Computer Science and Information Systems 18(3):979–999 https://doi.org/10.2298/CSIS200710055Y

Deep Reinforcement Learning for Resource Allocation with Network Slicing in Cognitive Radio Network*

Siyu Yuan[1,2], Yong Zhang[1,2**], Wenbo Qie[1], Tengteng Ma[1,2], and Sisi Li[1]

[1] School of Electronic Engineering,
Beijing University of Posts and Telecommunication
100876, Beijing, China
{yuanisyu, yongzhang, qwb, mtt, ssl123}@bupt.edu.cn
[2] Beijing Key Laboratory of Work Safety Intelligent Monitoring,
Beijing University of Posts and Telecommunications
100876, Beijing, China

Abstract. With the development of wireless communication technology, the requirement for data rate is growing rapidly. Mobile communication system faces the problem of shortage of spectrum resources. Cognitive radio technology allows secondary users to use the frequencies authorized to the primary user with the permission of the primary user, which can effectively improve the utilization of spectrum resources. In this article, we establish a cognitive network model based on underlay model and propose a cognitive network resource allocation algorithm based on DDQN (Double Deep Q Network). The algorithm jointly optimizes the spectrum efficiency of the cognitive network and QoE (Quality of Experience) of cognitive users through channel selection and power control of the cognitive users. Simulation results show that proposed algorithm can effectively improve the spectral efficiency and QoE. Compared with Q-learning and DQN, this algorithm can converge faster and obtain higher spectral efficiency and QoE. The algorithm shows a more stable and efficient performance.

Keywords: cognitive radio network, network slicing, resource allocation, deep reinforcement learning.

1. Introduction

With the development of wireless communication technology, wireless communication services around the world have shown a trend of rapid movement, huge capacity and mechanism intelligence. The fifth-generation cellular network is the key technology of the current wireless communication technology. The deployment of 5G networks will promote the rapid development of IoT (Internet of Things) and cloud computing services such as 4K video, VR (Virtual Reality), AR (Augmented Reality), driverless cars, intelligent power grids, and telemedicine [14]. 5G network has the characteristics of network virtualization and programmability, and uses a new technology called network slicing [4]. Network slicing is an on-demand networking model that allows operators to separate multiple virtual networks on a unified infrastructure. Each network slice is logically isolated

* This paper is an extented version of [27] which is published in International Conference on Human-Centered Computing 2019.

** Corresponding author

from the wireless access network to the core network to adapt to various types of applications. The 5G network supports three general service scenarios: eMBB (Enhanced Mobile Broadband), URLLC (Ultra-reliable and Low Latency Communication) and mMTC (Massive Machine-type Communications). eMBB refers to the further improvement of user experience and other performance based on existing mobile broadband business scenarios. The intuitive feeling is that the transmission rate has been greatly improved, which is mainly used for 4K video and large file download. URLLC is characterized by high reliability and low latency, and is mainly used for unmanned driving and remote surgery. In order to provide better performance and cost-effective services, network slicing has a lot of research space in terms of resource management. By using resource management algorithms, the wireless network can effectively increase the total transmission rate of the wireless access network [17], spectrum efficiency [11], and user-perceived QoE [8]. mMTC scenario is mainly used for large-scale IoT services.

At present, people's demand for data rate is higher and higher, and the demand for spectrum resources is also increasing. However, spectrum resources are very scarce. According to current spectrum policies, most of the available spectrum has been allocated or licensed to wireless service providers. In order to solve the problem of spectrum scarcity, cognitive radio technology has become the key to solving this problem [12]. Cognitive radio technology monitors the working conditions of authorized users by sensing the spectrum environment, and dynamically schedules the available idle spectrum under the premise of causing interference within a certain range to the authorized users, thereby improving spectrum utilization. In a cognitive radio network, according to the different ways that cognitive users access idle licensed spectrum, the sharing of licensed spectrum can be divided into two models (overlay and underlay). In the overlay mode, cognitive users can only use authorized spectrum when authorized users are not communicating. Underlay mode allows cognitive users to use the spectrum to which authorized users belong to perform data transmission with authorized users at the same time. Cognitive users will cause certain interference to authorized users, but the interference should be guaranteed within a certain range. In order to restrict the interference caused by cognitive users, the interference temperature constraint plays a key role in the allocation of cognitive radio resources. Interference temperature is a concept defined by the FCC (Federal Communications Commission) in order to improve spectrum utilization efficiency and study the application of cognitive radio [22], which is used to quantify the communication interference of cognitive users.

Currently in China, the 230 MHz frequency band is used for the construction of electric power wireless private networks. It is a dedicated spectrum resource specifically allocated to industries such as power, water power, and geology. Many frequency bands in electric power wireless private networks are licensed frequency bands. Private network users cannot use the licensed frequency bands of other private networks, which makes the 230MHz frequency band have weak transmission capabilities and low spectrum utilization [2]. With the development of wireless communication technology, the current power wireless private network based on the LTE system has begun to evolve to 5G, and the application of multi-slice services needs to be carried out in the spectrum awareness environment. Applying cognitive radio technology to 5G networks can effectively solve the problem of spectrum scarcity, improve spectrum utilization, and provide effective help for the construction of 5G-based power wireless private network systems.

Reinforcement learning algorithms are used to solve decision-making problems and obtain optimal strategies through continuous interaction with the environment. The most widely used reinforcement learning algorithm is Q-Learning [21]. In order to solve complex control problems, deep reinforcement learning combines reinforcement learning with deep learning to learn control strategies from high-dimensional raw data. The basic idea of deep reinforcement learning is to use deep learning to automatically learn abstract features of large-scale input data, and then use reinforcement learning based on deep learning feature representation to learn and optimize problem solving strategies. The DeepMind team first proposed DQN (Deep Q Network) in 2013 for playing Atari video games and obtaining high scores [13]. Later, DQN appeared many variants, such as DDQN (Double Deep Q Network) [19], D3QN (Dueling Double Deep Q Network) [20] and DQN with prioritized experience replay [16].Currently, reinforcement learning has been widely used in the field of wireless communication resource allocation [23,24,26,1].

In this article, we apply a DDQN algorithm and propose a deep reinforcement learning framework called CNDDQN for cognitive radio networks. This deep reinforcement learning framework is used to solve the resource allocation problem in cognitive radio networks with network slicing. Under the cognitive radio network underlay model, this framework jointly optimizes the overall spectrum efficiency of the cognitive network and the QoE of the secondary users by managing the channel selection and power allocation of the secondary users. This framework learns the optimal resource allocation strategy by establishing a mapping between known primary user channel selection and power allocation strategies and secondary user channel selection and power allocation strategies. We first introduce a cognitive radio network model combined with network slicing. Secondly, we introduce the basic concepts of reinforcement learning algorithms, Q-Learning and DDQN algorithms. Subsequently, we show the details of the CNDDQN algorithm. Finally, we conduct simulation experiments on the CNDDQN algorithm to verify the stability and effectiveness of the CNDDQN algorithm.

