A Multi-agent Reinforcement Learning Based Power Control Algorithm for D2D Communication Underlaying Cellular Networks

Wentai Chen and Jun Zheng(✉)

National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, Jiangsu, People's Republic of China
{wtchen, junzheng}@seu.edu.cn

Abstract. This paper considers the power control problem in device-to-device (D2D) communication underlaying a cellular network and explores the application of the machine learning (ML) approach in power control for improving the system throughput. Two multi-agent reinforcement learning (MARL) based algorithms are proposed for performing power control of D2D users (DUs): centralized Q-learning algorithm and distributed Q-learning algorithm. In the centralized algorithm, all DU pairs sharing the same RB use a common Q table in the learning process, while in the distributed algorithm each DU pair maintains its own Q table. Simulation results show that both the centralized algorithm and the distributed algorithm can converge to the same optimum Q values, and the distributed algorithm can converge faster than the centralized algorithm. Moreover, both the proposed Q-learning algorithms outperform the random power control algorithm in terms of the system throughput and satisfaction ratio.

Keywords: D2D communication · Power control · Multi-agent reinforcement learning · MARL · Q learning

1 Introduction

D2D communication has widely been considered as one of the promising technologies for 5G mobile cellular networks and beyond. In device-to-device (D2D) communication, a couple of closely located mobile devices are allowed to build direct connection, and communicate directly with each other with no need to pass through a base station (BS). As a result, D2D communication can effectively offload BS' traffic load, increase spectral efficiency and system throughput, reduce data transmission latency, and extend device battery lifetime [1]. To increase spectral efficiency, D2D users are usually allowed to share the spectrum resources of cellular users for communication. However, reusing spectrum resources would lead to severe interference between D2D users and cellular users. To achieve good system performance, it is critical to effectively mitigate such interference through efficient resource management, including spectrum allocation and power control. In the context of power control, extensive work has been conducted to study the power control problem in D2D communication [2–5]. However, most

S. Han et al. (Eds.): AICON 2019, LNICST 287, pp. 77–90, 2019.
https://doi.org/10.1007/978-3-030-22971-9_7

existing work uses traditional approaches in solving the problem. With recent advances in artificial intelligence (AI), the machine learning (ML) approach is receiving an increasing attention in the area of wireless communication. Even so, the study on the application of the ML approach in resource management for D2D communication is still limited. It is interesting to further explore the advanced ML approach in resource management for improving the performance of D2D communication, which motivated us to conduct this work.

In this paper, we consider the power control problem in D2D communication underlaying a cellular network and explore the application of the ML approach in solving the problem. Two multi-agent reinforcement learning (MARL) based power control algorithms are proposed for D2D communication: centralized Q-learning algorithm and distributed Q-learning algorithm. In the centralized algorithm, all D2D user (DU) pairs sharing the same resource block (RB) use a common Q table in the learning process, which makes the time and space complexity for calculating the value of the Q table exponentially increase as the number of DU pairs grows. To solve the problem, the distributed algorithm allows each DU pair to maintain a Q table of its own, which can dramatically reduce the time and space complexity for calculating the value of the Q table. Simulation results are shown to evaluate the performance of the proposed MARL-based power control algorithms in terms of the system throughput and satisfaction ratio.

The rest of this paper is organized as follows. Section 2 reviews recent related work. Section 3 describes the system model and formulates the power control problem. Section 4 presents the proposed power control algorithms. Section 5 shows simulation results to evaluate the performance of the proposed algorithms. Section 6 concludes this paper.

2 Related Work

The power control problem has been widely studied for D2D communication underlaying cellular networks in the literature [2–5]. In [2], Lee et al. proposed two centralized and distributed power control schemes for a D2D communication system. The former sets a limit on the interference of D2D users to ensure the cellular users to work with sufficient coverage probability; the latter uses an optimal on-off power control strategy, which maximizes the throughput of the D2D links. In [3], Ren et al. considered a vehicle-to-vehicle (V2V) communication system supported by a D2D underlaying cellular system (D2D-V), and introduced a power control framework based on convex function programming to achieve the optimal performance in terms of the system sum rate. In [4], Silva and Fodor proposed a binary power control (BPC) scheme for D2D communications. An objective function is introduced considering the power consumption for BPC, and a sub-optimal BPC solution is obtained to D2D power control problem. In [5], Sun et al. proposed a novel power control scheme based on stochastic channel-state information (CSI), and employed the opportunistic access control to reduce the interference caused by D2D communication and maximize the area energy efficiency, which overcomes the difficulty to acquire real-time CSI.

