[29] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 1994.

[30] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[31] D. P. Bertsekas, *Dynamic Programming Optimization Control*, vol. 1, no. 2. Belmont, MA, USA: Athena Scientific, 1995.

[32] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

[33] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.

[34] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, Nov. 2017, pp. 66–83.

[35] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1019–1027.

[36] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX conf. OSDI*, Nov. 2016, pp. 265–283.

[37] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

KHOI KHAC NGUYEN was born in Bac Ninh, Vietnam. He received the B.S. degree in information and communication technology from the Hanoi University of Science and Technology (HUST), Vietnam, in 2018. He is currently pursuing the Ph.D. degree with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, U.K. His research interests include machine learning and deep reinforcement learning for real-time optimization in wireless networks, and the massive Internet of Things (IoT).

TRUNG Q. DUONG (S'05–M'12–SM'13) received the Ph.D. degree in telecommunications systems from the Blekinge Institute of Technology (BTH), Sweden, in 2012. From 2013 to 2017, he was a Lecturer (an Assistant Professor) with Queen's University Belfast, U.K., where he has been a Reader (an Associate Professor), since 2018. He is the author or a coauthor of over 350 technical articles published in scientific journals (210 articles) and presented at international conferences (140 articles). His current research interests include wireless communications, signal processing, Internet of Things (IoT), machine learning, and big data analytics. He received the Best Paper Award at the IEEE Vehicular Technology Conference (VTC-Spring), in 2013, the IEEE International Conference on Communications (ICC), in 2014, the IEEE Global Communications Conference (GLOBECOM), in 2016, and the IEEE Digital Signal Processing Conference (DSP), in 2017. He was a recipient of the prestigious Royal Academy of Engineering Research Fellowship (2016–2021) and the prestigious Newton Prize 2017. He also serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE TRANSACTIONS ON COMMUNICATIONS and a Lead Senior Editor for the IEEE COMMUNICATIONS LETTERS.

NGO ANH VIEN received the B.S. degree in computer engineering from the Hanoi University of Science and Technology, Vietnam, in 2005, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2009. He was a Postdoctoral Researcher with the National University of Singapore, from 2009 to 2011, the Ravensburg-Weingarten University of Applied Sciences, Germany, from 2011 to 2013, and the Machine Learning and Robotics Laboratory, University of Stuttgart, from 2013 to 2017. He has been a Lecturer (an Assistant Professor) with Queen's University Belfast, U.K., since 2017. His research interests include machine learning and robotics.

NHIEN-AN LE-KHAC received the Ph.D. degree from the Institut polytechnique de Grenoble, France. He is currently a Lecturer/Assistant Professor with the School of Computer Science, University College Dublin (UCD). He is also the Director of UCD Forensic Computing and Cybercrime Investigation Program—an international M.Sc. program for the law enforcement officers specializing in cybercrime investigation. To date, more than 1000 students from 76 countries graduated from this program. He has published more than 100 scientific articles in international peer-reviewed journal and conferences in related disciplines. He received the prestigious UCD School of Computer Science Outstanding Teaching Award, in 2018. He is also a Principal Investigator of EU DG-Home Research Grant. He is also a Science Foundation of Ireland (SFI) Co-Principal Investigator and a Funded Investigator.

LONG D. NGUYEN was born in Vietnam. He received the B.S. degree in electrical and electronics engineering and the M.S. degree in telecommunication engineering from the Ho Chi Minh City University of Technology (HCMUT), Vietnam, in 2013 and 2015, respectively, and the Ph.D. degree in electronics and electrical engineering from Queen's University Belfast (QUB), U.K., in 2018. He was a Research Fellow with Queen's University Belfast, U.K., for a part of the Newton Project, from 2018 to 2019. He is currently an Adjunct Assistant Professor with Duy Tan University, Vietnam. His research interests include convex optimization techniques for resource management in wireless communications, energy efficiency approaches (heterogeneous networks, relay networks, cell-free networks, and massive MIMO), and real-time embedded optimization for wireless networks and the Internet of Things (IoT).