IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE FOR PHYSICAL-LAYER WIRELESS COMMUNICATIONS

Received June 20, 2019, accepted July 15, 2019, date of publication July 19, 2019, date of current version August 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930115

Non-Cooperative Energy Efficient Power Allocation Game in D2D Communication: A Multi-Agent Deep Reinforcement Learning Approach

KHOI KHAC NGUYEN[1], TRUNG Q. DUONG[1], (Senior Member, IEEE), NGO ANH VIEN[1], NHIEN-AN LE-KHAC[2], AND MINH-NGHIA 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, D04 V1W8 Ireland
[3]Institute of Research and Development, 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 Newton Prize 2017 and the Newton Fund Institutional Link through the Fly-by Flood Monitoring Project under Grant ID 428328486, which is delivered by the British Council.

ABSTRACT Recently, there is the widespread use of mobile devices and sensors, and rapid emergence of new wireless and networking technologies, such as wireless sensor network, device-to-device (D2D) communication, and vehicular ad hoc networks. These networks are expected to achieve a considerable increase in data rates, coverage, and the number of connected devices with a significant reduction in latency and energy consumption. Because there are energy resource constraints in user's devices and sensors, the problem of wireless network resource allocation becomes much more challenging. This leads to the call for more advanced techniques in order to achieve a tradeoff between energy consumption and network performance. In this paper, we propose to use reinforcement learning, an efficient simulation-based optimization framework, to tackle this problem so that user experience is maximized. Our main contribution is to propose a novel non-cooperative and real-time approach based on deep reinforcement learning to deal with the energy-efficient power allocation problem while still satisfying the quality of service constraints in D2D communication.

INDEX TERMS Energy efficient wireless communication, power allocation, D2D communication, multi-agent reinforcement learning, deep reinforcement learning.

I. INTRODUCTION

With a fast-growing number of mobile devices and sensors, wireless networks (e.g., heterogeneous networks (HetNets), ultra-dense networks, and unmanned aerial vehicle (UAV) networks) become more autonomous, complex and dynamic in nature. At the same time, it incurs a fast escalation of energy demand and requirements for efficient resource allocation. In other words, the critical problem for wireless network energy efficiency is the trade-off between energy consumption and guaranteed performance (e.g., throughput or quality of service (QoS)). Technologies towards energy-efficient wireless networks have established a long research history in which most studies are focused on energy-efficient data communications [1]. To obtain better performance, a substantial effort has been spent on optimization theory to develop more efficient algorithms to gain optimal or near-optimal solutions. However, many previous works assume a static network environment. In advanced wireless networks with an enormous number of devices, the environment is often dynamically unstable. Therefore, it is desirable to enable network nodes to have autonomous decision-making ability. The decisions must be based directly on local observations, e.g., power allocation, spectrum access, and interference management, to maximize the network performance.

Reinforcement learning (RL) [2] is a sub-field of machine learning which offers a mathematically principled framework

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

studying how an autonomous agent makes optimal sequential decisions. An RL-based agent learns to make optimal decisions through trial-and-error interactions directly with a dynamic environment in which the objective is to maximize the task's performance measure, e.g. user experience. In an interactive learning fashion, the agent takes actions and observes the results of the interactions to make decisions. Over the last 20 years, there have been many successful applications of RL ranging from robotics [3], natural language processing [4] to game solving [5]. One of the most famous applications of RL is AlphaGo [6], the first ever computer program which can beat a world-class professional player in the *Go* board game.

In the area of wireless communications, RL has recently emerged as a powerful tool to deal with problems in modern networks. However, under the uncertain, stochastic, and large-scale environment, the computational complexity of existing techniques grows exponentially and becomes unmanageable. This makes existing learning approaches slower to converge as the agent has to explore the whole state space repeatedly in order to search for the best policy for an entire system. Consequently, standard RL approaches become inefficient and impractical in complex domains. Recently, deep learning [7] techniques have brought revolutionary advances in computer vision. Deep learning is a sub-field of machine learning concerned with flexible and scalable algorithms to optimize complex functions on a very large dataset. Over the past few years, deep learning has been applied successfully in many areas, for example, face recognition [8]–[11], cybersecurity [12], and game playing [6]. A combination of deep learning and RL leads to a new platform, called deep reinforcement learning (DRL) [13].

This work proposes to apply DRL to optimize power allocation in D2D communication. In particular, we develop a single-agent reinforcement learning and multi-agent reinforcement learning that are able to obtain an optimal policy. We then propose three different low complexity algorithms which are based on i) deep Q-learning, ii) double deep Q-learning, and iii) dueling deep Q-learning. These proposed approaches can solve the non-cooperative game in D2D communication in milliseconds. Our major contributions of this paper can be summarized as follows,

- We propose a novel method based on deep reinforcement learning for power allocation problem in D2D communication. Based on this method, each D2D transmitter can optimize its power used for information transmission in order to adapt to the dynamics of the environment. Our method will be based on deep Q-networks, double deep Q-networks, and dueling deep Q-networks algorithms.
- We directly tackle the non-cooperative problem in D2D communication. We assume that each D2D pair in the network is parsimonious and unaware of other D2D pairs' power allocation schemes and conditions. After a certain period of time, all D2D pairs broadcast their strategies to peers in order to calculate the overall performance of the network.
- We perform extensive experiments with the aim not only to demonstrate the efficiency of the proposed solution in comparison with other conventional methods but also to provide running time results in details.

The remainder is organized as follows. Section II discusses related work using reinforcement learning and deep learning for physical layer and resource allocation. The system model and the resource allocation problem in D2D communications are described in Section III. Section IV describes single-agent reinforcement learning and multi-agent reinforcement learning. In Section V, we show how we adapt existing RL algorithms to solve the non-cooperative resource allocation problem. In Section VI, we present numerical simulations to verify the proposed schemes. Finally, Section VII concludes this paper.

II. RELATED WORKS

Energy efficient power allocation is a challenging problem in modern wireless networks that often has many QoS constraints. In particular, D2D communication is always restricted by computational processing and limited knowledge about the environment. Several solutions have been proposed to tackle these problems, for example [14]–[18]. In [14], the author proposed a joint link scheduling and power allocation optimization algorithm to maximize the system throughput of D2D-assisted wireless caching networks. However, this work assumes that the network has a single channel in order to simplify the analysis and implementation. In addition, it assumes a centralized control mechanism, therefore it can only result in sub-optimal solutions. Moreover, this method requires all data to be collected by the base station and thus introduces a time delay in the network. An alternative algorithm is based on the logarithmic inequality to find an optimal resource allocation strategy for UAV communication systems [17]. This algorithm shows promising performance in terms of both the sum rate and running time. However, it requires all data to be processed in a centralized mode, therefore the running time increases rapidly as the number of D2D pairs increases. Very recently, the real-time optimization has been considered to increase the running time of optimization for radio resource (see, e.g., [17], [19] and references therein)

Over the past few years, RL has been applied widely in wireless networks to allow each node possess its own self-organizing function, which leads to a distributed control mode. For examples, the work in [20] proposes a cooperative Q-learning method to deal with a resource allocation problem in heterogeneous wireless networks to maximize the sum capacity of the network while being able to ensure QoS and fairness to users. Regarding problems in D2D communication, the work in [21] proposes two approaches using RL: convergence-based and epsilon-greedy Q-learning. Their objective is to find an optimal energy-efficient power allocation policy for the energy harvesting problem.

