IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 7, 2019, accepted December 19, 2019, date of publication December 23, 2019, date of current version January 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961914

INVITED PAPER

Smart Power Control for Quality-Driven Multi-User Video Transmissions: A Deep Reinforcement Learning Approach

TICAO ZHANG AND SHIWEN MAO, (Fellow, IEEE)
Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA
Corresponding author: Shiwen Mao (smao@ieee.org)

This work was supported in part by the NSF under Grant IIP-1822055 and Grant ECCS-1923717, and in part by the Wireless Engineering Research and Education Center (WEREC), Auburn University, Auburn, AL, USA.

ABSTRACT Device-to-device (D2D) communications have been regarded as a promising technology to meet the dramatically increasing video data demand in the 5G network. In this paper, we consider the power control problem in a multi-user video transmission system. Due to the non-convex nature of the optimization problem, it is challenging to obtain an optimal strategy. In addition, many existing solutions require instantaneous channel state information (CSI) for each link, which is hard to obtain in resource-limited wireless networks. We developed a multi-agent deep reinforcement learning-based power control method, where each agent adaptively controls its transmit power based on the observed local states. The proposed method aims to maximize the average quality of received videos of all users while satisfying the quality requirement of each user. After off-line training, the method can be distributedly implemented such that all the users can achieve their target state from any initial state. Compared with conventional optimization based approach, the proposed method is model-free, does not require CSI, and is scalable to large networks.

INDEX TERMS Multi-user video transmission, multi-agent deep reinforcement learning, power control, quality of experience.

I. INTRODUCTION

Due to the popularization of wireless multimedia communication services and applications, such as mobile TV, 3D video, 360-degree video, multi-view video, and augmented reality (AR), there is an explosive growth of mobile data traffic. It is expected that the mobile traffic will increase seven-fold from 2017 to 2022 [1]. Moreover, 82% of the mobile data will be video related by 2022 [1]. The dramatically increasing video data demand brings great challenges to the present and future wireless networks [2].

Device-to-device (D2D) communications have been regarded as an emerging 5G communication technology to meet the increasing data demand [3]–[5]. In D2D communications, nearby devices can establish local links so that traffic flows directly between them instead of through a base station (BS). As a result, the system spectrum efficiency and the system coverage can be potentially improved. Also, delay can be significantly reduced. However, interference management is becoming a challenging problem with the presence of D2D links [6]. Specifically, in a multi-user communication network, a transmitter may increase its transmit power to ensure a better video quality for the corresponding receiver, but at the same time, it may degrade the performance of the links it interferes with.

Transmit power control, as a physical layer issue, has been well studied since the first generation cellular networks [7]. Many centralized interference management methods have been developed. The weighted minimum mean square error (WMMSE) algorithm [8] and fractional programming (FP) algorithm [9] are typical centralized algorithms. These algorithm often require precise channel state information (CSI) for all the links, which will incur considerable signaling overhead. Moreover, the complexity of centralized algorithm increases with the number of users, bringing about heavy computational pressure on the power

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

controller. To reduce the signaling overhead and better adapt to large scale networks, a series of distributed algorithms have been developed. For example, in [10], the power allocation problem in cognitive wireless network was formulated as a noncooperative game. A stochastic power allocation with conjecture-based multi-agent Q-learning approach was proposed. The authors in [11] proposed a Stackelberg game based power control scheme for D2D communication underlay cellular networks. By introducing a new co-tier price factor, the distributed power control algorithm can mitigate the cross-tier interference effectively. Despite their good performance, current solutions often require frequent information exchange and cannot guarantee an optimal performance.

Meanwhile, we have observed that these physical layer technologies generally aim to optimize the transmission data rate or bit error rate (BER), they do not improve the user's quality of service (QoS) or quality of experience (QoE) directly, when users are watching a specific video. Given the same transmission bandwidth, different videos generally have different qualities. As an application layer performance metric, video quality directly reflect the user satisfaction level in contrast to physical layer metrics. In future mobile networks, it is more important to develop a cross-layer interference management approach that jointly considers the physical layer issues as well as the user's requirement and experience [12], [13]. Motivated by this observation, some cross-layer video transmission designs have been proposed. For example, the authors in [14] designed a quality-driven scalable video transmission framework in a non-orthogonal multiple access (NOMA) system and proposed a suboptimal power allocation algorithm. This algorithm leverages the hidden monotonic property of the problem and it has a polynomial time complexity. The recent work [15] proposed a spatial modulation (SM) and NOMA integrated system for multiuser video transmission. Efficient algorithms are proposed to perform optimal power control so that the user's QoE can be maximized. A novel cross-layer optimization framework is proposed in [16] for scalable video transmission over OFDMA networks. The proposed iterative algorithm can jointly maximize the achievable sum rate and minimize the distortion among multiple videos. In our recent work [17], a cross-layer optimization framework for softcast video transmission is developed and analyzed. Compared with physical layer-only designs, such cross layer optimization for video transmissions help users enjoy a better perceived video quality. Despite the success of these algorithms, they require that every user to have full knowledge of the CSI for all the links, which may be infeasible in practice. Besides, the formulated problem is generally non-convex. The developed method often lead to a sub-optimal solution.

Recently, machine learning (ML) has achieved great success in a variety of fields, such as computer version and speech recognition. Deep reinforcement learning (DRL), as a powerful ML technique, has show high potential for many challenging tasks, such as human-level control [18] and computer games [19]. In DRL, the agent considers the long-term reward, rather than simply obtaining the instant maximum reward. This is quite important for resource optimization problems in wireless networks, where the channel state changes rapidly. There is now an increasing interest on incorporating DRL into the design of wireless networking algorithms [20], such as mobile off-loading [21], dynamic channel access [22], [23], mobile edge computing and caching [24], [25], dynamic base station on and off [26], TCP congestion control [27], and resource allocation [28]–[31].

In particular, the authors in [28] consider the problem of power control in a cognitive radio system consisting of a primary user and a secondary user. With DRL, the secondary user can interact with the primary user efficiently to reach a target state after a few number of steps. Another work in [29] demonstrates the potential of DRL for power control in wireless networks. Instead of searching for the near-optimal solution by solving the challenging optimization problem, the authors develop a distributed dynamic power control scheme. This method is model-free and the system's weighted sum rate can be maximized. The authors in [30] investigated the spectrum sharing problem in vehicular networks with a DRL based solution. The multiple vehicle-to-vehicle (V2V) agents dynamically allocate their power and spectrum in a cooperate way so that their sum capacity can be maximized.

In this paper, we consider the power allocation and interference management problem in a multi-user video transmission system from the point of a cross-layer optimization. To the best of our knowledge, this is the first work that attempts to integrate DRL for interference management to improve users' video viewing quality. The main contributions of this paper are summarized as follows.

- The proposed algorithm is based on multi-agent deep Q-learning, which amenable to distributed implementation. It is model-free and does not require labeled training data. It can be applied to arbitrary network configurations.
- Each agent does not need to know other agents' CSI. The complexity of the proposed algorithm does not increase with the network size. This method can be applied to very large networks.
- This work is a cross layer design which considers both the physical layer issues as well as the application layer video-related design factors. By properly designing the reward function, users can actually work in a cooperative manner to achieve a high level of satisfaction.