The key contributions of this article are as follows:

1) This paper proposes a cognitive radio model in the 5G network slicing scenario, which provides effective help for the construction of 5G-based electric power wireless private network system.

2) The resource allocation algorithm proposed in this paper considers user QoE and jointly optimizes the network spectrum efficiency and user QoE to ensure the user experience.

3) This paper proposes a resource allocation algorithm based on DDQN to solve the overestimation problem of DQN algorithm.

The remaining chapters of this paper are arranged as follows. Section 2 introduces some research work related to this article. Section 3 introduces the system model of the cognitive radio network and the formulation process of the resource allocation problem. Section 4 introduces the proposed deep reinforcement learning algorithm (CNDDQN). The simulation results and analysis are in Section 5. We summarize this article in Section 6.

2. Related Work

Resource allocation in cognitive radio networks has been widely studied, [18,6] summarizes these existing studies. The main optimization objectives of resource allocation in cognitive radio network include maximizing throughput, spectrum efficiency and energy efficiency, minimizing interference and ensuring the quality of service of users. [7] proposes a distributed user association and resource allocation algorithm based on matching theory to maximize the total throughput of primary and secondary users. [9] proposes a method based on deep reinforcement learning for cognitive uplink users of cellular networks, and deployed some sensors to help secondary users collect signal strength information at different locations in the wireless environment. Therefore, the secondary user can realize spectrum sharing with the primary user without knowing the power allocation strategy of the primary user. However, [9] does not consider the channel selection strategy of secondary users.

As the key technology of 5G network, network slicing technology is considered in many kinds of resource allocation scenarios. There are some researches on the application of network slicing technology in cognitive radio network resource allocation scenarios [10,3]. In [15], the network slicing technology is classified into spectrum level, infrastructure level and network level network slicing. In [10], the allocation of wireless slicing resources among multiple users is modeled as a bankruptcy game, which realizes the fairness of allocation. [3] proposes a multi-time-scale cognitive radio network slicing resource allocation model. The resource allocation model can be decomposed into inter-slice subchannels pre-assignment in large time period and intra-slice subchannels and power scheduling in same time slot. [3] formulates the inter-slice problem as an integer optimization problem and intra-slice problem as a mixed optimization problem with integer variables, and adopts Lyapunov optimization method with heuristic subchannel assignment procedure and a fast barrier-based power allocation procedure. The above papers use traditional optimization methods, such as game theory and Lyapunov optimization. These traditional optimization methods need to transform the optimization objectives into convex optimization problems to obtain the optimal solution, which has certain restrictions on the communication network scenarios. For example, the locations of users are fixed, and more users will bring higher algorithm complexity and longer calculation time.

In order to solve the problem of resource allocation in complex communication network scenarios, we propose a reinforcement learning architecture to solve the problem of resource allocation optimization in communication networks. The existing reinforcement learning algorithms applied to resource allocation are mainly divided into distributed multi-agent reinforcement learning algorithm [5] and centralized single agent reinforcement learning algorithm [27,25]. The centralized algorithm needs global information, has better utility value, and can balance the whole network users. Distributed algorithm only needs to know local information, so it has less communication cost. [27] proposes a centralized reinforcement learning algorithm based on DQN, which uses underlay access mode to maximize the spectrum efficiency of secondary users under the interference temperature limit acceptable for the primary user. But the network model of [27] does not consider network slicing. [5] proposes a distributed reinforcement learning algorithm based on Q-Learning and SARSA. The secondary users are organized into a random dynamic team in a decentralized and cooperative way, which speeds up the convergence speed

of the algorithm, improves the network capacity, and obtains the optimal energy-saving resource allocation strategy. But [5] only considers a single kind of service slice (high rate service slice) in the network model, and due to the use of table-based Q-learning and SARSA algorithm, the state space becomes discrete space, and there is a certain quantization error when segmenting the state space. [25] proposes a graph convolutional network-based reinforcement learning algorithm based on DQN. Secondary users are formed into a graph, and the information features are extracted by graph convolution, and then the DQN algorithm is used for policy learning to maximize the data rate of secondary users on the premise of the quality of service of users. In this paper, we propose a centralized reinforcement learning algorithm based on DDQN, and use DDQN algorithm to solve the problem of over estimation of DQN algorithm, so as to speed up the convergence speed and stability of the algorithm. In addition, in the network scenario, we consider the scenario where multiple service slices are combined with cognitive radio networks, and we consider rate-sensitive eMBB service slices and delay-sensitive URLLC service slices. In terms of optimization goals, if only the overall spectral efficiency of the cognitive network is optimized, this may sacrifice the user experience of some users. Therefore, we jointly optimize the spectral efficiency of the cognitive network and the user-perceived QoE of each user.

3. System Model and Problem Formulation

In this section, the system model and problem formulation are described.

3.1. System Model

This article considers a downlink OFDMA (Orthogonal Frequency Division Multiple Access) cellular cognitive network, as shown in Fig. 1. This network model has one PBS (Primary Base Station) and one CBS (Cognitive Base Station). The PUs (Primary Users) are associated with PBS, and the SUs (Secondary Users) are associated with the CBS. The PBS and CBS share the same spectrum resource. The SU adopts the underlay access model, and within the interference acceptance range of the PU, the SU is allowed to use the licensed frequency band resources of the PU. In Fig. 1, the black line indicates the communication between PU and PBS, the blue line indicates the communication between SU and CBS. The red line indicates the interference from the CBS to the PU, which should be controlled within a certain range.

3.2. Problem Formulation

In this scenario, secondary users are divided into two categories. The two types of secondary users have different service types and communication requirements. One type of secondary users are high-rate users, and the other type of secondary users are low-latency users. For these two types of secondary users, by using network slicing technology, high-rate users are associated with eMBB slices, and low-latency users are associated with URLLC slices. The set of secondary users associated with the eMBB slice is $SU^{eMBB} = \{1, 2, ..., N_{su}^{eMBB}\}$, and the set of secondary users associated with the URLLC slice is $SU^{URLLC} = \{1, 2, ..., N_{su}^{URLLC}\}$. Therefore, the set of all secondary

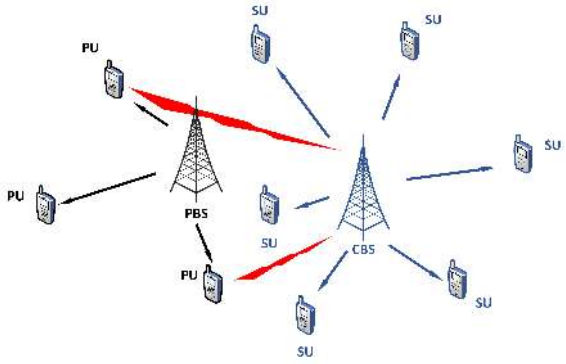

Fig. 1. Cognitive Radio Network Model

users is $SU = \{1, 2, ..., N_{su}\}$, where $N_{su} = N_{su}^{eMBB} + N_{su}^{URRLC}$ is the total number of secondary users. The total primary user set is $PU = \{1, 2, ..., N_{pu}\}$, where N_{pu} is the total number of primary users.