Recent advances in the ML technology have attracted much interest in using ML in D2D power control and several ML-based power control algorithms have already been proposed in the literature [6–8]. In [6], two centralized and distributed Q-learning algorithms were proposed for a single-cell cellular system with D2D users sharing the same resource blocks. In the proposed algorithms, all the D2D users were treated as different agents and the goal of the algorithms is to select the optimal D2D transmission power to maximize the system throughput by maintaining a two-state Q table. In [7], a cooperative reinforcement learning algorithm was proposed for the adaptive power allocation problem. The state action reward state action (SARSA), one of the on-policy reinforcement learning algorithms, was used to simulate the power control decision process of each D2D agent. The learning process is similar to that in [6], while the system model and state information in SARSA are more sophisticated. In [8], a Q-learning based power control algorithm was proposed for a single cell with a single cellular user. However, the system model is not realistic and the components of Q-learning were not well designed.

3 System Model and Problem Formulation

In this section, we first describe the system model and then formulate the power control problem in D2D communication considered in this paper.

3.1 System Model

We consider a single-cell cellular system consisting of one base station (BS), a set of M cellular users (CUs), and N D2D user (DU) pairs. The set of M CUs is denoted by $\mathcal{C} = \{C_1, C_2, ..., C_M\}$ and the set of N DU pairs is denoted by $\mathcal{D} = \{D_1, D_2, ..., D_N\}$. Here, a DU pair consists of the transmitter (T_x) of one DU and the receiver (R_x) of another DU, which communicate with each other without passing through the BS. The system has K orthogonal resource blocks (RBs), which are denoted by $\mathcal{B} = \{B_1, B_2, ..., B_K\}$. We assume that the DU pairs work in an underlay mode and are allowed to share the uplink spectrum resources (i.e., RBs) of CUs. Each CU occupies one RB which can be shared by multiple DU pairs, and one DU pair can only occupy one RB. The BS is able to obtain the CSI of both CUs and DU pairs. Moreover, we assume that the transmission power of each CU is fixed to p_c and that of each DU is adjustable. Each DU can select a transmission power level from a set of values, which is denoted by $\mathcal{P} = \{p_1, p_2, ..., p_L\}$.

Considering that each DU pair is allowed to share the uplink RBs of CUs, there exist three types of interference in the system, which are illustrated in Fig. 1:

(1) I_1: the interference from the transmitter T_x of a DU pair to the BS;
(2) I_2: the interference from a CU to the receiver R_x of a DU pair, where the CU and the DU pair share the same RB;
(3) I_3: the interference from the transmitter T_x of one DU pair to the receiver R_x of another DU pair, where the two DU pairs share the same RB.

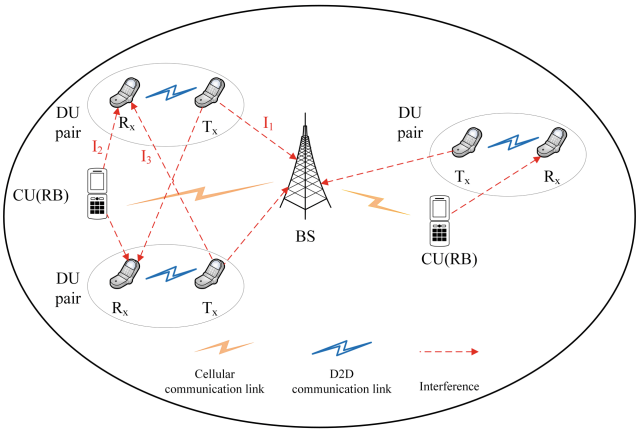

Fig. 1. System model

3.2 Problem Formulation

In this paper, we focus on the power control problem in D2D communication in the single-cell cellular system described in Fig. 1. For simplicity, we assume that the RB allocation is fixed. Specifically, we assume that $M = K$. Each CU is allocated a different RB and each DU pair is randomly allocated one RB.

