IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 16, 2019, accepted August 19, 2019, date of publication August 23, 2019, date of current version September 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937108

Multi-Agent Deep Reinforcement Learning-Based Cooperative Spectrum Sensing With Upper Confidence Bound Exploration

YU ZHANG[1,2], (Senior Member, IEEE), PEIXIANG CAI[1,2], CHANGYONG PAN[1,2], AND SUBING ZHANG[3]

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Key Laboratory of Digital TV System of Guangdong Province and Shenzhen City, Research Institute of Tsinghua University in Shenzhen, Shenzhen 518057, China
[3]China Electronics Standardization Institute, Beijing 10007, China

Corresponding author: Peixiang Cai (cpx16@mails.tsinghua.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 91738202 and Grant 61790553, in part by the Science Fund for Creative Research Groups of NSFC under Grant 61321061, and in part by the Shenzhen Science and Technology Plan Projects under Grant JCYJ20180306170614484.

ABSTRACT In this paper, a multi-agent deep reinforcement learning method was adopted to realize cooperative spectrum sensing in cognitive radio networks. Each secondary user learns an efficient sensing strategy from the sensing results of some of the selected spectra to avoid interference to the primary users and to coordinate with other secondary users. It is necessary to balance exploration and exploitation in the learning process when using deep reinforcement learning methods, helping explain that upper confidence bound with Hoeffding-style bonus has been adopted in this paper to improve the efficiency of exploration. The simulation results verify that the proposed algorithm, when compared with the conventional reinforcement learning methods with ε-greedy exploration, is much easier to achieve faster convergence speed and better reward performance.

INDEX TERMS Cooperative spectrum sensing, deep reinforcement learning, cognitive radio, upper bound confidence.

I. INTRODUCTION

The rapid development of wireless communication has been accompanied by spectrum resources which are becoming scarcer. However, some investigations [1], [2] have shown that most of the licensed spectra are significantly underutilized, demonstrating that the efficient utilization of spectrum resources has attracted substantial interest of both academic and industry community, thereby promoting the development of cognitive radio (CR) technology [3]. The secondary users (SUs) in CR networks, under the premise of ensuring that the primary users (PUs) are not interfered, can access to the idle spectra for their data transmission. For example, if the PUs start to use their allocated spectra, the SUs need to vacate these spectra immediately, thereby making it necessary for the SUs to sense the changing idle spectra caused by the actions of the PUs, suggesting the importance of the spectrum sensing technology to establish CR networks [4].

The associate editor coordinating the review of this article and approving it for publication was Emmanouil Pateromichelakis.

The SUs may communicate with each other to improve the reliability of sensing results [5], [7]. Besides, *cooperative spectrum sensing* is often required [8], [12] to schedule the sensed spectra of each SU to perceive more idle spectra with limited sensing ability [13], [14]. This paper will study the aforementioned cooperative spectrum sensing problem.

In 2017, the Defense Advanced Research Projects Agency (DARPA) launched and carried out the spectrum collaboration challenge (SC2) essentially in a CR network scenario, calling for smarter technology in spectrum utilization. It is reasonable to apply the rapidly evolving machine learning technology in CR networks [15]. Reference [16] uses Q-learning to propose a centralized algorithm to cope with the spectrum sensing and access problem. Reference [17] implements a distributed Q-learning based spectrum sensing algorithm in which each SU regards the behavior of other SUs as parts of the environment. It is essential for Q-learning to store the estimated values of the cumulative discounted reward (which are usually called as Q-values) of every state-action pairs. Additionally, a linear value function approximation

method is adopted in [18] to approximate the Q-values based on a linear combination of features, the implementation of which can contribute to reducing the required storage ability in large networks but carefully selected features are needed to ensure the accuracy of approximation.