There are k channels for users to use. The channel set is $C = \{1, 2, ..., k\}$ and the bandwidth of each channel is B. Therefore, the total network bandwidth is $W = k * B$. Assuming that each primary user can occupy multiple channels at the same time, the primary user-channel association matrix is $PCA = \{a_{n,k}^{pc}\}_{N_{pu}*k}$. If PU n occupies the channel k, then $a_{n,k}^{pc} = 1$, otherwise $a_{n,k}^{pc} = 0$. Each secondary user can only occupy one channel, and the secondary user-channel association matrix is $SCA = \{a_{n,k}^{sc}\}_{N_{su}*k}$. If the secondary user n occupies the channel k, then $a_{n,k}^{sc} = 1$, otherwise $a_{n,k}^{sc} = 0$.

The channel gain matrix of each primary user and PBS is $PPG = \{g_n^{pP}\}_{N_{pu}}$, and the channel gain matrix of each primary user and CBS is $PCG = \{g_n^{pC}\}_{N_{pu}}$. The channel gain matrix of each secondary user and PBS is $SPG = \{g_n^{sP}\}_{N_{su}}$, and the channel gain matrix of each secondary user and CBS is $SCG = \{g_n^{sC}\}_{N_{su}}$.

Assume that the maximum transmission power of the PBS and CBS are $P_{\max}^{PBS}$ and $P_{\max}^{CBS}$ correspondingly. $P_{n,k}^{pu}$ indicates the transmission power of the primary user n on the channel k, and $P_{n,k}^{su}$ indicates the transmission power of the secondary user n on the channel k.

According to the definition of the signal-to-interference and noise ratio, (1)(2) are the expressions of the signal-to-interference ratio of the PU and SU.

$$\delta^{pu} = \frac{\sum\limits_{k \in C} a_{n,k}^{pc} \cdot g_n^{pu,PBS} \cdot P_{n,k}^{pu}}{\sum\limits_{a \in PU, a \neq n} \sum\limits_{k \in C} a_{n,k}^{pc} \cdot a_{a,k}^{pc} \cdot g_a^{pP} \cdot P_{a,k}^{pu} + \sum\limits_{a \in SU} \sum\limits_{k \in C} a_{n,k}^{pc} \cdot a_{a,k}^{sc} \cdot g_a^{pC} \cdot P_{a,k}^{su} + \sigma^2}. \tag{1}$$

$$\delta^{su} = \frac{\sum\limits_{k \in C} a_{n,k}^{sc} \cdot g_n^{sC} \cdot P_{n,k}^{su}}{\sum\limits_{a \in PU} \sum\limits_{k \in C} a_{n,k}^{sc} \cdot a_{a,k}^{pc} \cdot g_a^{sP} \cdot P_{a,k}^{pu} + \sum\limits_{a \in SU, a \neq n} \sum\limits_{k \in C} a_{n,k}^{sc} \cdot a_{a,k}^{sc} \cdot g_a^{sC} \cdot P_{a,k}^{su} + \sigma^2}. \tag{2}$$

According to the Shannon channel formula $R = B \cdot \log(1+\delta)$, the transmission rate of the primary user R_n^{pu} and secondary user R_n^{su} can be calculated. Therefore, the total transmission rate of the cognitive network is $R_{cn} = \sum_{n \in SU} R_n^{su}$. (3) is the total spectrum efficiency of the cognitive network.

$$\eta_{cn} = \frac{R_{cn}}{W} = \frac{\sum_{n \in SU} B \cdot \log(1+\delta_n^{su})}{k * B} = \frac{1}{k} \cdot \sum_{n \in SU} \log(1+\delta_n^{su}) \,. \tag{3}$$

The user's QoE is mainly reflected by the user's communication needs. The user's QoE is defined as the ratio of the number of packets meeting the communication requirements to the total number of packets. The communication demand of eMBB slice users is that the transmission rate is higher than a certain threshold, and the communication demand of URLLC slice users is that the transmission delay is lower than a certain threshold.

The transmission rate of the data packet is expressed by the user's transmission rate, and the transmission delay of the data packet is composed as shown in Fig. 2.

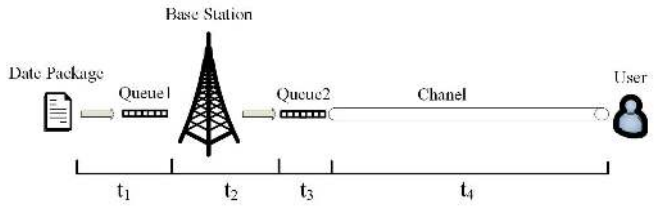

Fig. 2. Transmission Delay of Data Packet

The transmission delay of data packets is mainly composed of the queue delay (t_1) when entering the base station, the operation delay of the channel allocated at the base station (t_2), the queue delay of entering the channel (t_3), and the transmission delay of transmitting in the channel (t_4). In order to simplify the transmission delay model of the data packet, t_1 and t_2 belong to the transmission delay of the base station, the value of t_4 is very small, they are not considered in this paper. Therefore, the transmission delay of the data packet is the queue delay of the data packet entering the channel t_3. We use the M/M/1 queue model to calculate the queue delay. According to the average waiting time formula of the M/M/1 queue model $W_s = 1/(\mu - \lambda)$, where μ is the service rate and λ is the arrival rate. We can get the queue delay $t_3 = 1/(r_{package} - \lambda)$, where λ is the arrival rate of each data packet, $r_{package} = R_n/L$ is the transmission rate of each data packet, R_n is the transmission rate of the user, L is the packet length of the data packet. We assume that the packet length is normally distributed. Therefore, (4) is the transmission delay of the data packet.

$$t = \frac{1}{R_n/L - \lambda} \,. \tag{4}$$

Let $t_{\max}$ and $R_{\min}$ be the threshold for the data packet transmission delay and transmission rate to meet the communication requirements. The expression that meets the communication requirements is shown in (5). The user's QoE is equal to the ratio of the number of packets that meet the inequality requirements to the total number of packets.

$$\begin{cases} R_n \geq R_{\min} & \text{for eMBB users} \\ t = \frac{1}{R_n/L - \lambda} \leq t_{\max} & \text{for URLLC users} \end{cases}. \tag{5}$$

In order to balance the spectral efficiency and the user's QoE, we set the attention coefficient $\alpha \in [0, 1]$ between the spectral efficiency and the user's QoE. $\alpha = 1$ means that the optimization goal is only to maximize the system spectral efficiency, and $\alpha = 0$ means that the optimization goal is only to maximize the user QoE. Therefore, our optimization goal is (6).

$$\max[\alpha\eta_{cn} + (1-\alpha)QoE]\,. \tag{6}$$

The interference temperature is defined as the ratio of the interference power to the corresponding bandwidth $IT = \frac{P_{\text{interference}}}{k_{cons}W}$, where $P_{\text{interference}}$ is the power of the interference noise in the channel, k_{cons} is the Boltzmann constant, and W is the total bandwidth of the cognitive network. Therefore, the total interference temperature of the cognitive network is (7).

$$IT = \frac{\sum\limits_{a\in SU}\sum\limits_{k\in C} a_{a,k}^{sc} \cdot g_a^{pC} \cdot P_{a,k}^{su}}{k_{cons} \cdot W}\,. \tag{7}$$

Let the maximum interference temperature caused by the cognitive network acceptable to the PU be $IT^{\max}$.