The remainder of this paper is organized as follows. The system model and the problem formulation are discussed in Section II. In Section III, we develop a multi-agent DRL algorithm for power control. Simulation setup is provided in Section IV. Experimental results are given in Section V, followed by conclusions in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. PHYSICAL LAYER MODEL

We consider a wireless network consisting of N users, where all the users share a common spectrum resource. As shown

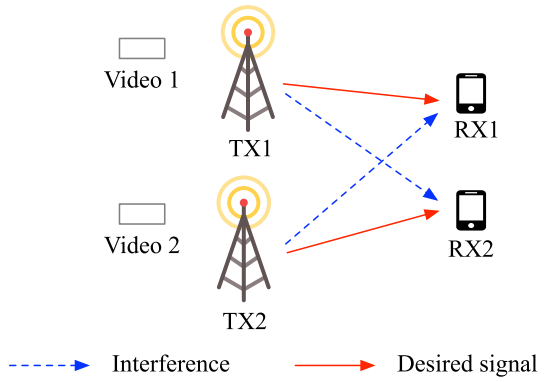


FIGURE 1. System model for a radio network with multiple video users ($N = 2$).

in Fig. 1, each user consists of a transmitter and receiver pair. Each receiver requests a specific video from its corresponding transmitter. We assume a cooperative system that different users can exchange information with each other, including channel gain, power control vectors, and some acknowledgment (ACK) signals. The information exchange process can be implemented to occur once per time slot, either in a wireless or a wired manner. Conventional technologies, such as Zigbee [32], can be used to convey this information to other users in a timely fashion. Note that Zigbee uses a different frequency, hence it generates no interference to the video users. The users dynamically adjust their transmit power based on the information collected from their neighbor users. Each user has a minimum QoE requirement for the received video. We aim to develop an optimal power control policy so that their combined QoE is maximized and all the users' minimum QoE requirements are satisfied.

Let p_i, $i = 1, 2, \ldots, N$, denote the transmit power of user i. Let h_{ij} be the channel gain from transmitter Tx_i to receiver Rx_j, $i, j \in \{1, 2, \ldots, N\}$. The signal-to-noise-plus-interference ratio (SINR) at receiver i can be computed as

$$SINR_i = \frac{|h_{ii}|p_i}{\sum_{j\neq i}|h_{ji}|p_j + \sigma_i^2}, \quad i, j \in \{1, 2, \ldots, N\}, \tag{1}$$

where σ_i^2 is the noise power at receiver i. We consider a free-space propagation model. So the channel gain is

$$h_{ij} = \left(\frac{\lambda}{4\pi d_{ij}}\right)^2, \tag{2}$$

where λ is the signal wavelength and d_{ij} is the distance between transmitter Tx_i and receiver Rx_j. We denote the distance matrix as $\mathbf{D} = [d_{ij}]$.

Since all the users share the same frequency spectrum for video transmissions, they have the same bandwidth B. The data transmission rate for user i can be expressed as

$$R_i(\mathbf{p}) = B\log_2(1 + SINR_i), \tag{3}$$

where $\mathbf{p} = [p_1, p_2, \ldots, p_N]$ is the transmit power allocation vector. It can be seen that the transmission rate of each user is determined by the transmit power allocation vector.

B. VIDEO TRANSMISSION MODEL

For video applications, PSNR is a common objective performance measure, which is highly correlated to user-perceived video quality. The relationship between PSNR value Q and distortion is given by

$$PSNR = Q = 10\log_{10}\left(\frac{255^2}{MSE}\right), \tag{4}$$

where the mean-squared-error (MSE) is used to characterize distortion.

In [33], the author propose a general semi-analytical rate-distortion (R-D) model, which has been verified for scalable video coding (SVC) in [34]. With this model, the relationship between the rate and distortion at the encoder side can be predicted. Specifically, the video coding rate for user i can be expressed as a function of PSNR Q_i as follows.

$$F_i(Q_i) = \frac{\theta_i}{255^2 10^{-Q_i/10} + \alpha_i} + \beta_i, \quad Q_i \geq Q_{i,min}, \tag{5}$$

where $Q_{i,min}$ is the minimum PSNR value corresponding to the minimum rate $F_{i,min}$. The parameters θ_i, α_i and β_i depends on the video content, encoder, and the RTP packet loss rate. These parameters can be obtained with a curve-fitting method over at least six empirical R-D samples [34], [35] and a relevant number of iterations, to achieve a high accuracy.

The authors in [36] further simplify this model to reduce complexity by eliminating the parameter α_i, i.e.,

$$F_i(Q_i) = \frac{\theta_i}{255^2 10^{-Q_i/10}} + \beta_i, \quad Q_i \geq Q_{i,min}. \tag{6}$$

In this case, only four R-D samples are sufficient to determined the model. In this paper, we adopt the simplified model, although the developed method also applies to any other R-D models.

Without loss of generality, we assume that the overhead introduced by the network stack layers (e.g., header and trailer bits) is constant, so we ignore this overhead for simplification. As a result, the physical layer rate (3) is assumed to be equal to the transmission rate at the application layer (6), i.e., $R_i(\mathbf{p}) = F_i(Q_i)$. The relationship between the PSNR of a received video and its corresponding transmit power can thus be expressed as

$$\begin{aligned} Q_i(\mathbf{p}) &= F_i^{-1}(R_i(\mathbf{p})) \\ &= -10\log_{10}\left(\frac{\theta_i}{R_i(\mathbf{p}) - \beta_i}\right) + 20\log_{10} 255. \end{aligned} \tag{7}$$

C. QUALITY-DRIVEN POWER ALLOCATION PROBLEM

The ultimate goal of power control is to improve the overall video quality of all users. We formulate this problem as follows.

$$\max_{\mathbf{p}} \quad Q(\mathbf{p}) = \frac{1}{N}\sum_{i=1}^{N} Q_i(\mathbf{p}) \tag{8}$$

$$\text{s.t.} \quad 0 \leq p_i \leq p_{max}, \quad \forall i \tag{9}$$

$$Q_i \geq Q_{i,min}, \quad \forall i, \tag{10}$$

where $Q_i(\mathbf{p})$ is given in (7). Note that (9) is the system power constraint and (10) is the video quality constraint, which depends on a variety of factors such as the video content, encoder setting, and the user's quality requirement.

Based on (3) and (6), it can be seen that PSNR is a monotone function in terms of SINR. The quality constraint (10) can thus be replaced by the corresponding SINR constraint. To simplify expression, we rewrite Problem (8) as follows.

$$\max_{\mathbf{p}} \quad \phi\left(\frac{f_1(\mathbf{p})}{g_1(\mathbf{p})}, \frac{f_2(\mathbf{p})}{g_2(\mathbf{p})}, \ldots, \frac{f_N(\mathbf{p})}{g_N(\mathbf{p})}\right) \tag{11}$$

$$\text{s.t.} \quad 0 \le p_i \le p_{max}, \ \forall i \tag{12}$$

$$SINR_i(\mathbf{p}) \ge SINR_{i,min}, \quad \forall i, \tag{13}$$

where $\phi(\mathbf{x})$ is an increasing function on $\mathbb{R}_+^N$, expressed as

$$\phi(\mathbf{x}) = -\frac{10}{N}\log_{10}\prod_{i=1}^{N}\left(\frac{\theta_i}{B\log_2(1+x_i)}\right) + \frac{20}{N}\log_{10}255, \tag{14}$$

and

$$f_i(\mathbf{p}) = |h_{ii}| \cdot p_i \tag{15}$$

$$g_i(\mathbf{p}) = \sum_{j\neq i}|h_{ji}| \cdot p_j + \sigma_i^2. \tag{16}$$

It can be seen that Problem (11) actually belongs to the class of generalized linear fractional programming (GLFP) problems. In addition, combining together with the structure of functions $f_i(x)$ and $g_i(x)$, this problem is actually non-convex [37]. Generally speaking, there is no efficient solutions to find the global optimal solution within polynomial time.