Neural networks in deep learning have been proven to be capable of approaching functions. DeepMind proposed the deep reinforcement learning (DRL) or deep Q-learning network (DQN) using neural networks to approximate Q-values by combing the deep learning with the reinforcement learning [19]. With sufficient training, neural networks can capture precise Q-values to find the optimal policy. Recent years have witnessed the wide study on DQN in the field of dynamic spectrum access problems in CR networks [20], [22]. However, the investigations to apply DQN in cooperative spectrum sensing problem are scarce so far. To the best of our knowledge, this paper serves as the first work to search the optimal cooperative spectrum sensing policy in CR networks by applying DQN.

Besides, the balance between *exploration* and *exploitation* has also been widely studied to analyze the learning efficiency of diverse reinforcement learning methods. Generally speaking, exploration means to explore the policy space to search the optimal policy, whereas exploitation means to adopt the policy with the best reward based on previous experience, suggesting that there is a tradeoff between exploration and exploitation because sufficient exploration is needed to avoid a suboptimal policy. However, the performance deteriorates due to many policies with worse reward are explored.

The heuristic methods, such as ε-greedy methods, are adopted in most of conventional reinforcement learning implementations, including [15], [18], [20], [22], which with probability $1-\varepsilon$ choose the current best action and with probability ε choose action randomly, leading to exploration complexity proportional to experiment time T. A smarter way is to reduce the value of ε as the learning progresses gradually, bringing about less chance of exploration after enough information has been gained, then it can almost always choose the current optimal choice. Recently, [23] proved that for an episodic Markov decision process (MDP), utilizing the exploration strategy based on the upper confidence bounds with Hoeffding-style (UCB-H) bonus can contribute to achieving exploration complexity proportional to $\sqrt{T}$, which the same as theoretically optimal result. Consequently, the UCB-H method is adopted to improve the exploration efficiency, sequentially speeding up the convergence of DQN in this paper. To the best of our knowledge, this is also the first work to combine DQN with UCB-H to solve cooperative spectrum sensing problem in the CR networks.

The rest of this manuscript is organized as follows. Section II briefly introduces the system model of multi-agent cooperative spectrum sensing problem and Section III provides the implementation of cooperative spectrum sensing strategy based on reinforcement learning with UCB-H. In Section IV we propose the algorithm based on DQN with UCB-H. The simulation results and corresponding analysis are given in Section V, and this paper is concluded in Section VI.

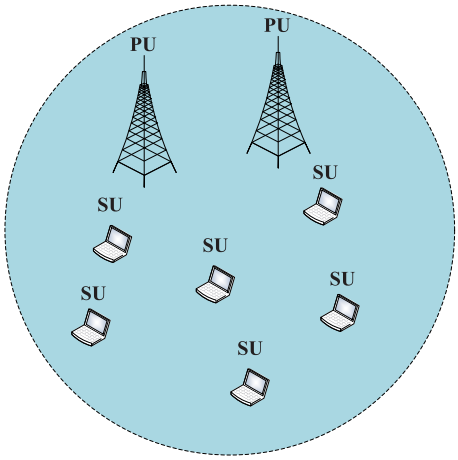

FIGURE 1. The scene of a CR network. The SUs should access the spectra without interference to the PUs.

II. MULTI-AGENT COOPERATIVE SPECTRUM SENSING PROBLEM

In this section, we introduce the system model of the proposed multi-agent cooperative spectrum sensing problem. There are N_{p} PUs, N_{s} SUs and M spectra in a CR network as shown in Fig. 1. We assume that PUs occupy some spectra with certain rules so that their actions can be combined together as an MDP. SUs should ensure that they access spectra without interference to PUs. For the sake of brevity, we do not consider the power control strategy of users. Generally, when a PU occupies the spectra, all SUs cannot occupy these spectra. Owing to the hardware and power constraints, each SU can only sense K ($K < M$) spectra in each time slot. SUs do not know the regular patterns that PUs follow to occupy spectra, hence, they need to predict the idle spectra based on the previous sensing results. We assume that the PUs' occupancy of the spectra is time-slotted, and the clock synchronization of SUs and PUs is sufficiently accurate. Due to the limited spectrum sensing technologies including the energy detection algorithms, false alarm and missed detection occur with probability P_{f} and P_{m}, respectively.