With the emergence of neural networks, deep learning algorithms have received considerable attention in the field

of wireless communications [22]–[28] because they can offer a powerful optimization tool for non-convex, non-differentiable and complex objective functions. It works by constructing a very big neural network and adjusting the network's weights using stochastic gradient descent, i.e. using *mini-batch* (a small subset of all available data). In the resource allocation problem, deep learning combined with damped three-dimensional message-passing can be used to minimize the weighted sum of the secondary interference power in cognitive radio network [27]. Another work using deep recurrent neural networks [28] proposed to look for an optimal resource allocation scheme for the mobile users and the IoT users in the non-orthogonal multiple access (NOMA)-based heterogeneous IoT. These deep learning-based algorithms are shown to achieve optimal solutions with a reasonably fast convergence rate and low computational complexity. Deep learning has recently become a popular research topic in physical layer as well [26]. The work in [22] used convolutional neural networks to improve the accuracy of automatic modulation recognition in the cognitive radio.

Deep neural networks have also been applied to solve many existing problems in massive multiple-input multiple-output (MIMO) very efficiently [23], [24]. Long short-term memory networks are also used in [25] to detect channel characteristics automatically. After being optimized offline, the trained neural network-aided NOMA system can achieve much improved performance in terms of the error rate and sum data rate. In comparison to conventional approaches, deep learning-based methods have shown better performance and improved computation in many domains such as beamforming design method and hybrid precoding. In addition, it shows great potential to applications in real-time scenarios.

In recent years, DRL has received much interest as an alternative and powerful tool to deal with complex optimization problems in wireless network [29], [30]. The work in [29] proposed a DRL based joint mode selection and resource management approach to optimize the system power consumption for green fog radio access networks. Each user equipment is assumed to operate in either cloud RAN or D2D mode. The resource to be managed includes both the radio resource and the computing resource. Through DRL, this work shows that the network controller can be optimized to make optimal decisions under the dynamics of edge caching states.

DRL is also applied in vehicle-to-vehicle communication (V2V) [30] to find an optimal sub-band and power level for transmission in every link. Each V2V link is considered as an RL agent in which the spectrum and transmission power are considered as actions and optimized depending on all information as environment observations such as: local channel state information, interference levels, the resource allocation decision of instantaneous channel conditions, and exchanged information shared from the neighbors at each time slot. In principle, the RL agent learns autonomously how to balance between minimizing the interference of V2V links to the vehicle-to-infrastructure networks and satisfying the stringent latency constraints imposed on V2V link.

However, most existing approaches assume data samples are independently and identically distributed. Furthermore, each node is assumed knowing about other nodes' condition and their resource allocation strategies. Moreover, these approaches are still non-trivial to scale to real-time scenarios due to the high computational complexity. In this paper, we present a non-cooperative energy efficient approach based on DRL. By utilizing the advantages of deep learning, the proposed algorithms can solve the non-cooperative game and enable each D2D transmitter to have a self-organizing function to adapt to the dynamic environment in milliseconds.

III. SYSTEM MODEL AND PROBLEMS FORMULATION

We consider a communication system that includes N D2D pairs, where D2D transmitter (D2D-Tx) and D2D receiver (D2D-Rx) are randomly distributed within the coverage of one base station (BS). Each D2D-Tx/D2D-Rx is equipped with a single antenna. We denote D2D(i) as the ith D2D pair, where $i \in \{1, 2, \ldots, N\}$.

We assume that the location of D2D-Tx and D2D-Rx are $(x^n_{\mathrm{Tx}}, y^n_{\mathrm{Tx}})$ and $(x^n_{\mathrm{Rx}}, y^n_{\mathrm{Rx}})$, respectively. The distance between the ith D2D-Tx and D2D-Rx pair is calculated using Euclidean distance as

$$R_i = \sqrt{(x^i_{\mathrm{Tx}} - x^i_{\mathrm{Rx}})^2 + (y^i_{\mathrm{Tx}} - y^i_{\mathrm{Rx}})^2}. \tag{1}$$

The channel power gain between D2D-Tx and D2D-Rx is defined as

$$h_{ii} = \beta_0 p_i^2 R_i^{-\alpha_h}, \tag{2}$$

where β_0 is the channel power gain at the reference distance d_0, p_i is an exponentially distributed random variable with unit mean, R_i is the distance between the ith D2D-Tx and D2D-Rx pair, and α_h is the path loss exponent for D2D links.

The total interference plus noise at each D2D user includes the interferences from all D2D-Txs and the additive white Gaussian noise (AWGN). We use p_i ($p_i^{min} \le p_i \le p_i^{max}$) and γ_i to designate the transmission power and received signal-to-interference-plus-noise ratio (SINR) at D2D i, respectively. The SINR at the ith D2D user is written as

$$\gamma_i = \frac{p_i h_{ii}}{\sum_{j\in N}^{j\neq i} p_j h_{ji} + \sigma^2}, \tag{3}$$

where σ is the AWGN's power.

The goal of power allocation is to ensure that no SINR falls below its threshold γ_i^* which is chosen to guarantee the QoS constraint as

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

We focus on a system performance that encompasses the D2D pair capacity. This suggests us to define a reward function that can be used in a RL algorithm later as follows:

$$\mathcal{R}_i = \frac{W \ln(1 + \gamma_i)}{p_i}, \tag{5}$$

where W is a bandwidth. The reward of each D2D user is a function of the joint actions of all D2D pairs.

Formally, the power allocation game can be defined as a constrained optimization problem as follows:

$$\max \sum_{i}^{N} \frac{W}{p_i} \ln\left(1 + \frac{p_i h_{ii}}{\sum_{j\neq i}^{j\in N} p_j h_{ji} + \sigma^2}\right), \tag{6}$$

$$s.t\ \gamma_i \geq \gamma^*, \quad \forall i \in N, \tag{7}$$

$$p_i^{min} \leq p_i \leq p_i^{max}. \tag{8}$$

There are some challenges solving the above optimization problem because of the fact that every D2D user does not know the power allocation strategies in other D2D pairs. The D2D user can only obtain its own local information such as environment state and its own power allocation scheme. Therefore, we propose a multi-agent reinforcement learning approach to find an optimal power allocation scheme for each D2D user.