III. THE MULTI-AGENT DEEP REINFORCEMENT LEARNING APPROACH

In the proposed multi-user video transmission system, the transmitter of each user dynamically adjusts its transmit power based on the observed environment state. The action taken at the next time slot depends on the current observations, hence it can be modeled as a Markov Decision Process (MDP). We develop a multi-agent deep reinforcement learning approach to solve the problem.

A. OVERVIEW OF DEEP REINFORCEMENT LEARNING

Reinforcement learning (RL) is an effective technique to solve the MDP problems. In RL, agents learn an optimal policy through interactions with the environment, by receiving an intermediate reward together with a state update after taking each action. The received reward as well as the observed new state will help adjust the control policy. The process will continue until an optimal policy is found.

The most representative RL algorithm is Q-learning, where the policy is updated by an action-value function, referred to as the Q-function. Let $\mathcal{S}$ denote the set of possible states and $\mathcal{A}$ denote the set of discrete actions. The policy $\pi(s, a)$ is the probability of taking an action $a \in \mathcal{A}$ when given a state $s \in \mathcal{S}$. At time instant t, the agent takes action $a^t \in \mathcal{A}$ when observing a state $s^t \in \mathcal{S}$. Then the agent receives a reward r^t and the next state s^{t+1} is observed. The Q-learning algorithm aims to maximize a certain reward over time. For example, we can define the reward function as

$$R^t = \sum_{\tau=0}^{\infty}\gamma^\tau r^{t+\tau}, \tag{17}$$

where $\gamma \in (0, 1]$ is a discount scalar representing the tradeoff between the immediate and future rewards. $\gamma = 0$ means we only care about the immediate reward. A lager γ means earlier period rewards play a more important role.

Under a policy $\pi(s, a)$, the Q-function of the agent with action a and state s is defined as

$$Q_\pi(s, a) = \mathbb{E}_\pi[R^t | s^t = s, a^t = a]. \tag{18}$$

Q-learning aims to maximize the Q-function (18). The optimal action-value function, $Q^*(s, a) \triangleq \max_\pi Q_\pi(s, a)$, obeys the Bellman optimality equation, as

$$Q^*(s, a) = \mathbb{E}_{s^{t+1}}\left[r^{t+1} + \gamma \max_{a'} Q^*(s^{t+1}, a') | s^t = s, a^t = a\right], \tag{19}$$

where s^{t+1} is the new state after executing the state-action pair (s, a). Let $q(s, a)$ be the state action-value function in the iteration process. Q-learning updates $q(s^t, a^t)$ as follows.

$$q(s^{t+1}, a^{t+1}) \leftarrow q(s^t, a^t) + \delta\left[r^{t+1} + \gamma \max_{a'} q(s^{t+1}, a') - q(s^t, a^t)\right], \tag{20}$$

where δ is the learning rate.

Q-learning uses a Q-table to approximate the Q-function. When the state and action spaces are discrete and small, learning the optimal policy π is possible with Q-learning. However, when the state and action spaces become continuous and large, the problem becomes intractable. Deep Q-learning utilizes a deep Q-Network (DQN), i.e., a deep neural network (DNN), to approximate the mapping table. DQL inherits the advantages of both RL and deep learning.

Suppose the DQN is expressed as $q(:, :, \Theta^t)$, where Θ^t are the parameters of the DQN. As the *quasi-static target network* method implies [18], we define two DQNs: the target DQN with parameters Θ^t_{target} and the trained DQN with parameters Θ^t_{train}. Θ^t_{target} is updated to be equal to Θ^t_{train} once every T_u time slots. Using the target network can help stabilize the overall network performance. Instead of training with only the current experience, the DQN uses a randomly sampled mini-batch from the experience replay memory, which stores the recent tuples (s^t, a^t, r^t, s^{t+1}).

With experience replay, the least squares loss of training DQN for a sampled mini-batch D^t can be defined as

$$\mathcal{L}(\Theta^t_{train}) = \sum_{(s^t, a^t, r^t, s^{t+1})\in D^t}\left(y^t_{DQN}(r^t, s^{t+1}) - Q(s^t, a^t; \Theta^t_{train})\right)^2, \tag{21}$$

where the target output is

$$y^t_{DQN}(r^t, s^{t+1}) = r^t + \gamma \cdot \max_{a'} Q(s^{t+1}, a', \Theta^t_{target}). \quad (22)$$

This experience replay strategy ensures that the optimal policy will not lead to a local minimum. In each training step, the stochastic gradient descent algorithm is used to minimize the training loss (21) over the mini-batch D^t.

B. MULTI-AGENT DRL FOR RESOURCE ALLOCATION

In the resource sharing scenario illustrated in Fig. 1, multiple users attempt to transmit video data to the target receivers, which can be modeled as a multi-agent DRL problem. Each user is an agent and interacts with the unknown communication environment to gain experiences. The experience is then used to guide the transmit power control policy. At the first glance, the power allocation problem seems to be a competitive game. If each agent maximizes its transmit power, the other users may receive severe interference. In this paper, we turn this competitive game into a cooperative game by properly designing the reward function. This way, the global system performance can be optimized.

The multi-agent RL based approach is divided into two phases: (i) the offline training phase and (ii) the online implementation phase. We assume that the system is trained in a centralized way but implemented in a distributed manner. To be more specific, in the training phase, each agent adjusts its actions based on a system performance-oriented reward. In the implementation phase, each agent observes its local states and selects the optimal power control action.

As shown in Fig. 2, each agent n receives a local observation of the environment and then takes an action. These actions form a joint action vector. The agent then receives a joint reward and the environment evolves to the next state. The new local states are observed by the corresponding agents. When the reward is shared by all the agents, the cooperative behavior is encouraged.

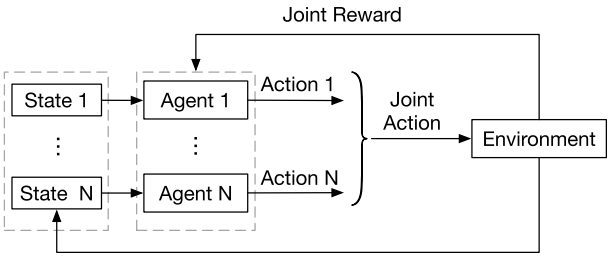


FIGURE 2. The multi-agent DRL model.

For each agent, the power control process is an MDP. Independent Q-learning [38] is one of the most widely used methods to solve the MDP problem with multiple agents. In independent Q-learning, each agent learns a decentralized policy based on its local observation and action, treating other agents as part of the environment. Note that, each agent would face a non-stationary problem as other learning agents are updating their policies simultaneously. One promising solution is to use single agent-DQN, which computes the joint actions for all agents [39]. However, the complexity will grow proportional to the size of the state-action space. Moreover, the single agent approach is not suitable for distributed implementation, which may limit its use in large networks. Recently, there has been several multi-agent DRL variants, however, there is no theoretical guarantees despite their promising empirical performance [40], [41]. In this paper, we limit the convergence analysis by providing simulation results in Section V, which is also employed in similar prior works [42], [43]. Specifically, we investigate the impact of the learning rate on the convergence performance.

C. MDP ELEMENTS

As depicted in Fig. 2, we proposed a multi-agent DRL approach where each user serves as an agent. In order to utilize the DRL for power control, the state space, the action space, and the reward function need to be properly designed.