Constraint C1 indicates that each secondary user can only be associated with one channel. Constraint C2 is the maximum total power constraint of the cognitive base station. Constraint C3 is the main user's interference temperature constraint on the cognitive network.

Therefore, the optimization problem can be expressed as (8- 11).

$$\max[\alpha\eta_{cn} + (1-\alpha)QoE]\,. \tag{8}$$

$$s.t.C1 : \sum_{k\in C} a_{n,k}^{sc} \leq 1, \forall n \in SU\,. \tag{9}$$

$$C2 : 0 \leq \sum_{n\in SU} P_{n,k}^{su} \leq P_{\max}^{CBS}\,. \tag{10}$$

$$C3 : \frac{\sum\limits_{a\in SU}\sum\limits_{k\in C} a_{a,k}^{sc} \cdot g_a^{pC} \cdot P_{a,k}^{su}}{k_{cons} \cdot W} \leq IT^{\max}\,. \tag{11}$$

Due to the nonlinear constraints of continuous variables (such as $P_{a,k}^{su}$) and binary variables (such as $sca_{a,k}$), the optimization problem is a non-convex problem. Using deep reinforcement learning to solve such non-convex problems is a common method. Therefore, we propose a deep reinforcement learning algorithm to solve this optimization problem.

4. Deep Reinforcement Learning for Optimization Problem

4.1. Reinforcement Learning

Reinforcement learning is a common method for solving decision problems. Reinforcement learning has two basic elements (state and action). Performing a certain action in a certain state is a strategy. Agents need to obtain a good strategy in continuous exploration and learning. If the state is regarded as an attribute and the action is regarded as a mark, reinforcement learning is similar to supervised learning. They are all trying to find a mapping from known attribute/state to the mark/action. In this way, the strategy in reinforcement learning is equivalent to the classifier and regressor in supervised learning. However, in practical problems, reinforcement learning does not have supervised learning as labeled information. Usually results are obtained after trying actions, so reinforcement learning is to continuously adjust the previous strategy through the feedback of the result information, so the algorithm can learn what kind of action to choose in which state to get the best result.

Reinforcement learning is usually described using MDP (Markov Decision Process). The agent is in an environment, and each state is the agent's perception of the current environment. The agent can only affect the environment through actions. When the agent performs an action, the environment will be transferred to another state with a certain probability. At the same time, the environment will feedback a reward to the agent according to the potential reward function. This process is shown in Fig. 3.

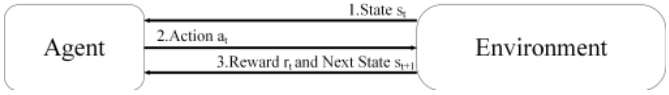

Fig. 3. Basic Process of Reinforcement Learning

Then, we define two value functions—state value function and action value function. The state value function $V(s)$ is defined as the expectation of the long-term reward that the state s can obtain at the moment t. The state value function represents the value of a state, regardless of which action the state chooses. It takes the current state as the starting point to make a weighted sum of all possible actions, the expression is $V_\pi(s) = E_\pi[R_t|S_t = s]$, where π is the strategy, and the expression is $\pi(a|s) = P[A_t = a|S_t = s]$. The action value function $G(s, a)$ is defined as the long-term reward that can be obtained by selecting the action a in state s at the moment t. The action value function represents the value of an action in a certain state. It is the weighted sum of all possible long-term rewards for a given state and action, the expression is $G_\pi(s, a) = E_\pi[R_t|S_t = s, A_t = a]$.

Usually, a limited Markov decision process consists of a quadruple $M = (S, A, P, R)$. Where S represents the limited state set space, A represents the action set space, P represents the state transition probability matrix, and R represents the expected reward value. The Markov decision process relies on the Markov assumption that the probability of the next state S_{t+1} depends only on the current state S_t and action A_t, not on the previous state or action. In the Markov decision process, given a state $s \in S$ and an action $a \in A$, it will transition to the next state $s' \in S$ with a certain probability. $P^a_{ss'}$ is the state transition

probability, which means that starting from the state s and taking action a, we will reach the state s' with the probability of $P^a_{ss'}$, the expression is $P^a_{ss'} = P(S_{t+1}|S_t = s, A_t = a)$. $r^a_{ss'}$ is the expected reward, which means starting from the state s, taking action a, and transferring to the state s', the expression is $r^a_{ss'} = E(r_{t+1}|S_t = s, A_t = a, S_{t+1} = s')$.

Because reinforcement learning can be summarized as obtaining an optimal strategy by maximizing rewards. However, if it is only the maximum instantaneous reward, it will only select the action with the largest reward from the action space every time, which becomes the simplest greedy policy. In order to achieve the maximum current reward value including the future, the total reward from the current moment until the end state reaches the goal is maximized. Therefore, the cumulative discount reward function $R(t)$ is constructed with the expression as $R(t) = \sum_{k=0}^{n} \gamma^k r_{t+k+1}$, where $\gamma \in [0, 1]$ is the discount coefficient, which indicates the degree of influence of the current reward in the future. $\gamma = 0$ means that the learned strategy is short-sighted and only considers even rewards and $\gamma = 1$ means that the rewards at all times are equal. Combining the definition of the state value function and the cumulative discount reward function, we can obtain the Bellman equation form of the state value function, as shown in (12- 16).

$$V_\pi(s) = E_\pi[R_t|S_t = s] \tag{12}$$

$$= E_\pi(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots|S_t = s) \tag{13}$$

$$= E_\pi(r_{t+1}|S_t = s) + E_\pi(\gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|S_t = s) \tag{14}$$

$$= \sum_a \pi(s, a) \sum_{s'} P^a_{ss'}\{R^a_{ss'} + \gamma E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|S_{t+1} = s']\} \tag{15}$$

$$= \sum_a \pi(s, a) \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V_\pi(s')] \tag{16}$$