Before we formulate the power control problem, we first analyze the signal to interference plus noise ratio (SINR) at a CU and at the receiver of a DU, respectively. For a CU that occupies the rth RB, the SINR at the CU is given by

$$SINC_{C_i}^r = \frac{p_{C_i}^r \cdot G_{C_i}^r}{\sigma^2 + \sum\limits_{D_j \in \mathcal{D}^r} p_{D_j}^r \cdot G_{D_j}^r}, i = 1, 2, \ldots, M; j = 1, 2, \ldots, N \tag{1}$$

where C_i denotes the ith CU, D_j denotes the jth DU pair; $\mathcal{D}^r$ denotes the set of DU pairs that share the rth RB; $p_{C_i}^r$ and $p_{D_j}^r$ denote the transmission power of C_i and D_j which share the rth RB, respectively; $G_{C_i}^r$ and $G_{D_j}^r$ denote the channel gains on the rth RB from the BS to C_i and D_j, respectively;σ^2 is the noise variance.

Similarly, for a DU pair that shares the rth RB, the SINR at the receiver of the DU pair is given by

$$SINC_{D_j}^r = \frac{p_{D_j}^r \cdot G_{D_j D_j}^r}{\sigma^2 + p_{C_i}^r \cdot G_{C_i D_j}^r + \sum\limits_{\substack{D_k \in \mathcal{D}^r \\ k \neq j}} p_{D_k}^r \cdot G_{D_k D_j}^r}, \tag{2}$$

where $G_{D_j D_j}^r$, $G_{C_i D_j}^r$, and $G_{D_k D_j}^r$ denote the channel gain on the link from the transmitter of D_j to the receiver of D_j, the channel gain from C_i to the receiver of D_j, and the channel gain from the transmitter of D_k to the receiver of D_j, respectively.

Next we formulate the power control problem. Given a set of available power levels $\mathcal{P} = \{p_1, p_2, ..., p_L\}$ and RB allocation for CUs and DU pairs, the power control problem under consideration is to find a set of optimal power levels $\mathcal{P}_D^* = \{p_{D_1}^*, p_{D_2}^*, \cdots, p_{D_N}^*\}$ for all the DU pairs so that the overall system throughput is maximized, i.e.,

$$\text{Objective}: \quad \max \sum_{r=1}^{K} \{\log_2(1 + SINR_{C_i}^r) + \sum_{D_j \in \mathcal{D}^r} \log_2(1 + SINR_{D_j}^r)\} \tag{3}$$

$$\text{subject to} \quad \text{SINR}_{C_i}^r \geq \tau_0, \tag{4}$$

$$p_1 \leq p_{D_j}^r \leq p_L, \qquad \forall j, r \tag{5}$$

where τ_0 denotes the minimum SINR requirement of a CU, constraint (4) ensures the SINR requirement of each CU, and constraint (5) ensures that the transmission power of each DU is limited to the range $[p_1, p_L]$.

In the next section, we will present two MARL-based power control algorithms to solve the above power control problem.

4 MARL-Based Power Control Algorithm

In this section, we first introduce the concepts of reinforcement learning and then present two MARL-based power control algorithms: centralized Q-learning and distributed Q-learning.

4.1 Reinforcement Learning

Reinforcement learning is an important branch of machine learning. A standard RL problem can be represented by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R)$, where $\mathcal{S}$ denotes a set of states; $\mathcal{A}$ denotes a set of actions that can be selected by an agent; $\mathcal{T}$ denotes a set of transition probabilities from one state to another, and R denotes the reward function. A standard RL process is illustrated in Fig. 2.

In a standard RL process, an agent interacts with the environment in a sequence of episodes, which are denoted by $t = 0, 1, 2, \ldots$ There are three steps in each episode. In step (1), the agent receives the current state S_t and reward R_t. In step (2), it takes the action A_t. In step (3), the environment transfers to another state S_{t+1}, and gives a new reward R_{t+1}. The agent starts learning from an initial state S_0, and continues the episodes until the learning process converges.