We will not gather the sensing results of all SUs to a fusion center to address the cooperative spectrum sensing as a centralized problem. Instead, we configure a distributed implementation in a multi-agent fashion. Each SU gathers information from the environment and other SUs to decide its own sensing policy.

There may be a collaboration channel like the control channel for interaction. In contrast, the SUs may not interact with each other to realize cooperation so that they may contend for the same spectra with collision. Hence, in practical scenes, the SUs may adopt two different time slot structures as depicted in Fig. 2.

In the first structure, each time slot is divided into a sensing minislot, a collaboration minislot, and an access minislot.

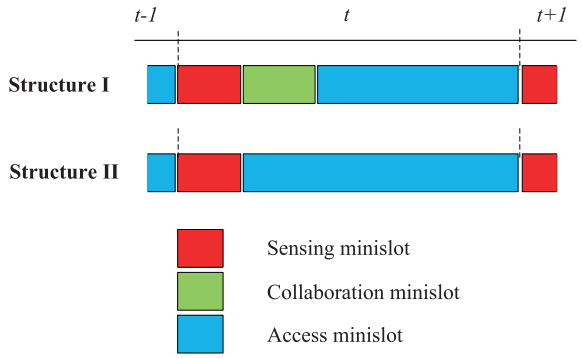

FIGURE 2. Two different time slot structures in the proposed multi-agent cooperative spectrum sensing problem.

In the collaboration minislot, each SU broadcasts its own sensing results with the serial numbers of its sensing spectra to other SUs without conflict. At the same time, the control information required for their spectrum access policy is transmitted in the collaboration minislot.

In the second structure, each time slot only contains a sensing minislot and an access minislot. The SUs cannot coordinate their spectrum access due to the lack of collaboration minislot. As a result, each SU can only access the spectra which are sensed as idle based on its own sensing result in the sensing minislot. If multiple SUs sense the same spectrum, a collision in the access minislot happens and the data transmission in this spectrum fails. The SUs can detect collision by acknowledgment messages and other methods, sequentially adjust their sensing policy to avoid sensing the same spectra as other SUs. It can be verified that the above two structures can achieve similar performance and the first structure will converge faster because the SUs can also obtain part of other SUs' sensing results through collision detection to realize coordination in the second structure.

III. COOPERATIVE SPECTRUM SENSING ALGORITHM BASED ON REINFORCEMENT LEARNING WITH UCB-H

In this section, we focus on the first slot structure as depicted in Section II and propose an algorithm to achieve superior cooperative spectrum sensing performance in distributed SUs.

We denote the set of *states* and *actions* by $\mathcal{S}$ and $\mathcal{A}$ whose size are $|\mathcal{S}|$ and $|\mathcal{A}|$, respectively. We use different subscripts to distinguish $s \in |\mathcal{S}|$ and $a \in |\mathcal{A}|$ of different SUs in different time slot. $s_{i,t} = [s^1_{i,t}, s^2_{i,t}, \cdots, s^M_{i,t}]$ is the state of the ith SU in time slot t, where $s^m_{i,t}$ represents its cooperative sensing result of the mth spectrum. $s^m_{i,t}$ has four different values. $s^m_{i,t} = 0$ means that the spectrum is sensed as occupied, and $s^m_{i,t} = 1$ means the spectrum is not sensed in this slot. When the spectrum is sensed as idle, $s^m_{i,t} = 2$ means that the ith SU broadcast the sensing result of this spectrum earlier than other SUs, and $s^m_{i,t} = 3$ means that one of other SUs broadcast the sensing result first. Differentiating the last two cases is helpful to coordinate the actions of each SU.

The action of an SU can also be composed of its sensing actions for each spectrum. If the number of sensed spectra of each SU is fixed in each slot, we can reduce the dimension of action space by denoting the action of the ith SU in time slot t by $a_{i,t} = [a^1_{i,t}, a^2_{i,t}, \cdots, a^K_{i,t}]$, where $a^k_{i,t} \in \{1, 2, \cdots, M\}$ represents the serial number of its kth sensed spectrum.