Combining the definition of the action value function and the cumulative discount reward function, we can obtain the Bellman equation form of the action value function through a similar derivation process, as shown in (17). (18) and (19) are Bellman optimality equations.

$$G_\pi(s, a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma \sum_{a'} G_\pi(s', a')]\,. \tag{17}$$

$$V_*(s) = E[R_t + \gamma \max_\pi V(s')|S_t = s]\,. \tag{18}$$

$$Q_*(s) = E[R_t + \gamma \max_{a'} Q(s', a')|S_t = s, A_t = a]\,. \tag{19}$$

The most common reinforcement learning algorithm is the Q-Learning algorithm. By introducing Q-Table, the action value function is described. The update formula of Q-Learning is (20). By constantly updating, we can get an excellent Q-Table to make the decision-making process.

$$Q(s, a) = Q(s, a) + \alpha[R(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)]\,. \tag{20}$$

4.2. Deep Reinforcement Learning: from Q-Learning to DQN

Q-Learning is a classic algorithm for reinforcement learning, but there is a problem that Q-Learning uses a Q-Table to store Q values. This makes Q-Learning limited to the action space and the state space are very small, and generally in discrete situations. If there are many types of states and actions in the model, the size of the Q-Table will become very large, even larger than the memory of the computer, and it is also very time-consuming to search in a huge table for each update. However, more complex tasks that are closer to the actual situation often have a large state space and action space. For the field of processing high-dimensional data, deep learning has a good performance. Deep reinforcement learning combines reinforcement learning and deep learning, using neural networks instead of the original table to calculate the value function.

DQN is a representative algorithm for deep reinforcement learning. Based on the original Q-Learning used Q-tables, the Q value (action value function) is calculated using a neural network in DQN algorithm. In the decision-making process, DQN takes the state as the input of the neural network, calculates the Q value of each action through the neural network, and then selects the action according to the principle similar to Q-Learning. Fig. 4 compares the Q value calculation process of Q-Learning and DQN. The original Q value $Q(s, a)$ is replaced by a new form with neural network parameters $Q(s, a; \theta)$, where θ represents the parameters of the neural network.

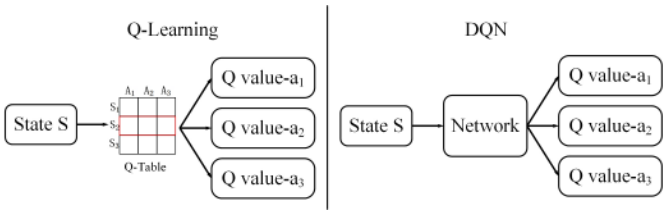

Fig. 4. Comparison of Calculation Process for Q of Q-learning and DQN

In order to reduce the problems caused by the correlation between data, DQN introduced two key technologies of experience replay and fixed target value network.

In supervised learning, each sample is independently identically distribution. However, the samples of reinforcement learning are obtained through the agent's continuous exploration, which makes the samples in reinforcement learning highly correlated and non-stationary, causing the training results difficult to converge. The experience replay technology is used to solve this problem. First put the collected samples into the sample pool, and then randomly select a sample from the sample pool for network training. Random sampling is used to remove the correlation between samples, making the samples independent of each other, thereby improving the stability and convergence of network training.

In the original Q-Learning, as described in (20), when we calculated the TD error, we obtained it by calculating the difference between the target Q and estimated Q. The calculation of the TD target is by using the Bellman equation. The TD target is the reward of the current action plus the highest Q value of the next state through attenuation. How-

ever, the same parameters are used when calculating the TD target and estimating the Q value. The correlation between the two makes the model prone to oscillation and divergence. In order to solve this problem, DQN builds an independent target Q network that is slower than the current Q network to calculate the TD target, which makes the possibility of oscillation and divergence during training reduced and more stable.

In Q-Learning, updating the Q value directly changes the value of the corresponding position in the table. In DQN, the Q value is updated by updating the parameters of the neural network. The update of the neural network parameters is based on the reverse transfer of the loss function. The loss function of DQN is defined as the square error form of target Q and estimated Q. (21) is the form of the loss function of DQN.

$$Loss^{DQN} = [r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)]^2 . \tag{21}$$

DQN still has the problem of overestimation. Overestimation means that the estimated value function is larger than the real value function, and its root is mainly in the maximization operation in Q-Learning. When calculating target Q, the maximum Q value in the next state is obtained. For real strategies and in a given state, the action that maximizes the Q value is not selected every time, because the general real strategies are random strategies, the selection of the maximum Q value of the action here will often result in the target value being higher than the real value. Double DQN solves the problem of overestimation on the basis of DQN. DDQN implements action selection and action evaluation with different value functions, and in DQN we have proposed two Q networks. Therefore, the step of DDQN calculating target Q can be split into two steps. In the first step, the action to maximize the Q value is obtained by estimated Q network. In the second step, the action value function corresponding to the action is obtained through the target Q network. Combining the two steps together, the loss function of DDQN can be obtained, as shown in (22).

$$Loss^{DDQN} = [r + \gamma Q(s', \operatorname*{argmax}_{\mathrm{a}'} Q(s', a'; \theta); \theta^-) - Q(s, a; \theta)]^2 . \tag{22}$$

Except for the change of the loss function, the main process of DDQN is the same as that of DQN. Fig. 5 is a flowchart of the operation of the DDQN algorithm.

4.3. CNDDQN (Cognitive Network Double Deep Q Network)

In this paper, we propose a Double DQN algorithm for solving the channel selection and power allocation problems in cognitive networks. In the cognitive network environment, the basic elements of reinforcement learning are set as follows.

The reinforcement learning agent is the overall cognitive network, and the DDQN algorithm runs in the CBS to manage the channel selection and power allocation of all cognitive users. The state of reinforcement learning is the SINR of the PU, which is recorded as $s_t = \{\delta_t^n\}_{1*N_{pu}}$.

The reinforcement learning action $a_t = \left\{\{a_t^{sc,n}\}_{1*N_{su}}, \{P_t^{su,n}\}_{1*N_{su}}\right\}_{1*2\cdot N_{su}}$ is the channel selection of the secondary user and the power allocation of the secondary user. Since the output of the DDQN algorithm is a discrete value, we divide the transmission power of cognitive users into 20 discrete power values on average. (23) is the action space of the transmission power of cognitive users.