1) STATE SPACE

At time slot t, the observed state for each agent is defined as $s^t = \{[I^t_1, I^t_2, \ldots, I^t_N], p^t_i, \Gamma^t_i\}$, where I^t_i is the indicator function, which shows whether the quality requirement of user i is satisfied or not. Specifically, it is defined as

$$I^t_i = \begin{cases} 1, & \text{if } Q^t_i > Q_{i,min} \\ 0, & \text{otherwise,} \end{cases} \quad (23)$$

p^t_i is agent i's current transmit power and Γ^t_i is the total interference that comes from the other agents, which is defined as

$$\Gamma^t_i = \sum_{j \neq i} |h_{ji}| \cdot p^t_j + \sigma^2_i. \quad (24)$$

Note that for each agent, p^t_i and Γ^t_i are local information that is readily available (no need for exchange).

2) ACTION SPACE

We assume that the transmitter of each agent chooses its transmit power from a finite set consisting of L elements,

$$\mathcal{A} = \left\{ \frac{p_{max}}{L}, \frac{2p_{max}}{L}, \ldots, p_{max} \right\}, \quad (25)$$

where p_{max} is the peak power constraint for each user. As a result, the dimension of the action space is L. The agent is only allowed to pick an action $a^t_i \in \mathcal{A}$ to update its transmit power. Increasing the size of action space may potentially increase the overall performance, meanwhile it also brings a larger training overhead and system complexity.

3) REWARD DESIGN

One reason that makes DRL appealing is its flexibility in handling the hard-to-optimize objective function. When the system reward is properly designed according to the objective function, the system performance can be improved. For our cross-layer video quality optimization problem, the objective

is to maximize the averaged users' quality while also satisfying the power constraints.

To achieve this goal, we define the reward function as follows [44]

$$r^t = \frac{1}{N}\sum_i q_i^t, \tag{26}$$

where

$$q_i^t = \begin{cases} Q_i^t, & \text{if } I_i^t = 1 \\ -100, & \text{otherwise,} \end{cases} \tag{27}$$

such that the user's video quality constraint (10) is satisfied.

So far we assume all the agents share the same reward r^t and the same action space s^t. In practice, such knowledge may be obtained at some additional communication cost. For the state space signal, the agents only need to monitor the ACK signal sent by each other to infer if the quality requirement is satisfied. The communication cost would be extremely low. For reward function design, each agent computes its own quality based on (1), (3), and (7), and then broadcast this information to other agents via message passing [30]. For large networks, the transmission of the exact quality value may occupy considerable wireless resources. A more feasible solution is that different agents observe only its nearby users' ACK signals and take the average quality among its neighboring users as the reward function. For example, we may design the state observed by agent i as

$$s_i^t = \left\{[I_n^t | n \in \mathcal{N}_i(K)], p_i^t, \Gamma_i^t\right\}, \tag{28}$$

where $\mathcal{N}_i(K)$ denotes the nearest K receivers (including agent i itself) of agent i. The reward function for each user can be designed as

$$r_i^t = \frac{1}{K}\sum_{j\in\mathcal{N}_i(K)} q_j^t. \tag{29}$$

This assumption is reasonable because in large networks, only nearby D2D users are in the same interference domain.

D. LEARNING ALGORITHM

1) TRAINING STAGE

We leverage deep Q-learning with experience replay to train multiple agents for optimal power control. It has been shown that Q-learning will converge to the optimal policy with probability 1 [45]. In deep Q-learning, DQN is used to approximate the action-value function. We assume that each agent maintains a dedicated DQN that takes an input of the current state and outputs the value functions corresponding to all actions.

The DQN is trained through multiple episodes. In each episode l, all agents concurrently explore the state-action space with the ϵ-greedy policy, i.e., the agent chooses the action that maximizes the estimated state-action value with probability ϵ^l and chooses a random action with probability $1-\epsilon^l$. The ϵ-greedy policy helps achieve a balance between exploitation of the current best Q-value function and exploration of a better option. Each episode consists of a maximum T steps. In each step t, all agents collect and store the state action and reward tuple, $(s_i^t, a_i^t, r^t, s_i^{t+1})$, in the experience replay. In each step, a mini-batch D^t is uniformly sampled from the replay memory. If all the users' video quality requirements are satisfied, the system randomly initialize the transmit power of all users and goes to the next episode. The training algorithm is presented in Algorithm 1.

Algorithm 1 Multi-Agent DRL Training Algorithm

```
Start environment simulator, generating channels;
Initialize Q-networks for all agents randomly;
Initialize p for all agents, and obtain s^0;
for each training episode do
    Randomly initialize the agents' transmit power;
    for each step do
        for each agent i do
            observe s_i^t;
            choose action a_i^t according to the ε-greedy policy;
        end for
        All agents take actions and receive reward r^{t+1};
        for each agent i do
            Update state s_i^{t+1};
            store (s_i^t, a_i^t, r^{t+1}, s_i^{t+1}) in replay memory D_i;
        end for
        for each agent i do
            Uniformly sample mini-batches from D_i;
            minimizing error between Q-network and the target network with stochastic gradient methods;
        end for
        if the QoE of each user is satisfied then
            Break;
        end if
    end for
end for
```

In the training stage, the agent reaches their target state if the action remains unchanged in the next state s_i^{t+1}. It is easy to show that the next state s_i^{t+1} is also a goal state. The agent will stay on the target state until the transmission is completed. As a result, the policy will converge, and we will obtain the largest estimated Q value.

2) IMPLEMENTATION STAGE

During the implementation stage, each agent observes the environment state and then selects an action, which maximizes the state-action value according to the trained Q-network. Afterwards, all agents transmit their video data with a proper power determined by their selected actions. The implementation algorithm is summarized in Algorithm 2. In most of the cases, each agent can reach their target state within 1 step. To solve the non-convergent problem, we add a testing loop. That is, if the agent cannot reach the target

Algorithm 2 DRL-Based Power Control Algorithm

Initialize the environment, agents randomly select initial power, and obtain the initial state s^0;

for each agent i **do**
  for each step **do**
    Select $a_i = \arg\max_{a \in \mathcal{A}} Q_i(s^0, a; \Theta_i^*)$;
    if the quality requirement of each user is satisfied **then**
      Break;
    end if
  end for
end for
Obtain the optimal power allocation $\mathbf{p} = [a_1, a_2, \ldots, a_N]$;

state, all the agents will explore the taken action based on the current state until all the agents' minimum quality is satisfied.

Since the training procedure can be performed offline over different episodes for different network topologies, video quality requirements, video types, and channel conditions, the heavy training complexity should not be a problem in practice. Meanwhile, the online implementation complexity is extremely low, which enables many real-time applications. In practice, the trained DQN can be updated only when the network topology and video sequences are dramatically changed.

IV. SYSTEM SETUP

We next carry out experiments to validate the performance of the proposed DRL-based power control method. The maximum power (in Watt) is set to $p_{max} = 0.4$ and L is 10. The bandwidth is set to $B = 500$kHz and the cell carrier frequency is set to 2.4GHz. The noise power density for each user is -174dBm/Hz. The distance between the transmitter and the receiver of each agent is fixed to be 50m. The agents are randomly located in a square area of 500m $\times$ 500m.

A. VIDEO CONFIGURATION

For simplicity, we assume our video library contains 2 video sequences in the common intermediate format (CIF). One video sequence is "Foreman," which has a low spatial-temporal content complexity. The other is "Football," which has a high spatial-temporal content complexity [35]. Each sequence is encoded by the High Efficiency Video Coding (HEVC) software [46]. We use the default low delay configurations to operate the encoder with both intra encoding and motion compensation. The group of picture (GOP) size is 4.