IV. REINFORCEMENT LEARNING FOR ENERGY EFFICIENT POWER ALLOCATION GAME IN D2D COMMUNICATION

In this section, we discuss the background of single-agent reinforcement learning and multi-agent reinforcement learning.

A. SINGLE-AGENT Q-LEARNING

We assume there is an agent whose task is to find an optimal policy through interactions with an environment. This problem can be formulated as a Markov Decision Process (MDP) defined as a 4-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}\rangle$, where $\mathcal{S} = \{s_1, s_2, \cdots, s_m\}$ is the finite set of state, $\mathcal{A} = \{a_1, a_2, \cdots, a_l\}$ is a set of agent discrete actions. The reward function $r = \mathcal{R}(s, a, s')$ is defined at state $s \in \mathcal{S}$, action $a \in \mathcal{A}$, and next state s'. The transition function $\mathcal{P}_{ss'(a)} = p(s'|s, a)$ defines the probability of next states s' if the agent is at state s and takes action a.

In reinforcement learning, the goal is to find the optimal policy $\pi^*(s)$ for each s, which maximizes the total expected discounted reward. Under policy π, the value of a state is defined as

$$V^{\pi}(s) = \mathbb{E}\left\{\sum_{k=0}^{\infty} \gamma^k r_i^{k+1} | s^0 = s\right\}, \tag{9}$$

where $0 \leq \gamma \leq 1$ is the discount factor, and $\mathbb{E}\{\cdot\}$ denotes the expectation operation, which is related to the stochastic property of the policy π and dynamics $\mathcal{P}_{ss'}(a)$. We can rewrite (9) to the Bellman equation as

$$V^{\pi}(s) = \mathbb{E}\{r_t | s_t = s\} + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}(a) V^{\pi}(s') \tag{10}$$

The optimal policy π^*, which is a mapping from the states to the optimal actions, would maximize the expected cumulative reward. There is at least one optimal strategy π^* that satisfies the following Bellman equation [31]

$$V^*(s) = V^{\pi^*} = \max_{a\in\mathcal{A}}\{\mathbb{E}(r(s, a)) + \gamma \sum_{s'\in\mathcal{S}} P_{ss'} V^*(s')\}. \tag{11}$$

The action-value function is the expected reward starting from state s, taking action a under the policy π

$$Q^{\pi}(s, a) = \mathbb{E}(r(s, a)) + \gamma \sum_{s'\in\mathcal{S}} P_{ss'} V(s') \tag{12}$$

The optimal policy $Q^*(s, a)$ is defined as

$$Q^*(s, a) = Q^{\pi^*} \tag{13}$$

Then we get

$$V^*(s) = \max_{a\in\mathcal{A}} Q^*(s, a) \tag{14}$$

Q-learning [2] is one of RL algorithms that tries to learn π^* without knowing the environment dynamics $\mathcal{P}_{ss'}(a)$. The agent learns by adjusting Q value according to the update rule,

$$Q^t(s, a) = (1 - \alpha) Q^t(s, a) + \alpha[r^t + \gamma \max_{a'\in\mathcal{A}} Q^t(s', a')], \tag{15}$$

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

B. MULTI-AGENT Q-LEARNING APPROACH

Our goal is to find a strategy to maximize energy-efficiency of a D2D communication system when the whole network consists of the large number of D2D users. Hence, we define D2D-Tx in each D2D pair as one RL agent as introduced in IV-A, which just only knows the information of their own environment. The network can be considered as a multi-agent system in which the optimal scheme consists of the optimal policies of individual agents. We now describe multi-agent Q-learning algorithm that is expected to solve the complex power control issue in this setting.

Multi-agent Q-learning has N agents corresponding to N D2D pairs, each agent is equipped with a classical Q-learning algorithm and learns without cooperating with the other agents. The ith agent is represented as a tuple $\langle \mathcal{S}_i, \mathcal{A}_i, \mathcal{R}_i, \mathcal{P}_i\rangle$, where $\mathcal{S}_i$ is a finite set of environment states, $\mathcal{A}_i$ is a finite set of agent actions, $\mathcal{R}_i$ is the reward function, and $\mathcal{P}_i$ is the state transition probability function of the ith agent. We define the agents, states, actions and reward function as follows:

Agent: Each agent is one D2D transmitter.
The system would consist of N such agents.

State: The state of ith D2D transmitter at time t is defined as

$$\mathcal{S}_i^t = (i, \mathcal{I}_i), \tag{16}$$

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

$$\mathcal{I}_i = \begin{cases} 1 & \text{for} \quad \gamma_i \geq \gamma^* \\ 0 & \text{for} \quad \text{otherwise} \end{cases} \tag{17}$$

Action: The action of each agent consists of a set of transmitting power levels. It is denoted as

$$\mathcal{A} = (a_1, a_2, \ldots, a_l), \tag{18}$$

where l represents that every agent has l power levels. There are many ways to choose actions based on the current action-value estimation, for example ϵ-greedy strategy, which is

described as follows:

$$\pi_a^s = \begin{cases} \arg\max_{a\in\mathcal{A}} Q(s,a) & \text{with probability} \quad 1-\epsilon \\ \text{random action} & \text{with probability} \quad \epsilon \end{cases} \tag{19}$$

Reward: The reward $\mathcal{R}_i$ of ith D2D user in state s_i is the immediate return due to the execution of action a_i

$$\mathcal{R}_i = \begin{cases} \dfrac{W\ln(1+\gamma_i)}{p_i} & \text{if} \quad \mathcal{I}_i = 1 \\ 0 & \text{if} \quad \mathcal{I}_i = 0 \end{cases} \tag{20}$$

The multi-agent Q-learning algorithm finds optimal Q-value $Q^*(s_i, a_i)$ in a recursive way after receiving a transition information $\langle s_i, a_i, s_i', \pi_i\rangle$, where $s_i^t = s_i \in \mathcal{S}_i$ and $s_i^{t+1} = s_i' \in \mathcal{S}_i$ are the environment states observed by agent i at time slot t and $t+1$, respectively; $a_i^t = a_i \in \mathcal{A}_i$ and $a_i^{t+1} = a_i' \in \mathcal{A}$ and π_i are the ith agent' action at time slot t and the transmission strategy during time slot t. The update rule for Q-learning is given by $Q_i^{t+1}(s_i, a_i)$:

$$Q_i^t(s_i,a_i) + \alpha^t\{r_i^t + \gamma \max_{a_i'\in\mathcal{A}} Q_i^t(s_i', a_i') - Q_i^t(s_i,a_i)\}, \tag{21}$$

where α is a learning rate and γ is a discount factor. The pseudo code of the multi-agent power allocation algorithm is depicted in Algorithm 1. At each iteration, each agent acts and updates its own policy independently.