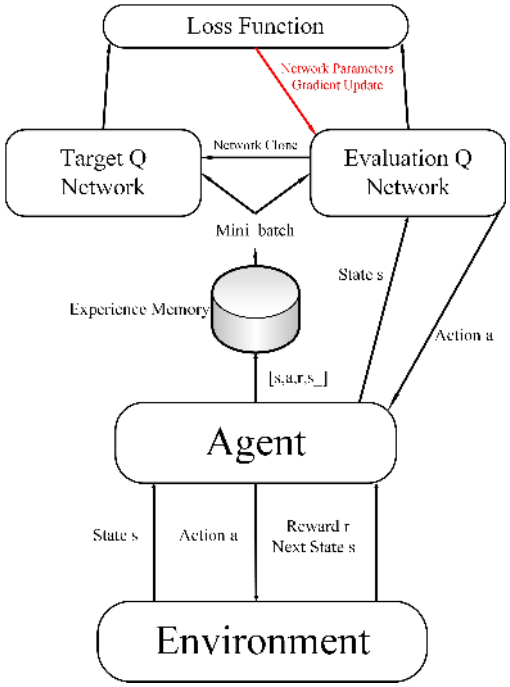

Fig. 5. Flowchart of DDQN Algorithm

$$P_{n,k}^{su} \in \{0, \frac{P_{\max}^{CBS}}{19}, \frac{2P_{\max}^{CBS}}{19}, ..., P_{\max}^{CBS}\}. \tag{23}$$

The reward function is modified on the basis of (8). The constraint condition C3 for interference temperature is added to the description of the reward function. If the constraint condition of the interference temperature is satisfied, a normal reward will be obtained. If the constraints of the interference temperature are not met, then only zero rewards can be obtained. The characteristics of the step function meet our expectations. We make the difference between the actual interference temperature and the interference temperature threshold to obtain the interference temperature threshold constraint function (24).

$$f(a_{a,k}^{sc}, P_{a,k}^{su}) = \varepsilon(IT^{\max} - \frac{\sum_{a \in SU} \sum_{k \in C} a_{a,k}^{sc} \cdot g_a^p \cdot P_{a,k}^{su}}{k \cdot W}). \tag{24}$$

$\varepsilon(x)$ is a step function, its characteristic is $\varepsilon = \begin{cases} 1 & x > 0 \\ 0.5 & x = 0 \\ 0 & x < 0 \end{cases}$.

The step function is discontinuous at 0, making it difficult during gradient descent. Therefore, we use the Sigmoid function to approximate the equivalent step function. Therefore, the interference temperature threshold constraint function is expressed as (25).

$$f(a_{a,k}^{sc}, P_{a,k}^{su}) = Sigmoid(IT^{\max} - \frac{\sum_{a\in SU}\sum_{k\in C} a_{a,k}^{sc} \cdot g_a^{pu,CBS} \cdot P_{a,k}^{su}}{k \cdot W}) . \quad (25)$$

The expression of Sigmoid function is $Sigmoid(x) = \frac{1}{1+e^{-x}}$. Combined with the interference temperature threshold constraint function, the reward function can be defined as shown in (26).

$$\text{Reward} = f(a_{a,k}^{sc}, P_{a,k}^{su}) * [\alpha\eta_{cn} + (1-\alpha)QoE]/, . \quad (26)$$

There are two neural networks in this algorithm, the training network and the target network. These two networks have the same structure, but the parameter updates are different. The neural network in this algorithm uses a simple fully connected neural network. The fully connected neural network contains 2 fully connected layers, the structure is shown in Fig. 6.

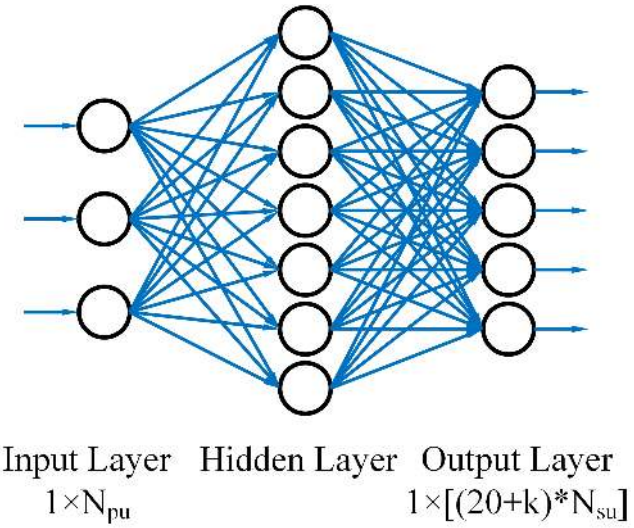

Fig. 6. Neural network structure (N_{pu}: total number of primary users, N_{su}: total number of secondary users, k: total number of channels)

The general steps of CNDDQN is summarized as shown in Algorithm 1.

5. Performance Evaluation

In this section, we first introduce the settings of cognitive network model parameters and DDQN hyperparameters. Then, we provide the simulation performance results.

Algorithm 1 The General Steps of CNDDQN

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ and target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
for each episode, M **do**
  Initialize network state s;
  for each step of an episode, T **do**
    CBS chooses an action $a_t = \arg\max_a Q(\phi(s_t), a; \theta)$ at state s_t with probability ε select a random action a_t;
    CBS completes channel and power allocation according to the selected action a_t;
    CBS calculates the reward r_t according to (26) through message passing;
    CBS observes the network state s_{t+1} through message passing;
    CBS stores transition (s_t, a_t, r_t, s_{t+1}) in D;
    CBS samples random minibatch of transitions (s_t, a_t, r_t, s_{t+1}) from D;
    $$y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \hat{Q}(s', \underset{a'}{\operatorname{argmax}}\, Q(s', a'; \theta); \theta^{-}) & \text{otherwise} \end{cases};$$
    CBS performs a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters θ;
    Every C steps, CBS resets $\hat{Q} = Q$;
    CBS sets $s_t = s_{t+1}$;
  end for
end for

5.1. Simulation Settings

In this model, there are 1 PBS and 1 CBS, and there are 10 PUs and 20 CUs. Among the 20 secondary users, 10 secondary users are associated with eMBB slices and 10 secondary users are associated with URLLC slices. The size of the model is 100m * 100m. The locations of base stations and users are fixed, and the distribution of base stations and users is shown in Fig. 7.

For the PBS and CBS, the maximum transmission power is $P_{\max}^{PBS} = P_{\max}^{CBS} = 46dBm$. For AWGN channels, the noise power is $\sigma^2 = 1e-7$. The standard deviation of shadow fading is set to 8dB. For model simplicity, channel fading only considers large-scale fading, the expression is $L(d) = 37 + 30\log(d)$, where d is the distance between the base station and the user. The network has a total of 20 channels, the bandwidth of each channel is 180kHz. For cognitive radio networks, the interference temperature acceptable to the primary user is 5dB. For the user's QoE, the rate threshold is set to 0.1Mbps and the delay threshold is 10ms. The hyperparameter settings for the DDQN algorithm are shown in Table 1.

5.2. Simulation Results

First, we show the performance of the DDQN algorithm at different learning rates. In the DDQN algorithm, the setting of the learning rate is very important. In the gradient descent process, the learning rate represents the step size of each update. Fig. 8 is a graph comparing the performance of the DDQN algorithm at different learning rates.