We enable rate control and change the target bit rate. Given a target bit rate, the video sequences are encoded into bit streams. We average the MSE between the reconstructed frame and the original frame over all the 20 frames. The PSNR value is then calculated based on (4). Based on the obtained samples, we estimate the video sequence parameters $\{\theta_i, \beta_i\}$ with a curve-fitting method. The estimated values for these parameters are listed in Table 1 and the corresponding rate-distortion curves are presented in Fig. 3.

TABLE 1. Optimal parameters for the two video sequences.

Video	θ	β
Foreman	2876	23.6
Football	13870	493.2

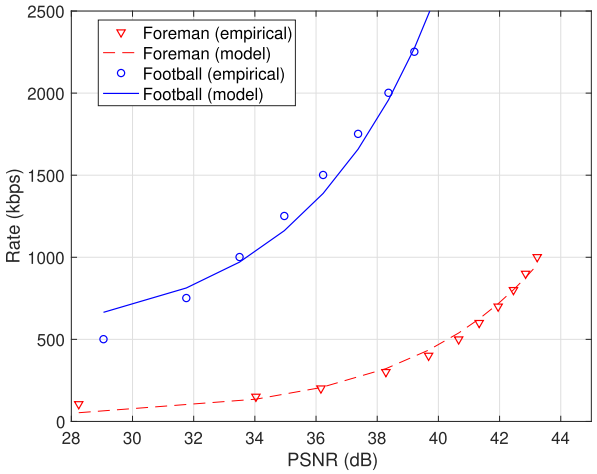


FIGURE 3. Rate-distortion curve for the two video sequences.

It can be seen that different video sequences generally exhibit quite different behaviors. For example, the rate of video "Football" increases rapidly with increased PSNR value, while the video sequence "Foreman" grows quite slowly. With the same transmission rate (e.g., 500kbps), the user who requests video sequence "Football" has a PSNR value of 29dB, while the users who request video sequence "Foreman" can enjoy a video quality up to around 41dB. Hence, simply perform physical layer resource optimization may not be optimal. A cross-layer optimization is indispensable.

B. DRL PARAMETERS

In our experiments, we choose a deep neural network (DNN) to approximate the action-value function. The DNN consists of three fully connected hidden layers, while each layer contains 32, 32, and 16 neurons, respectively. The rectified linear units (ReLUs) are used as the activation function. We adopt the Adam algorithm for loss optimization. The replay memory size is set to 200. The batch size is set to 8. The probability of exploring new actions linearly decreases with the number of episodes, from 0.9 to 0 for the first 1000 episodes. Algorithm 1 is used to train the network and Algorithm 2 is used for distributed implementation. The hyper-parameters are listed in Table 2.

V. SIMULATION RESULT AND DISCUSSIONS

A. TWO USERS

First of all, we consider the simplest case where there are two users. The distance matrix is randomly generated. For this

TABLE 2. DRL hyper-parameters.

Hyper-parameter	Value
Experience replay memory size	200
Experience replay mini-batch size	8
Learning rate δ used in *RMSProp*	0.5
Discount factor γ	0.9
ϵ in the ϵ-greedy policy	from 0.9 linearly decreases to 0
number of training episodes	3000
Target network update interval	5 s
Number of steps per episode	500

experiment, the distance matrix is generated as

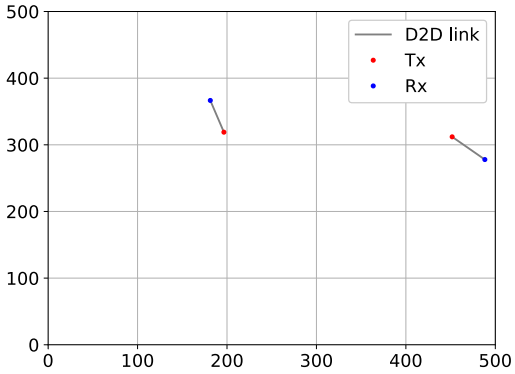
$$\mathbf{D} = \begin{bmatrix} 50 & 275 \\ 294 & 50 \end{bmatrix}. \tag{30}$$

The layout of the two users is shown in Fig. 4. The channel fading follows the free space propagation model, which is defined in (2). We also assume that the two users request video sequences "Football" and "Foreman" respectively. The quality requirement $Q_{i,min}$ for both videos is set to 42dB.

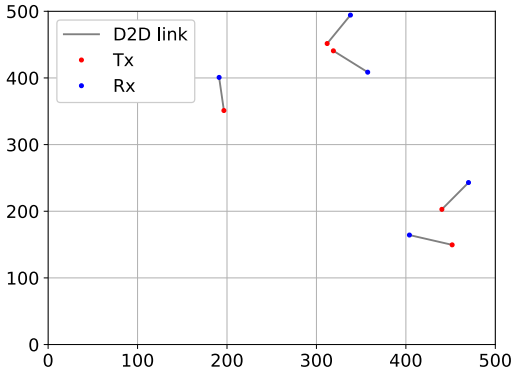
Our aim is to ensure that the average quality of all the users is maximized while each user's minimum quality requirement is also satisfied. Fig. 5 shows the training loss calculated by (21) for different δ values. It can be seen that with a large learning rate, the training loss converges to 0 quickly, while with a smaller learning rate value, the loss may converge slow. For example, when $\delta = 0.01$, even after 1500 episodes, the training loss still does not converge to 0. Meanwhile, when δ is moderate, e.g., $\delta = 0.5$, the training loss is generally very small across all the episodes. This is also confirmed by the training reward for different values of δ plotted in Fig. 6. When $\delta = 0.01$, the reward does not converge to a positive value, which means there is a penalty induced by that situation such that the user's quality requirement is not satisfied. If we choose δ to be 0.5 or 1, the training reward will stay at a stable state. However, the reward value corresponding to $\delta = 0.5$ is slightly larger than the reward value corresponding to $\delta = 1$. Based on these observations, a moderate value of learning rate is preferred. In our experiments, we choose $\delta = 0.5$.

Fig. 7 presents the two users' video quality performance versus the number of training episodes. We observe that at the beginning of the training stage, the users' video quality fluctuates slightly. This is because at first the ϵ value is large, the agents tend to explore new actions. As with more iterations, ϵ starts to decrease from 0.9 to 0. During this stage, the agent keeps on exploring the unknown environment while also exploiting the gained knowledge to train the target network. After 1000 episodes, the value of ϵ decreases to 0, the agent will stop exploring the environment; instead, it will choose the actions that have achieved the maximum state-action values. As a result, the quality curves for the two users remain stable.

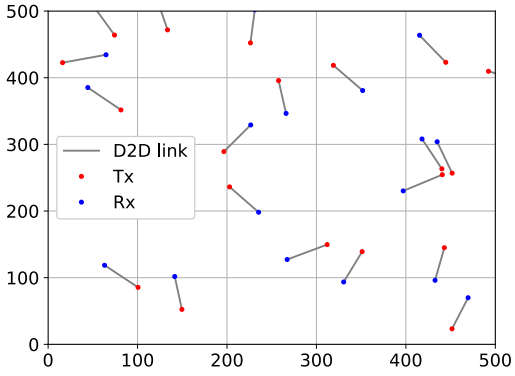
We next consider the distributed implementation stage. Note that the training stage may involve a high computational



(a) $N = 2$



(b) $N = 5$



(c) $N = 20$

FIGURE 4. Layout of the video users used in the simulations.

complexity. After the training process is done offline, the distributed online deployment should be very easy and fast. Fig. 8 demonstrates the performance of the proposed method. As benchmarks for the proposed algorithm, we introduced two baseline algorithms:

1) Random power method: each user randomly selects a transmit power from $\mathcal{A}$;
2) Maximum power method: each user transmits its video sequences at the maximum power p_{max}.