As mentioned in Section II, we assume that the PUs' actions can be formulated as an MDP. Q-learning is a classical reinforcement learning algorithm whose iterative formula is derived from MDP. Consequently, we adopt it as the basis to seek the optimal policy for dealing with the cooperative sensing problem.

The value $v(s, \pi)$ represents the expectation of cumulative discounted reward begin with state s and choose action using policy π. Denote the reward of choosing action a in state s by $r(s, a)$. Following the derivation in [24], according to the Bellman equation, there is an optimal strategy π^*, which can achieve the optimal value of the state s, that

$$v(s, \pi^*) = \max_a \left\{ r(s, a) + \lambda \sum_{s'} p(s'|s, a) v(s', \pi^*) \right\}, \quad (1)$$

where λ is the discount factor and $p(s'|s, a)$ is the probability that the state transits from s to s' when choosing action a.

Define $Q^*(s, a) = r(s, a) + \lambda \sum_{s'} p(s'|s, a) v(s', \pi^*)$ as the cumulative discounted reward begin with state s and action a, the optimal policy π^* can be obtained by $v(s, \pi^*) = \max_a Q^*(s, a)$. Using $Q(s, a)$ to represent the estimated value of $Q^*(s, a)$, it has proven in [25] that randomly initializing $Q(s, a)$ and using a learning rate $\alpha \in [0, 1)$ which will decay over time, $Q(s, a)$ can converge to $Q^*(s, a)$ using the observed reward R to update $Q(s, a)$ every iteration:

$$Q(s, a) = (1 - \alpha) Q(s, a) + \alpha [R + \lambda \max_{a' \in \mathcal{A}} Q(s', a')]. \quad (2)$$

And the set of $Q(s, a)$ of all state-action pairs is called Q-values as mentioned in Section I.

When a SU successfully accessed a spectrum, it got the reward from this spectrum. For simplicity, we assume that each spectrum gives the same reward and the spectrum access can be well scheduled through collaboration minislot in the context of the first slot structure. Besides, to avoid the reward of one idle spectrum being counted by multiple SUs, only the SU which broadcasted the sensing result earlier than other SUs in the collaboration minislot got the reward from this spectrum. Fig. 3 shows the schematic of the cooperative spectrum sensing algorithm based on Q-learning. In slot t, each SU sense the spectra according to $a_{i,t}$ to observe $s_{i,t+1}$, and the observed reward $R_{i,t}$ of the ith SU in slot t can be obtained directly from $a_{i,t}$ and $s_{i,t+1}$ that

$$R_{i,t} = \sum_{m \in a_{i,t}} I\left[s^m_{i,t+1} = 2\right], \quad (3)$$

where $I\left[s^m_{i,t+1} = 2\right]$ is the indicator function having value 1 if $s^m_{i,t+1} = 2$ and 0 otherwise. $R_{i,t}$ will be used to update the Q-values, and then the ith SU choose the action $a_{i,t+1}$ for the next slot based on $s^m_{i,t+1}$ according to the updated Q-values.

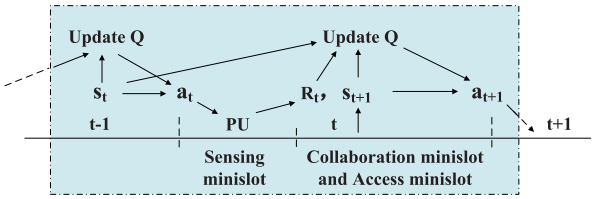

FIGURE 3. The schematic of cooperative spectrum sensing algorithm based on Q-learning.

To accelerate the convergence of Q-values in Q-learning, the tradeoff between exploration and exploitation must be taken into account when selecting the actions. While exploring the unknown environment, it is necessary to try to maximize the received reward. Solely choosing the action with the current maximum Q-values. Generally, the greedy method will fall into the local suboptimal solution due to insufficient exploration. However, if too many explorations are done, the learning process will select a number of actions with poor rewards, rendering the low reward obtained at the outset.