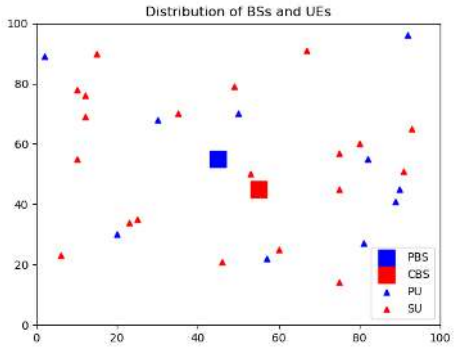

Fig. 7. Distribution of BSs and UEs

Table 1. Hyperparameters of DDQN Algorithm

Papameter	Value
Mini-batch size	32
Discount rate γ	0.995
Learning rate δ	0.005
ε-greedy	0.1
Activation function	Relu
Optimizer	Adam

The three curves in the figure are the images of the DDQN reward function value with the number of iterations when the learning rate is 0.05, 0.005 and 0.0005. When the learning rate value is small ($\delta = 0.0005$), the CNDDQN algorithm converges in about 1500 iterations, and the convergence speed is slow. As the learning rate increases, when $\delta = 0.005$,the convergence speed of the CNDDQN algorithm increases, and the CNDDQN algorithm converges in about 700 iterations. When the learning rate is too large ($\delta = 0.05$), the CNDDQN algorithm converges in about 400 iterations. But the final reward function convergence value is lower than the reward function convergence value when the learning rate is 0.005 and 0.0005. It can be seen that a low learning rate will lead to a slower convergence rate, requiring more iterations to achieve convergence. Too high a learning rate will cause CNDDQN to reach the final reward function convergence value lower than the normal learning rate convergence value. Therefore, the choice of learning rate should be moderate, too high and too low learning rate will make the performance of the algorithm decline. In the simulation of this paper, the learning rate $\delta = 0.005$ is an appropriate value.

We observe the curve of learning rate $\delta = 0.005$ in Fig. 8. We can find that the reward function value is low and unstable at the beginning of the iteration. As the training iteration progresses, the reward function continues to grow. After a certain number of iterations, the reward function completes convergence. This means that the CNDDQN algorithm has learned the optimal action strategy. After the CNDDQN algorithm converges, the jitter of the reward function is caused by ε-greedy in the CNDDQN algorithm.

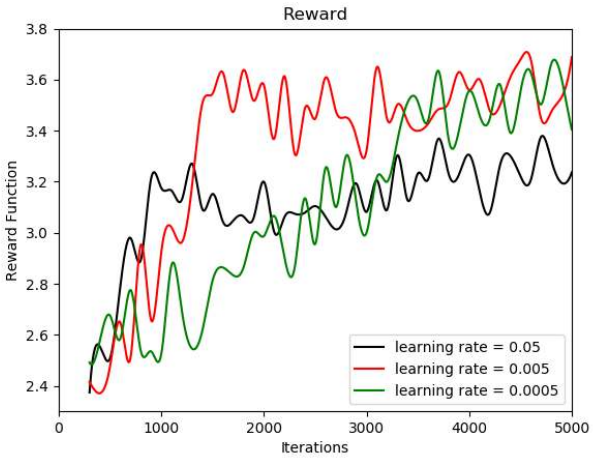

Fig. 8. Comparison of DDQN Performance with Different Learning Rates

The optimization objective in (6) is obtained by the linear combination of system spectral efficiency and QoE, and the attention coefficient is α. Fig. 9 shows the change curves of the average convergence value of user QoE and system spectral efficiency under different coefficients. As the attention factor increases, the CNDDQN algorithm's attention to the system spectral efficiency increases, the final average convergence value of the system spectral efficiency increases, and the final average convergence value of the user QoE decreases. It can be seen that a greater attention to system spectrum efficiency can result in a more superior system spectrum efficiency performance strategy, but at the same time it will cause a certain loss to the user's QoE performance. Similarly, in order to obtain superior user QoE performance strategies, a certain loss will be caused to the system spectrum efficiency.

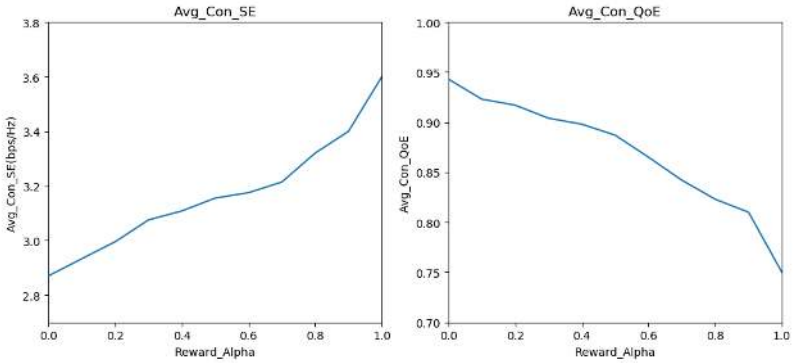

Fig. 9. Influence of Different Attention Coefficients on SE and User QoE

The comparative experiment algorithm selected in this paper is CNDQN algorithm and CNQ-learning algorithm, which is shown in Fig. 10. Compared with the CNDQN algorithm, the convergence speed of the CNDDQN algorithm has been significantly improved. The CNQ-learning algorithm uses a table to store Q values, so that not only the action space is discrete, but the state space is also discrete. This makes CNQ-learning's overall performance far from CNDDQN in complex cognitive radio scenarios.

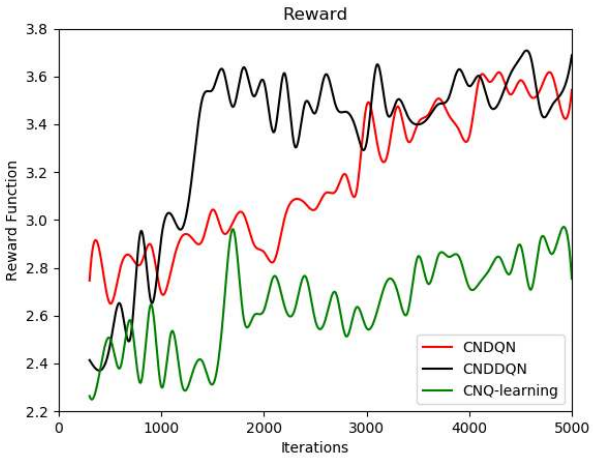

Fig. 10. Performance Comparison of Different Algorithms in Cognitive Radio Networks

6. Conclusion

In this article, we propose a resource allocation algorithm (CNDDQN) for cognitive radio with network slicing. This algorithm is used in cognitive radio scenarios in underlay mode. Under the interference acceptable to the primary user, the secondary user is allowed to access the frequency band authorized to the primary user. In order to quantify the interference caused by secondary users, we introduce the concept of interference temperature. In order to solve the proposed non-convex and NP-hard problem of resource allocation, we use a deep reinforcement learning algorithm (DDQN). The algorithm jointly optimizes the overall spectrum efficiency of the cognitive network and the QoE of the secondary user by managing the channel selection and power allocation of the secondary user. Through continuous iterative learning, the algorithm continuously updates the resource allocation strategy of the secondary users, and finally reaches the optimal resource allocation strategy. Simulation results show that compared with other reinforcement learning methods, the proposed CNDDQN can effectively achieve a near-optimal solution through a smaller number of iterations.