Algorithm 1 Multi-Agent Q-Learning for Power Allocation Optimization

Initialisation:
for all $i \in N$ **do**
  Initialise the value function $Q_i(s, a, s') = 0$
  Initialise the initial state s_i according to (17)
end for
while not convergence **do**
  for all D2D i, $i \in N$ **do**
    Select action a_i^t according to the strategy π_i (19)
    Measure SINR at the receiver according to (3)
    Update the reward r_i^t according to (20)
    Observe the new state s_i^{t+1}
    Update the action-value $Q_i(s_i, a_i)$ according to (21)
    Update the state $s_i^t = s_i^{t+1}$
  end for
end while

V. DEEP REINFORCEMENT LEARNING FOR POWER ALLOCATION OPTIMISATION IN D2D COMMUNICATION

The approaches based on standard Q-learning as described in IV-A and IV-B can work well when the problem has a small set of states and actions [2]. However, when the problem size increases, they yield many limitations. Firstly, the convergence rate might become slow when the problem is large and the number of agents N is sufficiently large. Hence it cannot be adapted to real-time scenarios. Secondly, the storage of the lookup table $Q_i(s, a, s')$ (for every agent i, states s, s', and action a) becomes impractical and has limited generalization ability. These issues can be solved using deep reinforcement learning [32] where both the use of function approximation (i.e. using deep neural networks to approximate Q_i) and deep learning (i.e. training via stochastic gradient descent) play an important role to scale Q-learning up to much larger domains.

Algorithm 2 Multi-Agent Deep Q-Learning for Power Allocation Optimization With Experience Replay

Initialisation:
for all $i \in N$ **do**
  Randomly initialise the Q-network $Q_i(s, a; w)$
  Randomly initialise the target Q-network $\hat{Q}_i(s, a; \hat{w})$
  Initialise the replay memory D with capacity C
end for
while not convergence **do**
  for all D2D i, $i \in N$ **do**
    for Iteration **do**
      Select action a_i^t using ϵ-greedy via $Q_i(s_i^t, a_i^t; w)$
      Measure SINR at the receiver according to (3)
      Update the reward r_i^t according to (20)
      Observe the new state s_i^{t+1}
      Store transition $(s_i^t, a_i^t, r_i^t, s_i')$ in D
      if the replay memory D is full **then**
        Sample a mini-batch of K transitions from D
        Using stochastic gradient to minimize the loss:

$$\left[r_i + \gamma \max_{a'\in\mathcal{A}} \hat{Q}_i(s_i', a_i'; \hat{w}) - Q(s_i, a_i; w)\right]^2 \tag{22}$$

      end if
      Update the state $s_i^t = s_i^{t+1}$
      Update target network: $\hat{w} = \tau\hat{w} + (1-\tau)w$
    end for
  end for
end while

In addition, the performance of RL algorithms might become unstable or even diverge when a non-linear function approximation is used. This is due to the fact that a small change of Q-value may result in a big change to the policy. To address this issue, the authors in [32] also propose a deep Q-network (DQN) which is basically a DRL method with two major techniques: experience replay mechanism and target Q-network,

- Experience replay mechanism: The algorithm store transitions $\langle s_t, a_t, r_t, s_t'\rangle$ in a replay memory (a buffer memory). At each learning step, a mini-batch is sampled randomly from the memory pool and then a stochastic gradient descent is used to update the weights of the network representing Q. By doing so, the previous experiences are exploited more efficiently as the algorithm can re-use them. It ensures the fairness and robustness of DRL. Additionally, by using the experience replay, the data is more likely independent and identically distributed.

- Target Q-network: In the training process, the Q-value functions are updated continuously. Thus, the value estimations can become unstable. To address this issue, DQN uses two networks: a target network $\hat{Q}$ and the main network Q. The main network is still updated and used as previously mentioned. $\hat{Q}$ is used to store an old and stable main Q network which helps slow down the changes. This technique can guarantee performance improvement gradually.

DQN inherits and promotes advantages of both reinforcement learning and deep learning technique, and thus it has a wide range of applications in practice from resource allocation to interference management and security. We now describe the main DQN algorithm and its two variants: Double Deep Q-Learning (DDQN) and Dueling Deep Q-Learning.

A. DEEP Q-LEARNING

We assume the experience replay stores the last D tuples of $\langle s, a, r, s' \rangle$ in a replay buffer. We take mini-batches of random samples from this buffer to train Q-networks to ensure the fairness and robustness. In DQN, we use two networks to storing the Q-value. The first one Q_w with weights w is constantly updated while the second one, the target Q-network $\hat{Q}$ with weights $\hat{w}$, is synchronized from the first network once a while.

To estimate the network, we optimize the following sequence of loss function:

$$L_i = \mathbb{E}\left[(y_i^{\text{DQN}} - Q(s_i, a_i; w)^2\right], \tag{23}$$

with

$$y_i^{\text{DQN}} = r_i + \gamma \max_{a' \in \mathcal{A}} \hat{Q}(s_i', a'; \hat{w}) \tag{24}$$

During the training stage of the Q-network, instead of using only current experience, the network parameters are updated through training by the random samples mini-batch $\tilde{D}$ from the memory pool D using stochastic gradient descent. In particular, the gradient updating w is computed as

$$\nabla_w L_i \approx \frac{1}{|\tilde{D}|} \sum_{\{s_i, a_i, r_i, s_i'\} \in \tilde{D}} 2(y_i^{\text{DQN}} - Q(s_i, a_i; w)) \nabla_w Q(s_i, a_i; w). \tag{25}$$

Experience replay increases data efficiency through experience reuse in multiple updates. Moreover, it can reduce variance through uniform sampling from the replay buffer, because this can sampling reduce the correlation among the samples. We propose a multi-agent Deep Q-Learning algorithm for power allocation in D2D communication as described in Algorithm 2 where τ is the network update parameter.

B. DOUBLE DEEP Q-LEARNING

Deep Q-learning is known to learn unrealistically high action values because it includes a maximization step over estimated action values, which tends to prefer overestimated to underestimated values. The core idea of the double Q-learning algorithm is to reduce overestimation by decomposing the max operation in the target into action selection and action evaluation [32].