The *regret* of policy, namely the difference of cumulative reward between this policy and the optimal policy, is usually used as the evaluation standard of the exploration efficiency. As mentioned in Section I, ε-greedy method has proven to have regret which is proportional to the experiment time T. Some methods gradually reduce the value of ε in the process of exploration, so that the probability of exploration decreases with the learning process to obtain a smaller regret. However, such heuristic methods fail to give a clear lower bound of regret to guarantee their performance. Recently, Chi Jin *et al.* have been inspired by the multi-armed bandit problems to propose the UCB-H algorithm [23]. They used powerful mathematical tools to prove the following theorem that their algorithm achieves better regret performance (proportional to $\sqrt{T}$) than conventional algorithms.

Theorem 1: For experiment time T in an episodic-MDP with H steps in each episodic, for any $p \in (0, 1)$, there exists an absolute constant $c > 0$, let $b_\tau = c\sqrt{H^3\iota/\tau}$, where $\iota := \log(|\mathcal{S}||\mathcal{A}|T/p)$ and τ is the times the state-action pair (s, a) has been visited, the regret of Q-learning with UCB-H bonus is at most $O(\sqrt{H^4|\mathcal{S}||\mathcal{A}|T\iota})$ with probability $1 - p$.

The Q-learning with UCB-H bonus in [23] is essentially using the UCB of Q-values in value iteration. We adjust the conclusion in the episodic-MDP to the general MDP, i.e., $H = 1$, which corresponds to the cooperative spectrum sensing scene aforementioned, then we can update the UCB of the Q-values as

$$Q(s, a) \leftarrow (1 - \alpha')Q(s, a) + \alpha'[R + V(s') + b_\tau], \quad (4)$$

where $V(s') = \max_{a' \in \mathcal{A}} Q(s', a')$ and $\alpha' = 2/(1 + \tau)$.

We implemented the algorithm based on multi-agent Q-learning with UCB-H for the proposed cooperative sensing problem, and the details of the proposed algorithm is illustrated in **Algorithm 1**.

Algorithm 1 Cooperative Spectrum Sensing Algorithm Based on Multi-Agent Q-Learning With UCB-H

Initial $Q_i(s, a)$ for each $i \in N_s$;
for $t = 1 : T$ **do**
  # Time slot t:
  # Sensing minislot:
  for $i \in N_s$ **do**
    # The ith SU:
    Sense the spectra according to $a_{i,t}$, get the sensing result;
  end
  # Collaboration minislot:
  for $i \in N_s$ **do**
    Send the local sensing result to other SUs and receive the sensing results from other SUs, obtain $R_{i,t}$ and $s_{i,t+1}$;
  end
  # Access minislot:
  for $i \in N_s$ **do**
    Use some access policy to access the spectrum;
  end
  # Learning the sensing policy:
  for $i \in N_s$ **do**
    # Calculate UCB-H:
    $N_i(s_{i,t}, a_{i,t}) = N_i(s_{i,t}, a_{i,t}) + 1$;
    $\tau = N_i(s_{i,t}, a_{i,t})$;
    $b_\tau = c\sqrt{\iota/\tau}$;
    # Update $Q(s, a)$ of the ith SU:
    $Q_i(s_{i,t}, a_{i,t}) =$
    $(1 - \alpha')Q_i(s_{i,t}, a_{i,t}) + \alpha'[R_{i,t} + V_i(s_{i,t+1}) + b_\tau]$;
    $V_i(s_{i,t}) = \max_{a'} Q_i(s_{i,t}, a')$;
    # Choose action $a_{i,t+1}$:
    $a_{i,t+1} = \arg\max_{a'} Q_i(s_{i,t+1}, a')$;
  end
end