We perform 1000 simulations and the users' PSNR values for the first 25 simulations are plotted in Fig. 8. In each

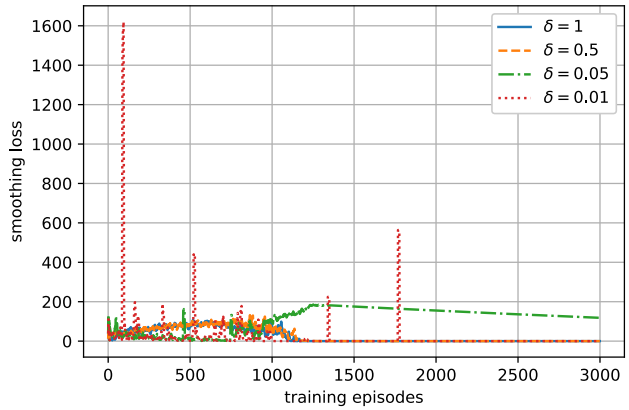


FIGURE 5. Loss function versus the number of training episodes ($N = 2$).

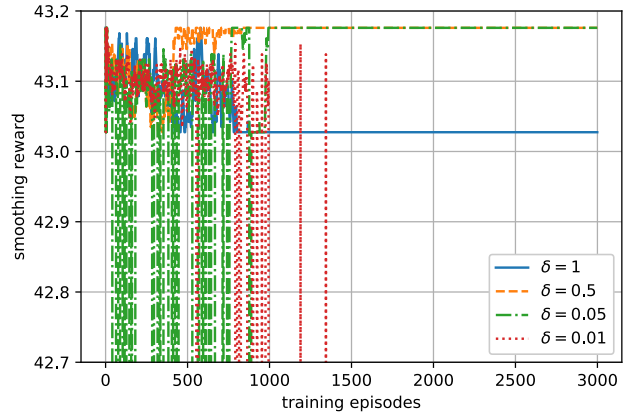


FIGURE 6. Reward versus the number of training episodes ($N = 2$).

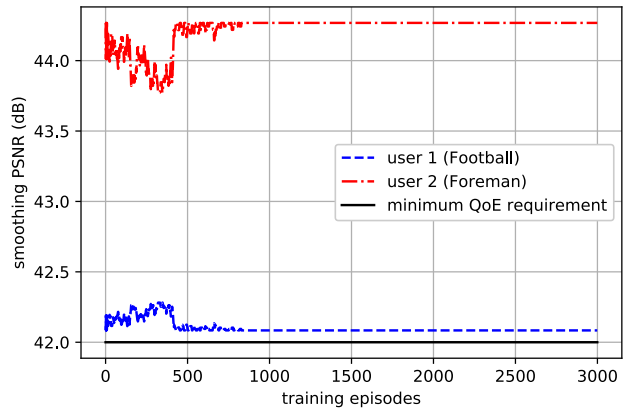


FIGURE 7. Users' QoE versus the number of training episodes ($N = 2$).

testing episode, the agent initializes the state by randomly generating transmit powers. Then both agents observe the state and take action according to the state-action value. Simulation results show that the agents can converge to the optimal action with 1 step from any initial state. In this process, no channel estimation is needed and no iterations are required. Hence this approach is quite fast. Moreover, with the DRL-based approach, both users' required quality

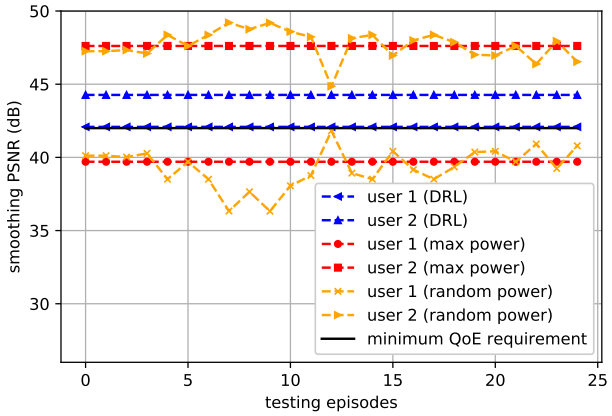


FIGURE 8. Users' QoE versus the number of testing episodes ($N = 2$).

are satisfied. As a comparison, the maximum power method and the random power method can only guarantee one user's quality requirement, while the other user's quality is below the minimum requirement.

To better compare these algorithms, we define the success rate as the ratio of the number of successful trials to the total number of tests. In all the 1000 simulations, the proposed DRL method achieves a success rate of 100%, but the success rate of the random power method is only 3% and the success rate of the maximum power method is 0. In practice, when the number of users is small and the action space is small, the users can randomly choose powers by trial and error and eventually obtain a feasible solution if the feasible solution exists. However, frequent information exchange and complex iterations are usually required, which would pose additional delays. When the network size grows large and the number of action space becomes large, the random power allocation method will no longer work. We will demonstrate this point in the next subsection.

B. LARGER NUMBER OF USERS

Now we consider the case of 5 users in the system; the layout of the users is shown in Fig. 4(b). The distance matrix is randomly generated as

$$\mathbf{D} = \begin{bmatrix} 50 & 361.9 & 362.9 & 275.7 & 95.2 \\ 279.1 & 50 & 201.3 & 170.8 & 294.1 \\ 301.7 & 131 & 50 & 62.5 & 261.7 \\ 289.2 & 133.9 & 56.9 & 50 & 248.8 \\ 53.0 & 318.1 & 308.8 & 221.8 & 50 \end{bmatrix}. \quad (31)$$

We assume that the first user requests video sequence "Football" with a minimum quality requirement of 34dB and the rest four users request the video sequence "Foreman" with a minimum quality requirement of 40dB. The training episode is set to be 10000. ϵ is linearly decreased from 0.9 to 0 for the first 3000 episodes.

The training loss and the training reward are depicted in Fig. 9 and Fig. 10, respectively. It can be seen that after around 3000 episodes, the training loss converges to 0. The reward approximates 40, which means there is no penalty and

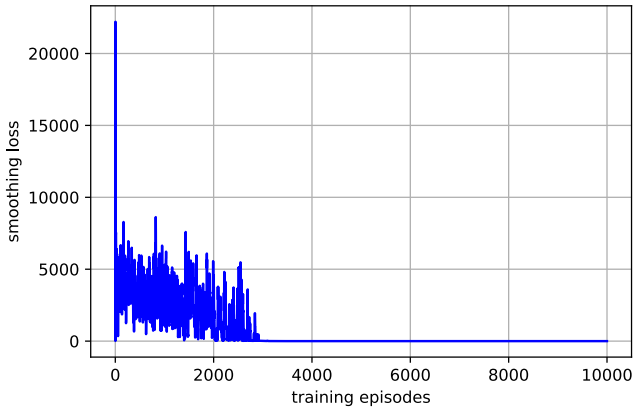


FIGURE 9. Loss function versus the number of training episodes ($N = 5$).