In Q-learning and Deep Q-learning, the max operation uses the same values to both select and evaluate an action. This can, therefore, lead to overoptimistic value estimates. To mitigate this problem, deep double Q-learning (DDQL) uses the following target

$$y_i^{\text{DDQN}} = r_i + \gamma Q(s_i', \arg\max_{a' \in \mathcal{A}} \hat{Q}(s_i', a_i')). \tag{26}$$

The details of our proposed multi-agent DDQL for power allocation in D2D communication is provided in Algorithm 3.

Algorithm 3 Multi-Agent Double Deep Q-Learning for Power Allocation Optimization With Experience Replay

```
Initialisation:
for all i ∈ N do
    Randomly initialise the Q-network Q(s, a; w)
    Randomly initialise the target Q-network Q̂(s, a; ŵ)
    Initialise the replay memory D with capacity C
end for
while Not convergence do
    for all D2D i, i ∈ N do
        for Iteration do
            Select action a_i^t using ε-greedy via Q_i(s_i^t, a_i^t; w)
            Measure SINR at the receiver according to (3)
            Update the reward r_i^t according to (20)
            Observe the new state s_i^{t+1}
            Store transition (s_i^t, a_i^t, r_i^t, s_i') in D
            if the replay memory D is full then
                Sample a mini-batch of K transitions from D
                Using stochastic gradient to minimize the loss:
                    [r_i + γ Q_i(s', arg max_{a'∈A} Q̂_i(s_i', a_i')) − Q(s_i, a_i)]^2
            end if
            Update the state s_i^t = s_i^{t+1}
            Update target network: ŵ = τŵ + (1 − τ)w
        end for
    end for
end while
```

C. DUELING DEEP Q-LEARNING

The main idea of Dueling Deep Q-learning is to reduce the variance in Q-learning, DQN and DDQL. Its main idea originates from a simple technique in statistics in which an estimate bias term can be used to reduce the variance of the Q-value function estimation.

As a result, the deep dueling Q-learning network model [33] consists of two sequences (or stream) of fully connected layers. Both streams are constructed in such a

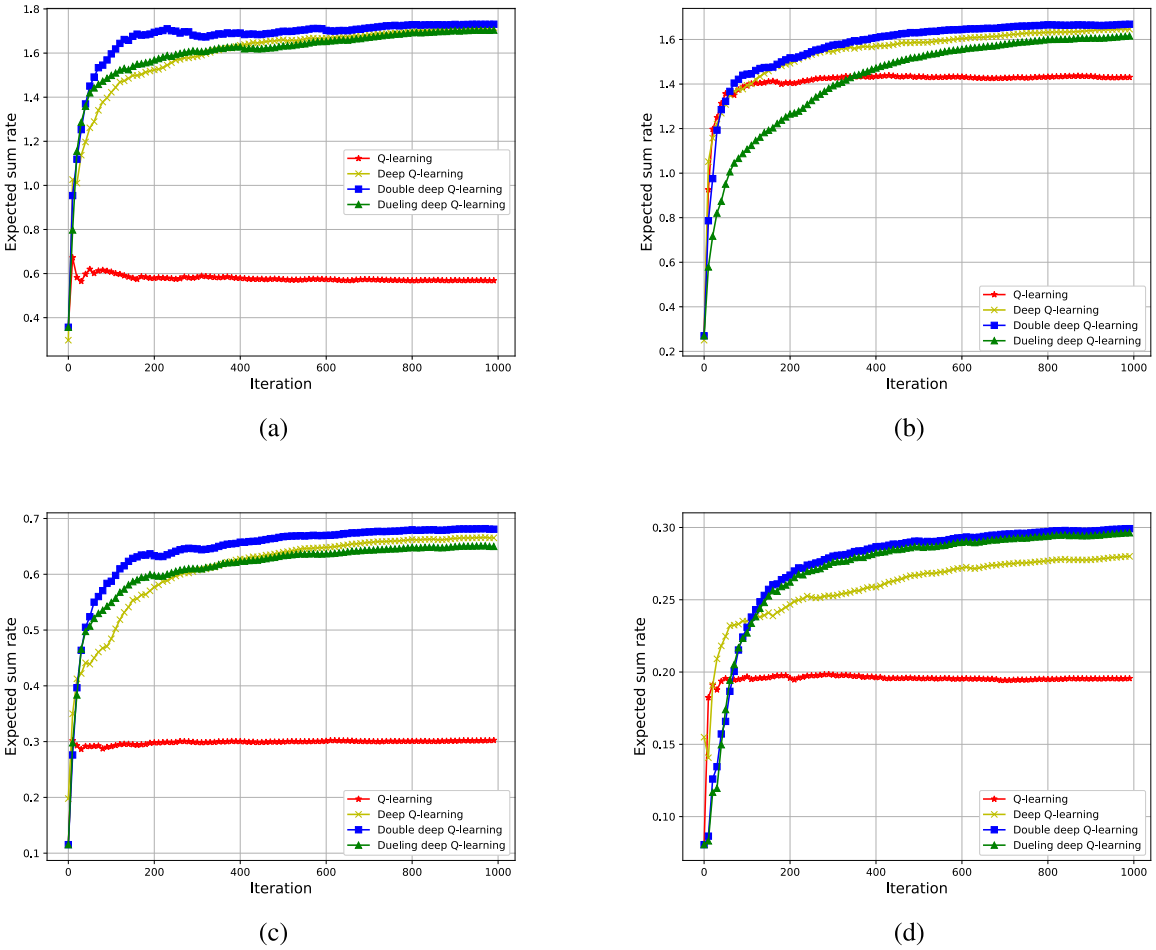


FIGURE 1. Expected sum rate of the network with different number of D2D pairs: (a) $N = 10$, (b) $N = 7$, (c) $N = 5$, (d) $N = 2$, with the action space of size $|\mathcal{A}| = 23$ and ϵ-greedy $\epsilon = 0.9$.

way that they have a capability of providing separate estimates of the value $V(s)$ and advantage $A(s, a)$ functions. Finally, the two streams are combined to produce a single output Q-function $Q(s, a) = A(s, a) + V(s)$. Since the output of the dueling network is a Q-function, it can be trained with many existing algorithms such as DDQL, DQN, and SARSA

The rationale use of the advantage function $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ is that this allows the agent to determine not just how good its actions are, but also how much better they turn out to be than expected. Intuitively, this allows the agent focus on where the network prediction is lacking. For the state-value function $V^\pi(s) = \mathbb{E}_{a\sim\pi(s)}[Q^\pi(s, a)]$, it follows that $\mathbb{E}_{a\sim\pi(s)[A^\pi(s,a)]} = 0$. In addition, given a deterministic policy, $a^* = \arg\max_{a'\in\mathcal{A}} Q(s, a')$, we have $Q(s, a^*) = V(s)$ and hence $A(s, a^*) = 0$.