In the above learning process, the agent selects its action in each episode according to a policy π, which is given by

$$\pi_t(a|s) = P(A_t = a|S_t = s), \quad a \in \mathcal{A}, s \in \mathcal{S}, \tag{6}$$

where $P(A_t = a | S_t = s)$ is the probability of selecting action a at state s. By selecting different actions and updating the current policy in each episode, the agent is able to make a better decision and reaches the optimal policy π^* after a number of episodes.

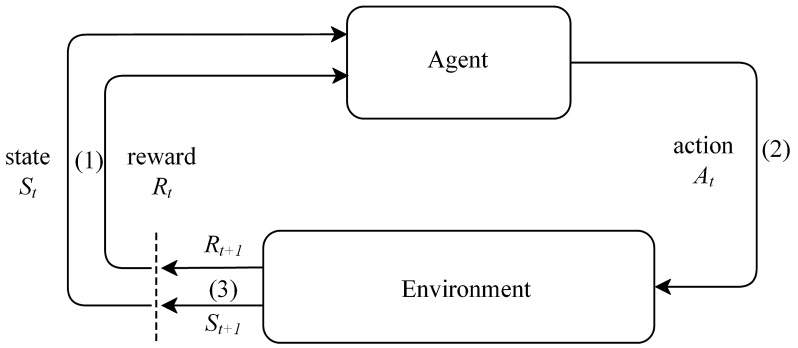

Fig. 2. Standard reinforcement learning process

To find an optimum policy, we introduce a value function $V_\pi(s)$ to determine the value of a state s under a given policy π, which is defined as the expectation of the discounted sum of the rewards in future episodes, i.e.,

$$V_\pi(s) = E_\pi(\sum_{i=1}^{\infty} \eta^{i-1} R_{t+i} | S_t = s), \tag{7}$$

where $E_\pi(\cdot)$ denotes the expected value of a random variable given that the agent follows the policy π, η is the discount rate, and R_{t+i} is the reward in the $(t + i)$th episode. The discounted sum of rewards in future episodes reflects an important feature of RL: delayed reward, i.e., the action selected by each agent relies on not only the immediate reward, but also all the rewards in subsequent episodes. The value function represents how good it is to perform a given action in a given state, and thus can be used to evaluate the effectiveness of the policy. A higher value of $V_\pi(s)$ means a better policy for the agent. A policy is called an optimal policy if the corresponding value function is higher than any other value functions.

Q learning is a typical form of reinforcement learning. Like all other RL algorithms, Q learning needs no prior knowledge about the environment. In Q learning, an agent learns how to behave based on the previous experience, which is traced by a Q function. The Q function is used to determine the value of an action a under a given state s and is defined as

$$Q_t(s, a) = E_\pi(\sum_{i=1}^{\infty} \eta^{i-1} R_{t+i} | S_t = s, A_t = a). \tag{8}$$

A Q function is represented by a two-dimensional table, in which each row represents a state of the environment, and each column represents an action of the agent.

The learning process starts from an initial state S_0, and the initial Q table is usually set to all-zero. At each episode, assuming the current state is $S_t = s$, the agent needs to select the best action $A_t = a$ according to the policy. After performing a, the agent receives a reward R_{t+1}, and the environment transfers to the next state S_{t+1}. It can be proved by induction that Q learning is able to converge to the optimal values if all the states can be visited infinitely as the learning process proceeds [9].

As assumed in Sect. 3.2, the RB allocation for CUs and DU pairs is fixed. Thus, the power control for the DU pairs on different RBs is independent. For this reason, the power control problem under consideration can be viewed as K independent power control sub-problems on K RBs. For a particular RB, all the DUs sharing the RB should select a proper power level from $\mathcal{P} = \{p_1, p_2, \ldots, p_L\}$ to reach the maximum throughput on this RB. To use Q learning to solve the problem, each DU pair can be considered as an individual agent. In this way, it becomes a multi-agent Q-learning problem. In the next two sections, we will present a centralized Q-learning algorithm and a distributed Q-learning algorithm to solve the problem.