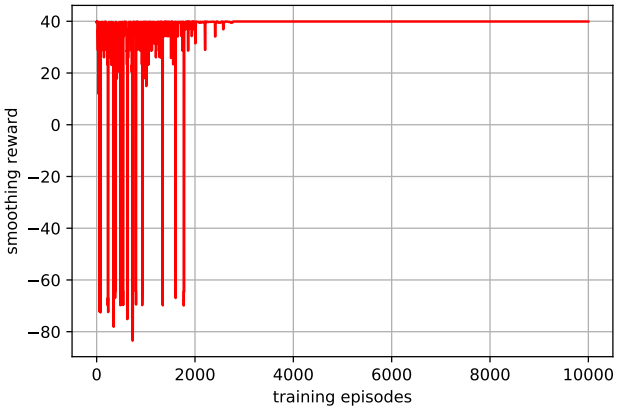


FIGURE 10. Reward versus the number of training episodes ($N = 5$).

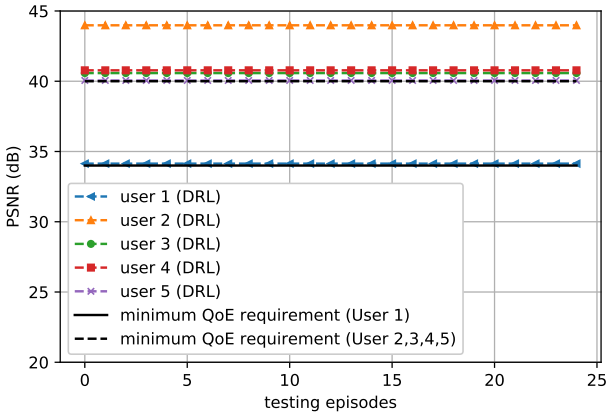


FIGURE 11. Users' QoE versus the number of testing episodes with the proposed method ($N = 5$).

all the users' quality requirements are satisfied in the training stage. In the distributed implementation stage, we perform 1000 testing episodes and the agents initialize their states randomly in each episode. The PSNR performance of each user for the first 25 episodes is depicted in Fig. 11. We find that all the agents can observe their local environments and reach their target QoE within 1 step. The success rate of the proposed multi-agent DRL approach has a success rate

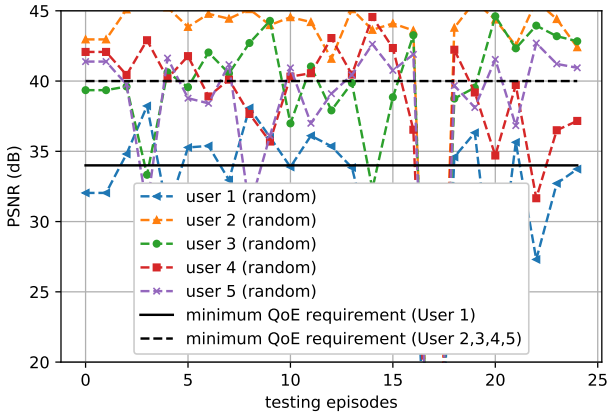


FIGURE 12. Users' QoE versus the number of testing episodes with the random power method ($N = 5$).

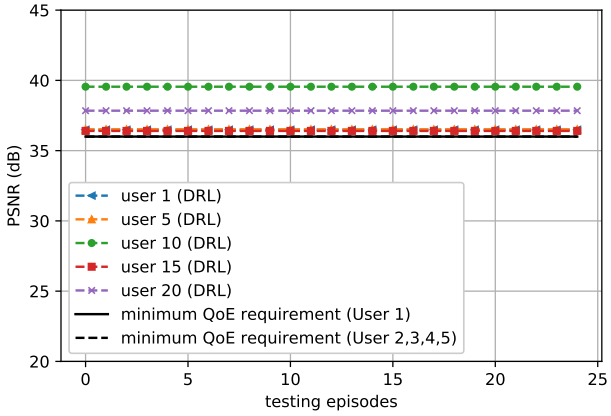


FIGURE 13. Users' QoE versus the number of testing episodes with the proposed method. ($N = 20$).

of 100% across all the testing episodes. As a comparison, the random power method and the maximum power method has a success rate of 0. We depicted the performance of the random power method in Fig. 14. In practice, for the proposed multi-agent DRL approach, we find that the agents may face a non-stationary problem, i.e., the trained DQN for each agent may not be able to reach the target state within 1 step. For example, when we set the learning rate $\delta = 0.1$, in most of the cases, the agent can reach a target state within 1 step with the trained DQN from arbitrary initial states. However, there are a few cases when agent cannot reach the target state within 1 step. This may be caused by the experience replay sampling process. The sampled DQN from the experience replay may not reflect the current dynamics. So far, there is no theoretical solutions which can solve this problem. Possible heuristic solutions include adding the training ϵ into the state [30], finding a proper value of the learning rate or in the testing stage we perform more iterations until the obtained state is feasible.[1]

[1]We add iterations in Algorithm 2 for each testing episode. If the current state is not the target state, all the agents perform a further action based on the current step until they reach the target state. In each iteration, only ACK signals are needed; other agents' quality requirements are not needed. So the communication cost is low compared to the training process.

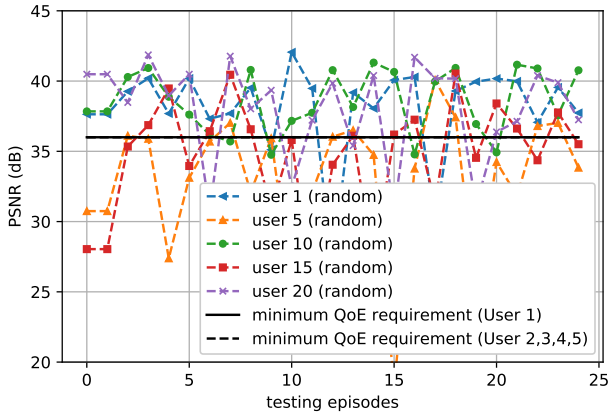


FIGURE 14. Users' QoE versus the number of testing episodes (N = 20) with the random power allocation method. ($N = 20$).

Now we consider a more challenging task where there are 20 users as shown in Fig. 4(c). Their locations are randomly generated. For simplicity, we assume that all of users request the same video sequence "Foreman" and their minimum quality requirement is set to 36dB. To better control the complexity, we assume that each agent only observe state from the nearest 5 neighbors, i.e., $K = 5$. The corresponding testing stage is shown in Fig. 13. Due to space limitation, we only plot 5 users' PSNR values. Actually, all the 20 users' video quality requirements are satisfied and their average quality is maximized. As a comparison, we present the PSNR for the random power allocation method in Fig. 14, where the users' PSNR values are obviously not stable. In some cases, the user's PSNR falls below 30dB. The success rates of both the random power allocation method and the maximum power allocation method are 0.

VI. CONCLUSION AND FUTURE WORK

In this paper, we studied the quality-aware power allocation problem for multi-user video streaming. We developed a distributed model-free power allocation algorithm, which help maximize the users' target quality. The proposed method does not require explicit channel state information, which would save significant resources. Experiment results showed that the developed multi-agent DRL approach can guarantee that all the users achieve their target quality requirements within few steps and the users' average quality is maximized. For future investigations, possible directions include

1) the randomness of the layout of the D2D channels and the content of the requested videos could be considered in the training process. The agent will take the channel state and the video contents as local state information. Efficient training algorithms need to be developed so that users can take action based on the local observation and the users' average quality could be maximized.
2) Currently, we start the training process based on the assumption that there exists at least one feasible solution. Theoretical methods should be provided to guarantee a quick examination to check if there exists a feasible solution before the training process.
3) Users may work on different channels in practice. In the future, a DRL based joint spectrum and power allocation method could be developed.