The function $Q(s, a)$ is only a parameterized estimate of the true Q-function as output. Therefore, there is a lack of identifiability because if given Q we cannot recover V and A uniquely, which leads to a poor performance. To address this issue, the authors in [33] propose an objective that forces the advantage function estimator to have zero advantage at the chosen action by letting the last module of the network implementing the forward mapping

$$Q(s, a) = V(s) + \left(A(s, a) - \max_{a'\in|\mathcal{A}|} A(s, a')\right). \quad (28)$$

For $a^* = \arg\max_{a'\in\mathcal{A}} Q(s, a') = \arg\max_{a'\in\mathcal{A}} A(s, a')$, we obtain $Q(s, a^*) = V(s)$. Hence, the stream $V(s)$ provides an estimate of the value function, while the other stream produces an estimate of the advantage function.

To simplify the computation of gradients, equation (28) can be transformed by replacing the max operator with an average as,

$$Q(s, a) = V(s) + \left(A(s, a) - \frac{1}{|\mathcal{A}|}\sum A(s, a)\right). \quad (29)$$

The rank of A is not be changed by subtracting the mean in equation (29), so it preserves any greedy or ϵ-greedy policy based on Q values and identifiability. It is important to note that equation (29) is viewed and implemented as part of the network but not as a separate algorithmic step. Moreover, the estimates $V(s)$ and $A(s, a)$ are computed automatically without any extra supervision or algorithmic modifications.

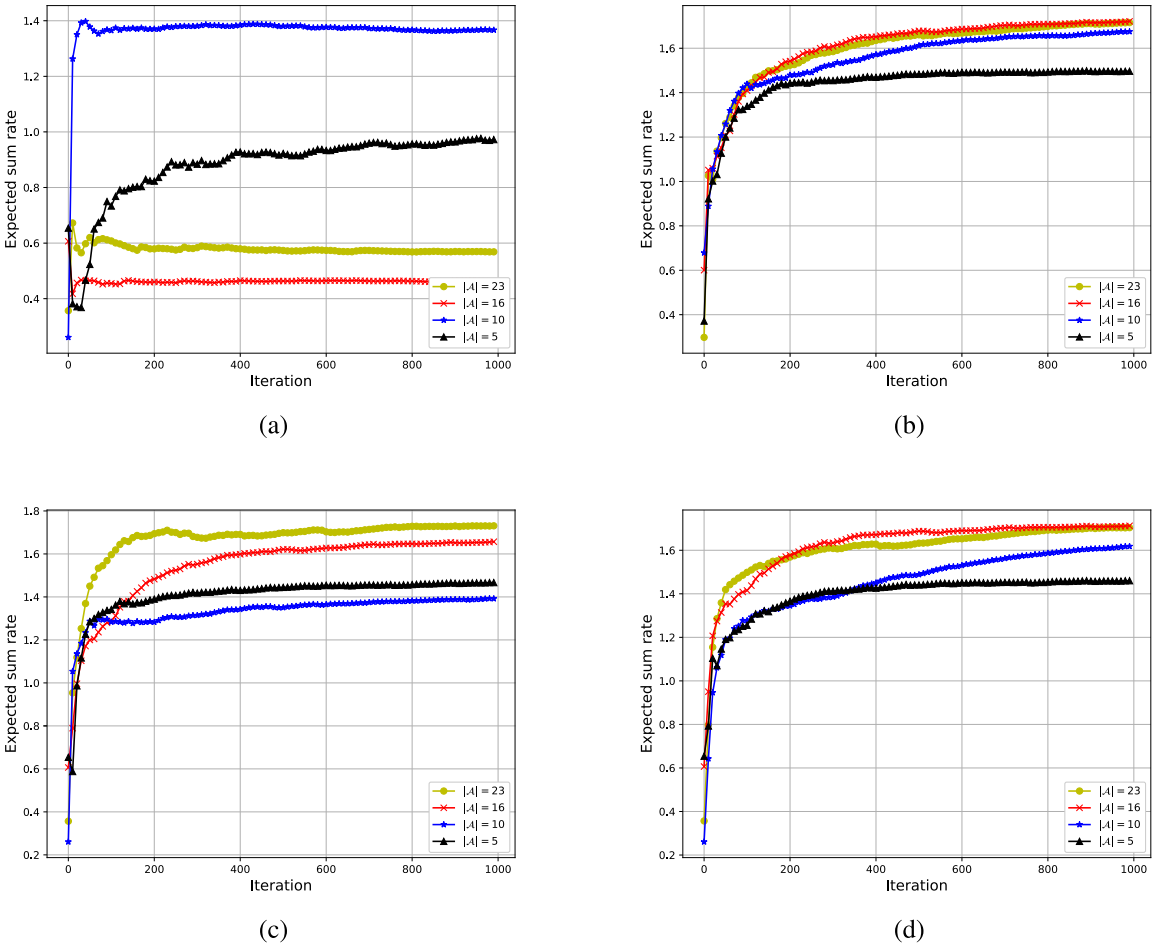


FIGURE 2. Performance results with the size of the action space $|\mathcal{A}| = 23$, $|\mathcal{A}| = 16$, $|\mathcal{A}| = 10$, $|\mathcal{A}| = 5$. (a) Q-learning. (b) Deep Q-learning. (c) Double deep Q-learning. (d) Dueling deep Q-learning.

The details of the power allocation based on dueling deep Q-learning algorithm is presented in Algorithm 4.

VI. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed methods on PC Intel(R) Core(TM) i7-8700 CPU @ 3.20Ghz. All algorithms are implemented using Tensorflow 1.13.1 [34]. The action-value function and its target networks are initialized with 1 hidden layer of 20 units. For initial values of weights and biases, we set to small random values according to a zero-mean Gaussian distribution with a standard deviation of 0.1. We use Adam optimizer [35] for training. The other simulation parameters are provided in Table 1.

TABLE 1. Simulation parameters.

Parameters	Value
Bandwidth (W)	1 MHz
Path-loss exponent	$\alpha_h = 3$
p_{max}	23 dBm
p_{min}	0 dBm
Channel power gain at the reference	$\beta_0 = -30dB$
Noise power density	$\eta = 0.5$
SINR QoS constraint	$\gamma^* = 0$ dB
Learning rate	$\alpha = 0.01$
Discount factor	$\gamma = 0.9$

A. PERFORMANCE COMPARISON ANALYSIS

Fig. 1 plots the performance results of four multi-agent deep reinforcement learning approaches with a different number of D2D pairs in coverage, $N = 2, 5, 7, 10$. The double deep Q-learning constantly performs better than other approaches in terms of the expected sum rate. The multi-agent classical Q-learning is less favorable due to its worst performance and high variance. At convergence, it seems that the performances of multi-agent double deep Q-learning and deep dueling deep Q-learning are comparable. These results suggest that using deep reinforcement learning and multi-agent learning significantly helps to find an optimal policy for non-cooperative energy-efficient power allocation in D2D communication.