4.2 Centralized Q-learning Algorithm for D2D Power Control

In this section, we present the proposed centralized Q-learning algorithm for D2D power control.

A. Component Definitions

We first define the basic components in the centralized Q-learning algorithm: agent, state, action, and reward. In the centralized Q-learning algorithm, an agent is defined as a DU pair in the cellular system. Thus, there are N agents in the whole system. An action of an agent is defined as the action that a DU pair takes to select a power level p from $\mathcal{P} = \{p_1, p_2, \ldots, p_L\}$. The joint actions for all DUs sharing a particular RB constitute a vector. A reward in the centralized Q-learning algorithm is defined as the conditional overall throughput of an RB in the cellular system. The value of a reward is determined using the following reward function:

$$R = \begin{cases} \log_2(1 + SINR^r_{C_i}) + \sum\limits_{D_j \in \mathcal{D}^r} \log_2(1 + SINR^r_{D_j}), & SINR^r_{C_i} \geq \tau_0 \\ -1 & \text{otherwise} \end{cases}. \tag{9}$$

According to Eq. (9), if the SINR requirement of C_i cannot be satisfied, i.e., $SINR^r_{C_i} < \tau_0$, the reward is set to -1 as a penalty term, which ensures the priority of C_i.

In a standard Q-learning algorithm, an agent needs to transfer between different states by selecting different actions, which usually takes a large number of episodes to converge. Furthermore, it is difficult to define the states in the Q-learning algorithm that matches the physical states in the single-cell cellular system [10]. According to [10], we use a single-state Q table in the centralized Q-learning algorithm and the state formulation is not needed.

B. Algorithm Description

The centralized Q-learning algorithm is an algorithm that is executed at the BS for performing power control for DU pairs. Unlike that in [6], it uses a single state Q table

in the learning process instead of a two-state Q table. In the learning process, the Q table for the rth RB is first set to all zeros. In each episode, all DU pairs on the rth RB select different power levels simultaneously, receive a reward, and update the Q table. The learning process continues until the Q table converges to the optimal values.

In each episode, an action is selected based on an ε-greedy strategy, which is described as follows:

- Select a random action with a probability ε;
- Select an action according to the maximum Q value of the current state with a probability $(1 - \varepsilon)$.

Here, ε is the threshold of the probability, which decays with the number of episodes as

$$\varepsilon = \varepsilon_{\min} + (\varepsilon_{\max} - \varepsilon_{\min}) \cdot \exp(-h \cdot t), \tag{10}$$

where $\varepsilon_{\max}$ and $\varepsilon_{\min}$ denote the upper limit and the lower limit of ε; h is a decay rate within [0, 1]; t is the index of the current episode. At the beginning of learning, ε is set to a value close to 1. Thus, the agent is likely to select a random action that it has not selected before to find more new states of the environment. As the learning process continues, the value of ε decreases accordingly, and thus the agent relies more on the learned policy. The ε-greedy strategy helps an agent explore more states and actions at the beginning of the learning process so that the convergence of Q learning can be ensured [9]. After an action is selected, the Q table is updated based on the following function:

$$Q^r_{t+1}(s,a) = Q^r_t(s,a) + \alpha[r_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q^r_t(s',a') - Q^r_t(s,a)], \tag{11}$$

where $Q^r_{t+1}(s,a)$ denotes the Q table in the $(t + 1)$th episode $Q^r_{t+1}(s,a)$; α is the learning rate ranging from 0 to 1, which decides how much the reward contributes in the update of the Q value; γ is the discount factor which varies from 0 to 1. The higher the value, the more the DU pairs rely on the future rewards than the current reward. According to Eq. (11), $Q^r_{t+1}(s,a)$ depends on the current Q value $Q^r_t(s,a)$, the reward for taking the action a under the state s, and the maximum future reward given the new state s' and all possible actions $a' \in \mathcal{A}$ under s'

The pseudo codes of the centralized Q-learning algorithm for D2D power control are given in Algorithm 1.