REFERENCES

[1] Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," Cisco, San Jose, CA, USA, Feb. 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html
[2] Y. Xu and S. Mao, *Mobile Cloud Media: State of the Art and Outlook*. Hershey, PA, USA: IGI Global, 2013, ch. 2, pp. 18–38.
[3] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in LTE-advanced networks: A survey," *IEEE Commun. Surv. Tutr.*, vol. 17, no. 4, pp. 1923–1940, 4th Quart., 2015.
[4] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.
[5] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 86–92, May 2014.
[6] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi, "Design aspects of network assisted device-to-device communications," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 170–177, Mar. 2012.
[7] M. Chiang, P. Hande, T. Lan, and C. W. Tan, "Power control in wireless cellular networks," *Found. Trends Netw.*, vol. 2, no. 4, pp. 381–533, Apr. 2008.
[8] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.
[9] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.
[10] X. Chen, Z. Zhao, and H. Zhang, "Stochastic power adaptation with multiagent reinforcement learning for cognitive wireless mesh networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2155–2166, Nov. 2013.
[11] G. Zhang, J. Hu, W. Heng, X. Li, and G. Wang, "Distributed power control for D2D communications underlaying cellular network using Stackelberg game," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
[12] Z. He, S. Mao, and T. Jiang, "A survey of QoE-driven video streaming over cognitive radio networks," *IEEE Netw.*, vol. 29, no. 6, pp. 20–25, Nov./Dec. 2015.
[13] M. Amjad, M. H. Rehmani, and S. Mao, "Wireless multimedia cognitive radio networks: A comprehensive survey," *IEEE Commun. Surveys Tutr.*, vol. 20, no. 2, pp. 1056–1103, 2nd Quart., 2018.
[14] X. Jiang, H. Lu, and C. W. Chen, "Enabling quality-driven scalable video transmission over multi-user NOMA system," in *Proc. IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 1952–1960.
[15] H. Lu, M. Zhang, Y. Gui, and J. Liu, "QoE-driven multi-user video transmission over SM-NOMA integrated systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 9, pp. 2102–2116, Sep. 2019.
[16] S. Cicalo and V. Tralli, "Distortion-fair cross-layer resource allocation for scalable video transmission in OFDMA wireless networks," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 848–863, Apr. 2014.
[17] T. Zhang and S. Mao, "Joint power and channel resource optimization in soft multi-view video delivery," *IEEE Access*, vol. 7, pp. 148084–148097, 2019.
[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[19] 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, D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
[20] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tutr.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019.

[21] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.
[22] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.
[23] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.
[24] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.
[25] 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.
[26] J. Liu, B. Krishnamachari, S. Zhou, and Z. Niu, "DeepNap: Data-driven base station sleeping operations through deep reinforcement learning," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4273–4282, Dec. 2018.
[27] K. Xiao, S. Mao, and J. K. Tugnait, "TCP-Drinc: Smart congestion control based on deep reinforcement learning," *IEEE Access*, vol. 7, pp. 11892–11904, 2019.
[28] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen, and H. Li, "Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach," *IEEE Access*, vol. 6, pp. 25463–25473, 2018.
[29] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.
[30] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," May 2019, *arXiv:1905.02910*. [Online]. Available: https://arxiv.org/abs/1905.02910
[31] M. Feng and S. Mao, "Dealing with Limited Backhaul Capacity in millimeter-wave systems: A deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 50–55, Mar. 2019.
[32] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.
[33] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.
[34] H. Mansour, V. Krishnamurthy, and P. Nasiopoulos, "Channel aware multiuser scalable video streaming over lossy under-provisioned channels: Modeling and analysis," *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1366–1381, Nov. 2008.
[35] K. Lin and S. Dumitrescu, "Cross-layer resource allocation for scalable video over OFDMA wireless networks: Tradeoff between quality fairness and efficiency," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1654–1669, Jul. 2017.
[36] S. Cicalò, A. Haseeb, and V. Tralli, "Fairness-oriented multi-stream rate adaptation using scalable video coding," *Signal Process., Image Commun.*, vol. 27, no. 8, pp. 800–813, 2012.
[37] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 57–73, Feb. 2008.
[38] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. ICML*, Amherst, MA, USA, Jun. 1993, pp. 330–337.
[39] J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson, "Stabilising experience replay for deep multi-agent reinforcement learning," in *Prof. ICML*, Sydney, NSW, Australia, Aug. 2017, pp. 1146–1155.
[40] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multi-agent systems: A review of challenges, solutions and applications," Dec. 2018, *arXiv:1812.11794*. [Online]. Available: https://arxiv.org/abs/1812.11794
[41] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent cooperation and competition with deep reinforcement learning," *PLoS ONE*, vol. 12, no. 4, Apr. 2017, Art. no. e0172395.
[42] U. Challita, L. Dong, and W. Saad, "Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4674–4689, Jul. 2018.
[43] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Multi-agent reinforcement learning: Independent vs. cooperative agents," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.
[44] F. A. Asuhaimi, S. Bu, P. V. Klaine, and M. A. Imran, "Channel access and power control for energy-efficient delay-aware heterogeneous cellular networks for smart grid communications using deep reinforcement learning," *IEEE Access*, vol. 7, pp. 133474–133484, 2019.
[45] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[46] *High Efficiency Video Coding (HEVC)*. Accessed: Nov. 13, 2019[Online]. Available: https://hevc.hhi.fraunhofer.de/

TICAO ZHANG received the B.E. and M.S. degrees from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Auburn University. His research interests include video coding and communications, machine learning, and optimization and design of wireless multimedia networks.

SHIWEN MAO (S'99–M'04–SM'09–F'19) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA (now The New York University Tandon School of Engineering).

He joined Auburn University, Auburn, AL, USA, as an Assistant Professor, in 2006, was the McWane Associate Professor, from 2012 to 2015, and has been the Samuel Ginn Distinguished Professor with the Department of Electrical and Computer Engineering, since 2015. He is currently the Director of the Wireless Engineering Research and Education Center, Auburn University, since 2015, and the Director of the NSF IUCRC FiWIN Center Auburn University site, since 2018. His research interests include wireless networks, multimedia communications, and smart grid. He is a Distinguished Speaker (2018–2021) and was a Distinguished Lecturer (2014–2018) of the IEEE Vehicular Technology Society.

Dr. Mao received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award, in 2019, the IEEE ComSoc MMTC Distinguished Service Award, in 2019, the Auburn University Creative Research and Scholarship Award, in 2018, the 2017 IEEE ComSoc ITC Outstanding Service Award, the 2015 IEEE ComSoc TC-CSR Distinguished Service Award, the 2013 IEEE ComSoc MMTC Outstanding Leadership Award, and the NSF CAREER Award, in 2010. He is a co-recipient of the IEEE ComSoc MMTC Best Journal Paper Award, in 2019, the IEEE ComSoc MMTC Best Conference Paper Award, in 2018, the Best Demo Award from the IEEE SECON 2017, the Best Paper Awards from the IEEE GLOBECOM 2019, 2016, and 2015, the IEEE WCNC 2015, and the IEEE ICC 2013 and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is an Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, the IEEE INTERNET OF THINGS JOURNAL, the IEEE/CIC CHINA COMMUNICATIONS, and the *ACM GetMobile*, as well as an Associate Editor of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE MULTIMEDIA, the IEEE NETWORKING LETTERS, and the *Digital Communications and Networks Journal* (Elsevier).