B. SCALABILITY ANALYSIS

We introduce another experiment in order to analyze how deep reinforcement learning-based methods are able to scale

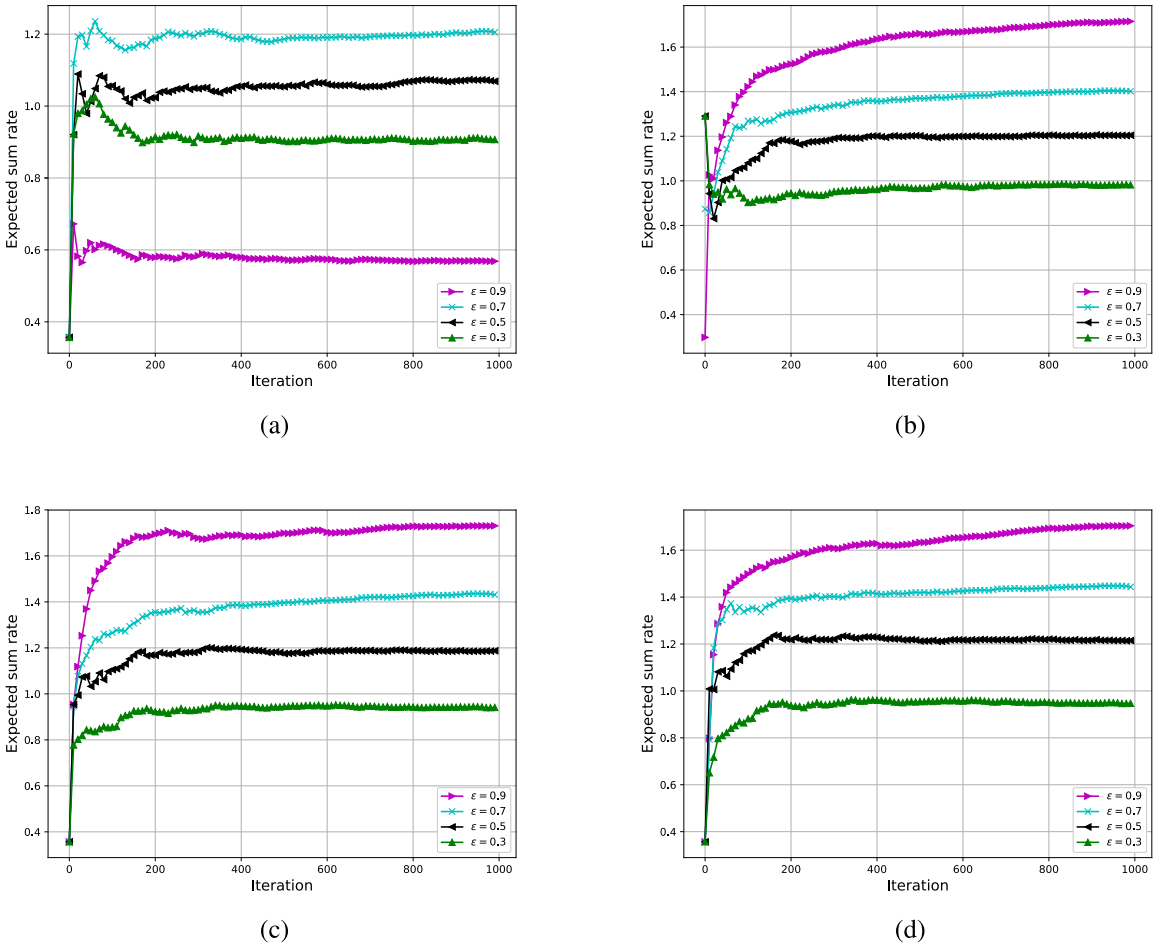


FIGURE 3. Performance results with different exploration strategies ϵ-greedy. (a) Q-learning. (b) Deep Q-learning. (c) Double deep Q-learning. (d) Dueling deep Q-learning, on action space $|\mathcal{A}| = 23$ and number of D2D pairs $N = 10$.

to large domains. We fix the number of pairs $N = 10$ but vary the size of the action space. It is well known that the problem complexity increases exponentially with the size of the action space. Fig. 2 compares the efficiency of the classical Q-learning and deep reinforcement learning approaches using a different size for the action space. The results show that the standard Q-learning approach is only suitable with a problem of small state and action space. On more complicated problems, the performance of Q-learning becomes very unstable. Meanwhile, the deep Q-learning, DDQL, and dueling DQL can still perform well and stable with a larger size of action space $\mathcal{A}$.

C. EXPLORATION/EXPLOITATION ANALYSIS

The balance between exploration and exploitation of deep reinforcement learning is a great challenge for use in practice. In this experiment, we measure the performance of deep reinforcement learning algorithms with different values of exploration rate ϵ. Fig. 3 shows that the comparison results for four proposed algorithms. The results suggest that all three deep reinforcement learning approaches are less sensitive to the choice of ϵ. That means when this parameter is set to 0.9, they perform quite stable.

TABLE 2. The running time of deep reinforcement learning in different environment scenarios.

	N = 2	N = 5	N = 7	N = 10
$\|\mathcal{A}\| = 5$	$0.39ms$	$1ms$	$1.38ms$	$1.95ms$
$\|\mathcal{A}\| = 10$	$0.41ms$	$1.02ms$	$1.40ms$	$1.98ms$
$\|\mathcal{A}\| = 16$	$0.42ms$	$1.02ms$	$1.41ms$	$2.02ms$
$\|\mathcal{A}\| = 23$	$0.42ms$	$1.02ms$	$1.42ms$	$2.02ms$

D. RUNNING TIME ANALYSIS

This experiment evaluates the running time of deep reinforcement learning during test time after training. Table 2 shows the average running time of deep reinforcement learning methods. Each D2D transmitter only needs a few milliseconds to select actions that have already been optimized offline. The selected actions are expected to maximize the performance of the network while satisfying the QoS constraints. The results suggest that our proposed methods are applicable for real-time optimization and practical use.