Algorithm 1 Centralized Q-learning algorithm for D2D power control

Input:

$\mathcal{B} = \{B_1, B_2, \ldots, B_K\}$ {a set of K resource blocks}

$\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ {a set of M cellular users}

$\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ {a set of N D2D users}

$\mathcal{P} = \{p_1, p_2, \ldots, p_L\}$ {a set of available power levels}

Output:

$\mathcal{P}_b^* = \{p_{D_1}^*, p_{D_2}^*, \Lambda\ , p_{D_N}^*\}$ {optimal power levels of all DU pairs}

Function:

$Q_t^r(s,a)$ {Q table for the rth RB in the tth episode under state s and action a}

Initialize:

for the rth RB, $r \in \{1,2, ..., K\}$

initialize $Q_t^r(s,a)$, $a \in \mathcal{A}$

Learning:

for:

select the rth RB, $r \in \{1,2, ..., K\}$

for:

select action $a \in \mathcal{A}$ according to the ε greedy strategy

execute a and

calculate the reward

update $Q_t^r(s,a)$ according to Eq. (11)

end for

end for

4.3 Distributed Q-learning Algorithm for D2D Power Control

In the centralized Q-learning algorithm, all DUs on the same RB update a common Q table with the size of $1 \times L^n$, where L is the number of available power levels, and n is the number of DU pairs on the RB. As the number of D2D pairs grows, the complexity increases exponentially and it is intractable to compute the value of the Q table. To solve this problem, we propose a distributed Q-learning algorithm, in which each agent maintains and updates its own Q table with the size of $1 \times L$. This can dramatically reduce the time and space complexity for calculating the value of the Q table.

The definitions of the agent, state, action and reward in the distributed Q-learning algorithm are the same as those in the centralized Q-learning algorithm. It is worth noting that in [6], the throughput of a DU pair is used as the reward, excluding the throughput of CUs and other DU pairs sharing the same RB. However, according to [9], the optimal Q value can be obtained only if the reward function remains the same as that in the centralized Q-learning algorithm. Thus, we keep the same reward function as that in Eq. (9). The update function for the distributed Q-learning algorithm [11] is given by

$$Q_{t+1}^{j}(s,a) = \max\{Q_t^j(s,a), r_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q_t^j(s',a')\}, \tag{12}$$

where $Q_t^j(s,a)$ denotes the Q value for D_j in the tth episode.

An important problem with the distributed Q-learning algorithm is the coordination between different agents during the learning process. Since each agent selects its own action independently and the change of the action will in turn affect the throughput of other agents, it is not possible for all the agents to select their actions simultaneously. To deal with this problem, a simple heuristic method is used for scheduling the update of the Q tables. Specifically, for the rth RB, the DU pairs update their Q tables by turns in one episode. After all the DU pairs on the rth RB have updated their Q tables, the next episode starts with the first agent until all the Q tables converge to the same optimal value. It can be proved that the optimal Q values for the distributed Q-learning algorithm and the centralized Q-learning algorithm are equal [9].

The pseudo codes of the distributed Q-learning algorithm for D2D power control is given in Algorithm 2.

Algorithm 2 Distributed Q-learning algorithm for D2D power control

Input:

$\mathcal{B} = \{B_1, B_2, \ldots, B_K\}$ {a set of K resource blocks}

$\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ {a set of M cellular users}

$\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ {a set of N D2D users}

$\mathcal{P} = \{p_1, p_2, \ldots, p_L\}$ {a set of available power levels}

Output:

$\mathcal{P}_b^* = \{p_{D_1}^*, p_{D_2}^*, \Lambda\ , p_{D_N}^*\}$ {optimal power levels of all DU pairs}

Function:

$Q_t^j(s,a)$ {Q table for D_j in the tth episode under state s and action a}

Initialize:

for D_j , $j \in \{1,2, ..., N\}$

initialize $Q_t^j(s,a) = 0$, $a \in \mathcal{A}$

Learning:

for:

select the rth RB, $r \in \{1, 2,..., K\}$

for:

select D_j, for all the DUs on the rth RB.

for:

select action $a \in \mathcal{A}$ according to the ε greedy strategy

execute a and

calculate the reward r_{t+1}

update $Q_t^j(s,a)$ according to Eq. (12)

end for

end for

end for

5 Simulation Results

In this section, we evaluate the performance of the proposed centralized and distributed Q-learning algorithms through simulation results. The simulation experiments were conducted on a simulator developed using python. We consider a single cell cellular system where the CUs and DU pairs are uniformly distributed. The parameters used in the simulation experiment are listed in Table 1.