Acknowledgments. The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This work is supported by the National Natural Science Foundation of China under Grant No.61971057.

References

1. Chen, J., Chen, S., Wang, Q., Cao, B., Feng, G., Hu, J.: iraf: A deep reinforcement learning approach for collaborative mobile edge computing iot networks. IEEE Internet of Things Journal 6(4), 7011–7024 (2019)
2. Guo, D., Zhang, Y., Xu, G., Hyeongchun, P.: Spectrum aggregation scheme in a wireless broadband data transceiver system. international conference on robotics and automation 33(5) (2018)
3. Jiang, H., Wang, T., Wang, S.: Multi-scale hierarchical resource management for wireless network virtualization. IEEE Transactions on Cognitive Communications and Networking 4(4), 919–928 (2018)
4. Katsalis, K., Nikaein, N., Schiller, E., Ksentini, A., Braun, T.: Network slices toward 5g communications: Slicing the lte network. IEEE Communications Magazine 55(8), 146–154 (2017)
5. Kaur, A., Kumar, K.: Energy-efficient resource allocation in cognitive radio networks under cooperative multi-agent model-free reinforcement learning schemes. IEEE Transactions on Network and Service Management 17(3), 1337–1348 (2020)
6. Kumar, A., Kumar, K.: Multiple access schemes for cognitive radio networks: A survey. Physical Communication 38, 100953 (2020)
7. LeAnh, T., Tran, N.H., Saad, W., Le, L.B., Niyato, D., Ho, T.M., Hong, C.S.: Matching theory for distributed user association and resource allocation in cognitive femtocell networks. IEEE Transactions on Vehicular Technology 66(9), 8413–8428 (2017)
8. Li, R., Zhao, Z., Sun, Q., Chihlin, I., Yang, C., Chen, X., Zhao, M., Zhang, H.: Deep reinforcement learning for resource management in network slicing. IEEE Access 6, 74429–74441 (2018)
9. Li, X., Fang, J., Cheng, W., Duan, H., Chen, Z., Li, H.: Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach. IEEE Access 6, 25463–25473 (2018)
10. Liu, B., Tian, H.: A bankruptcy game-based resource allocation approach among virtual mobile operators. IEEE Communications Letters 17(7), 1420–1423 (2013)
11. Ma, T., Zhang, Y., Wang, F., Wang, D., Guo, D.: Slicing resource allocation for embb and urllc in 5g ran. Wireless Communications and Mobile Computing 2020, 1–11 (2020)
12. Mitola, J., Maguire, G.Q.: Cognitive radio: making software radios more personal. IEEE Personal Communications 6(4), 13–18 (1999)
13. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv: Learning (2013)
14. Pengpeng, L.I., Zheng, N., Kang, P., Tan, H., Fang, J.: Overview and inspiration of global 5g spectrum researches. Telecommunication Engineering (2017)
15. Richart, M., Baliosian, J., Serrat, J., Gorricho, J.L.: Resource slicing in virtual wireless networks: A survey. IEEE Transactions on Network and Service Management 13(3), 462–476 (2016)
16. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv: Learning (2015)
17. Tang, L., Tan, Q., Shi, Y., Wang, C., Chen, Q.: Adaptive virtual resource allocation in 5g network slicing using constrained markov decision process. IEEE Access 6, 61184–61195 (2018)
18. Tarek, D., Benslimane, A., Darwish, M., Kotb, A.M.: Survey on spectrum sharing/allocation for cognitive radio networks internet of things. Egyptian Informatics Journal (2020)

19. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning pp. 2094–2100 (2016)
20. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning pp. 1995–2003 (2016)
21. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (1992)
22. Yongjun, X., Xiaohui, Z.: Optimal power allocation for multiuser underlay cognitive radio networks under qos and interference temperature constraints. China Communications 10(10), 91–100 (2013)
23. Zhang, Y., Kang, C., Ma, T., Teng, Y., Guo, D.: Power allocation in multi-cell networks using deep reinforcement learning pp. 1–6 (2018)
24. Zhang, Y., Kang, C., Teng, Y., Li, S., Zheng, W., Fang, J.: Deep reinforcement learning framework for joint resource allocation in heterogeneous networks pp. 1–6 (2019)
25. Zhao, D., Qin, H., Song, B., Han, B., Du, X., Guizani, M.: A graph convolutional network-based deep reinforcement learning approach for resource allocation in a cognitive radio network. Sensors 20(18), 5216 (2020)
26. Zhao, N., Liang, Y., Niyato, D., Pei, Y., Jiang, Y.: Deep reinforcement learning for user association and resource allocation in heterogeneous networks pp. 1–6 (2018)
27. Zheng, W., Wu, G., Qie, W., Zhang, Y.: Deep reinforcement learning for joint channel selection and power allocation in cognitive internet of things. In: International Conference on Human Centered Computing. pp. 683–692. Springer (2019)

Siyu Yuan, received the B.E in electronic science and technology from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering, BUPT, Beijing, China. His research interests include reinforcement learning, cognitive network slicing and wireless network resource allocation. Email:yuanisyu@bupt.edu.cn

Zhang Yong, received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 2007. He is a Professor with the School of Electronic Engineering, BUPT. He is currently the Director of Fab. X Artificial Intelligence Research Center, BUPT. He is the Deputy Head of the mobile internet service and platform working group, China communications standards association. He has authored or coauthored more than 80 papers and holds 30 granted China patents. His research interests include Artificial intelligence, wireless communication, and Internet of Things. Email: yongzhang@bupt.edu.cn

Qie Wenbo,received the B.E in Yanshan University, Beijing, China, in 2018. She is currently pursuing the M.S in electronic science and technology, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include reinforcement learning, cognitive network slicing, and resource allocation. Email:qwb@bupt.edu.cn

Ma Tengteng, received the B.S degree from the School of Science, Qufu Normal University, Jining, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China. Her current research interests include network slicing, QoS, cognitive radio and virtual network resource allocation. Email: buptteng@foxmail.com

Li Sisi, received the B.S degree from Beijing University of Posts and Telecommunications (BUPT) in 2019. She is currently pursuing the Ph.D. degree in computer science and technology at BUPT. Her research areas include wireless communication, network slicing, network resource allocation, mobile edge computing. Email: ssl123@bupt.edu.cn

Received: July 10, 2020; Accepted: January 12, 2021.