Algorithm 4 Multi-Agent Dueling Deep Q-Learning for Power Allocation Optimization With Experience Replay

Initialisation:
for all $i \in N$ **do**
  Randomly initialise the Q-network $Q(s, a; w)$
  Randomly initialise the target Q-network $\hat{Q}(s, a, \hat{w})$
  Initialise the replay memory D with capacity C
end for
while Not convergence **do**
  for all D2D i, $i \in N$ **do**
    for Iteration **do**
      Select the action a_i^t using ϵ-greedy via $Q_i(s_i^t, a_i^t; w)$
      Measure SINR at the receiver according to (3)
      Update the reward r_i^t according to (20)
      Observe the new state s_i^{t+1}
      Store transition $(s_i^t, a_i^t, r_i^t, s_i')$ in D
      if The replay memory D is full **then**
        Sample an uniform random mini-batch of K transitions from D
        Combine the value function and advantage functions as in equation (29)
        Update parameters of neural networks from learning batches using stochastic gradient descent to minimize the loss:

$$[r_i + \gamma Q(s', \max_{a' \in \mathcal{A}} \hat{Q}_i(s_i', a_i')) - Q(s_i, a_i)]^2 \quad (27)$$

      end if
      Update the state $s_i^t = s_i^{t+1}$
    end for
  end for
end while

VII. CONCLUSION

This paper presented non-cooperative deep reinforcement learning algorithms to optimize energy-efficient power allocation in D2D communication. We propose three different multi-agent deep reinforcement learning algorithms. We show that deep reinforcement learning is used to maximize the performance of the entire network while guaranteeing the QoS constraints in real-time scenarios. The results are very promising in terms of scalability to large domains, high performance, and low running time. For future research directions, we plan to investigate other deep reinforcement learning algorithms that are able to deal with continuous action problems. A further improved multi-agent learning strategy for non-cooperative problems is also a highly potential research direction.

REFERENCES

[1] 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.

[2] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 135. Cambridge, MA, USA: MIT Press, 1998.

[3] M. J. Matarić, "Reinforcement learning in the multi-robot domain," *Auton. Robots*, vol. 4, no. 1, pp. 73–83, 1997.

[4] L. Steels, "Language as a complex adaptive system," in *Proc. Int. Conf. Parallel Prob. Solv. Nature*. Springer, 2000, pp. 17–26.

[5] I. Ghory, "Reinforcement learning in board games," Dept. Comput. Sci., Univ. Bristol, Bristol, U.K., Tech. Rep. CSTR-04-004, May 2004.

[6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[8] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, vol. 1, no. 3, p. 6.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.

[11] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Dec. 2015, pp. 142–150.

[12] K. K. Nguyen, D. T. Hoang, D. Niyato, P. Wang, D. Nguyen, and E. Dutkiewicz, "Cyberattack detection in mobile cloud computing: A deep learning approach," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2018, pp. 1–6.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: https://arxiv.org/abs/1312.5602

[14] L. Zhang, M. Xiao, G. Wu, and S. Li, "Efficient scheduling and power allocation for D2D-assisted wireless caching networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2438–2452, Jun. 2016.

[15] R. Yin, C. Zhong, G. Yu, Z. Zhang, K. K. Wong, and X. Chen, "Joint spectrum and power allocation for D2D communications underlaying cellular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2182–2195, Apr. 2016.

[16] Y. Jiang, Q. Liu, F. Zheng, X. Gao, and X. You, "Energy-efficient joint resource allocation and power control for D2D communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6119–6127, Aug. 2016.

[17] 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.

[18] 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.

[19] 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.

[20] R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan, and D. Matolak, "A machine learning approach for power allocation in HetNets considering QoS," in *Proc. IEEE ICC*, May 2018, pp. 1–7.

[21] A. Masadeh, Z. Wang, and A. E. Kamal, "Reinforcement learning exploration algorithms for energy harvesting communications systems," in *Proc. ICC*, May 2018, pp. 1–6.

[22] 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.

[23] 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.

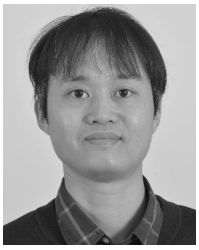
[24] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.

[25] 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.

[26] 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," 2019, *arXiv:1904.09673*. [Online]. Available: https://arxiv.org/abs/1904.09673
[27] 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.
[28] 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.
[29] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning-based mode selection and resource management for green fog radio access networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1960–1971, Apr. 2019.
[30] H. Ye and G. Y. Li, "Deep reinforcement learning for resource allocation in V2V communications," in *Proc. IEEE ICC*, May 2018, pp. 1–6.
[31] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA, USA: Athena Scientific, 1995, no. 2.
[32] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.
[33] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2015, *arXiv:1511.06581*. [Online]. Available: https://arxiv.org/abs/1511.06581
[34] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX conf. OSDI*, Nov. 2016, pp. 265–283.
[35] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 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 massive Internet of Things (IoTs).

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. He is currently with Queen's University Belfast, U.K., where he was a Lecturer (Assistant Professor), from 2013 to 2017 and has been a Reader (Associate Professor), since 2018. He is the author or coauthor of over 340 technical papers published in scientific journals (over 200 articles) and presented at international conferences (over 140 papers). His current research interests include the Internet of Things (IoT), wireless communications, molecular communications, and signal processing. He received the Best Paper Award at the IEEE Vehicular Technology Conference (VTC-Spring) 2013, the IEEE International Conference on Communications (ICC) 2014, the IEEE Global Communications Conference (GLOBECOM) 2016, the IEEE Digital Signal Processing Conference (DSP) 2017, and the International Wireless Communications & Mobile Computing Conference (IWCMC) 2019. He is a recipient of the prestigious Royal Academy of Engineering Research Fellowship, from 2016 to 2021, and received a prestigious Newton Prize 2017. He currently 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, also with the Ravensburg-Weingarten University of Applied Sciences, Germany, from 2011 to 2013, and also with the Machine Learning and Robotics Laboratory, University of Stuttgart, from 2013 to 2017. He has been a Lecturer (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 Institute National Polytechnique 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 the 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 papers in international peer-reviewed journals and conferences in related disciplines. He received the prestigious UCD School of Computer Science Outstanding Teaching Award, in 2018. He is a Principal Investigator of an EU DG-Home Research Grant. He is also a Science Foundation of Ireland (SFI) Co-Principal Investigator and a Funded Investigator.

MINH-NGHIA NGUYEN was born in Ho Chi Minh City, Vietnam. He received the B.Eng. degree in software and electronic systems engineering from Queen's University Belfast, in 2018. He is currently a Research Assistant with Duy Tan University, Da Nang, Vietnam. His research interests include physical-layer security, wireless communications, and artificial intelligence. He was a recipient of the Best Paper Award at the 2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing, in 2018, and the IET Prize 2018.