In the performance evaluation, we compare the proposed centralized and distributed Q-learning algorithms with a random allocation algorithm. The random algorithm randomly selects a power level from $\mathcal{P}$ for each DU pair. Moreover, we use the system throughput and satisfaction ratio as the performance metrics. The system throughput is defined as the throughput of all CUs and DU pairs in the system. The satisfaction ratio is defined as the number of CUs whose SINR requirements are satisfied over the total number of CUs in the system.

Table 1. Simulation parameters

Parameter	Value
M	20
N	10–100
K	20
L	5
Cell radius	500 m
p_1, p_2, p_3, p_4, p_5	{1, 6.5, 12, 17.5, 23}dBm
p_c	24 dBm
Noise power	−116 dBm/Hz
Resource block bandwidth	180 kHz
Gain model between user and BS	15.3 + 37.6lg(d(km)) dB
Gain model between two users	128 + 40lg(d(km)) dB
Learning rate α	0.9
Discount factor γ	0.9
τ_0	6 dB

Figure 3 compares the convergence of the optimal Q values with the centralized Q-learning algorithm and the distributed Q-learning algorithm under $M = 1$, $N = 5$, respectively. It can be observed that both the centralized algorithm and the distributed algorithm converge to the same Q value, which conforms to the conclusion in [9]. Meanwhile, it takes more episodes for the centralized algorithm to converge than for the distributed algorithm. This is because the centralized algorithm uses a larger Q table than the distributed algorithm.

According to Fig. 3, the centralized Q-learning algorithm and the distributed Q-learning algorithm converge to the same optimal Q value. Thus, the optimal power levels of the DU pairs with the two algorithms are equal, which results in the equal system throughput and satisfaction ratios. Therefore, we will only show the simulation results with the distributed Q-learning algorithm in the following figures.

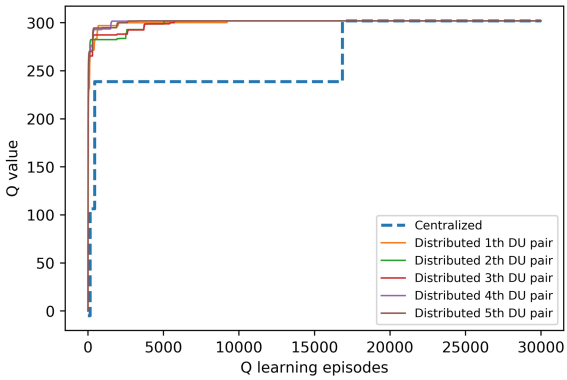

Fig. 3. Comparison of the convergence with the two algorithms ($M = 1$, $N = 5$)

Figure 4 shows the system throughput with the distributed Q-learning algorithm and a random algorithm, respectively. It is observed that the system throughput increases as the number of DU pairs increases. On the other hand, the system throughput with the distributed algorithm is larger than that with the random algorithm.

Figure 5 shows the satisfaction ratios of CUs with the distributed Q-learning algorithm and the random algorithm, respectively. It can be observed that the satisfaction ratio with the distributed algorithm is larger than that with the random algorithm. Moreover, as the number of DU pairs increases, the distributed algorithm can keep relatively stable at a higher value (>=0.9), while the satisfaction ratio with the random algorithm decreases dramatically.

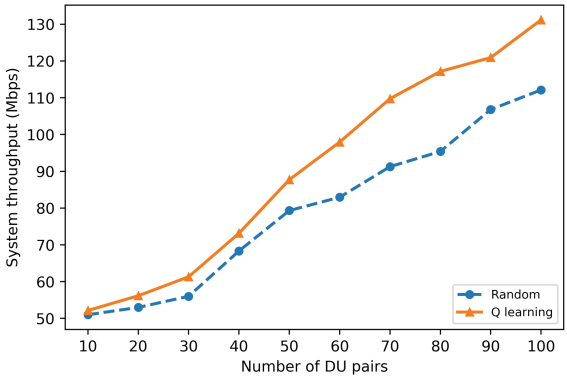

Fig. 4. System throughput with the distributed Q-learning algorithm

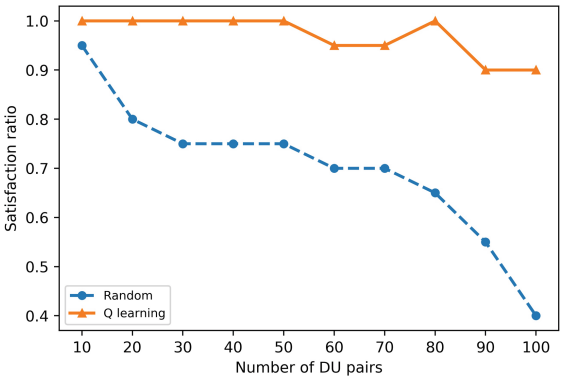

Fig. 5. Satisfaction ratio with the distributed Q-learning algorithm

6 Conclusions

In this paper, we considered the power control problem in D2D communication underlaying a cellular network and proposed two MARL-based algorithms for performing power control of D2D users: centralized Q-learning algorithm and distributed Q-learning algorithm. In the centralized algorithm, all D2D user (DU) pairs sharing the same RB use a common Q table in the learning process, while in the distributed algorithm each DU pair maintains its own Q table. Simulation results show that both the centralized algorithm and the distributed algorithm can converge to the same Q value, and the distributed algorithm can converge faster than the centralized algorithm. Moreover, both the proposed Q-learning algorithms outperform the random power control algorithm in terms of the system throughput and satisfaction ratio. In future work, we will explore to use the ML approach in joint spectrum allocation and power control for D2D communication underlaying cellular networks.

References

1. Asadi, A., Wang, Q., Mancuso, V.: A survey on device-to-device communication in cellular networks. IEEE Commun. Surv. Tutorials **16**(4), 1801–1819 (2014)
2. Lee, N., Lin, X., Andrews, J.G., Heath, R.W.: Power control for D2D underlaid cellular networks: modeling, algorithms, and analysis. IEEE J. Sel. Areas Commun. **33**(1), 1–13 (2015)
3. Ren, Y., Liu, F., Liu, Z., Wang, C., Ji, Y.: Power control in D2D-based vehicular communication networks. IEEE Trans. Veh. Technol. **64**(12), 5547–5562 (2015)
4. da Silva, J., Fodor, G.: A binary power control scheme for D2D communications. IEEE Wireless Commun. Lett. **4**(6), 669–672 (2015)
5. Sun, P., Shin, K.G., Zhang, H., He, L.: Transmit power control for D2D-underlaid cellular networks based on statistical features. IEEE Trans. Veh. Technol. **66**(5), 4110–4119 (2017)

6. Nie, S., Fan, Z., Zhao, M., Gu, X., Zhang, L.: Q-learning based power control algorithm for D2D communication. In: Proceedings of 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2016), Valencia, Spain, pp. 1–6 (2016)
7. Khan, M.I., Alam, M.M., Le Moullec, Y., Yaacoub, E.: Cooperative reinforcement learning for adaptive power allocation in device-to-device communication. In: Proceedings of 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore (2018)
8. Luo, Y., Shi, Z., Zhou, X., Liu, Q., Yi, Q.: Dynamic resource allocations based on Q-learning for D2D communication in cellular networks. In: Proceedings of 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, pp. 385–388 (2014)
9. Lauer, M., Riedmiller, M.: An algorithm for distributed reinforcement learning in cooperative multi-gent aystems. In: Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA, pp. 535–542 (2000)
10. Jiang, T., Zhao, Q., David, G., Burr, A.G., Clarke, T.: Single-state Q-learning for self-organized radio resource management in dual-hop 5G high capacity density networks. Trans. Emerg. Telecommun. Technol. **27**, 1628–1640 (2016). https://doi.org/10.1002/ett.3019
11. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. IEEE Trans. Neural Networks **9**(5), 1054 (1998)