IV. COOPERATIVE SPECTRUM SENSING ALGORITHM BASED ON DQN WITH UCB-H

There may be numerous spectra in practical CR networks. As a result, state space and action space are tremendous. In these cases, the number of Q-values which need to be stored may be very large, which will take up a number of storage resources. Besides, to converge Q-values, it is necessary to go through each state-action pair (s, a) at least once. Thereupon, the training time required for convergence is too long to be practical. $Q(s, a)$ is essentially a value function with input (s, a). Accordingly, some value function approximation methods can be executed at extracting features from the experienced state-action pairs to estimate Q-values, making the amount of required data much smaller than the number of state-action pairs. The linear value function approximation which is conducted in [18] represents Q-values with a linear combination of features, therefore the amount of data to be stored can be greatly reduced. However, the approximation accuracy is limited by the selected features for different spectra changing pattern.

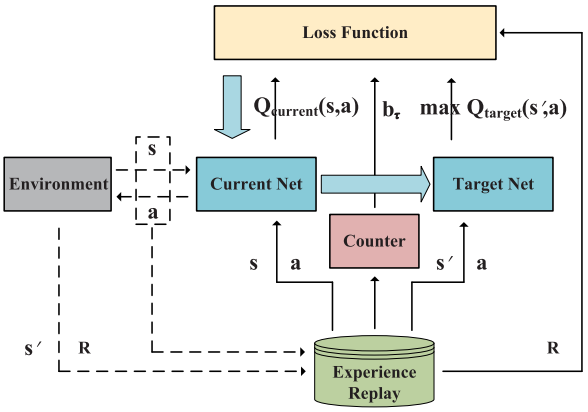

FIGURE 4. The schematic of DQN to be used in the cooperative spectrum sensing problem.

In this section, taking advantage of both deep learning and reinforcement learning, we adopted DQN to realize the superior performance of value function approximation in estimating Q-values using neural networks. Using UCB-H exploration in the learning process of DQN is favorable. It is illustrated in [23] that the initial state s in each episode should be randomly selected by the opponent to disrupt the correlation between data from each episode when using UCB-H exploration. When we applied the conclusion in episodic-MDP to the scene where $H = 1$, using continuous training data results in the correlation of data between each episode, leading to a large variance in the convergence speed of the training process. The objective of eliminating correlation in DQN is consistent with the requirement of UCB-H exploration. The samples obtained by Q-Learning are correlated, whereas the neural network is a supervised learning model that requires data with the independent and identical distribution. Thanks to the characteristics of Q-learning that it is an off-policy learning method which can use the previous experience to learn, DQN utilizes the experience replay method to store the past data in an experience replay memory and randomly samples them for follow-up learning, which can eliminate the correlation and non-stationary distribution of data. This also makes data utilization higher because a sample can be used multiple times. Besides, DQN builds two networks (the *Current network* and the *Target network*) with the same structure to estimate the target value and current value of Q-values, respectively. The slower parameter update of the Target network can avoid optimistic value estimation and can also cut off the correlation between training data. Thence, DQN is conducive to implementing the cooperative spectrum sensing algorithm which implants the UCB-H method to explore. The schematic of DQN is illustrated in Fig. 4 and the detailed process of the cooperative sensing based on multi-agent DQN with UCB-H is described in **Algorithm 2**.

The cooperative spectrum sensing algorithms based on DQN and Q-learning are the same in terms of their operation in each minislot. However, different from Q-learning which stores Q-values for all state-action pairs and using the formula in **Algorithm 1** to update Q-values, DQN uses the neural network to estimate Q-values. It stores a training sample to the experience relay memory each slot and begin to randomly sample training data (s, a, r, s') from experience relay memory after the samples are sufficient in T_s slots. Hereafter, the training sample is input into the Target network and Current network to obtain $Q^{\text{Target}}(s', a)$ and $Q^{\text{Current}}(s, a)$ as the estimation of $Q(s', a)$ and $Q(s, a)$, respectively. They are entered into the *loss function* together with the reward R and the calculated UCB-H bonus b_τ for gradient descent to update the parameters of the Current network, where the loss

Algorithm 2 Cooperative Spectrum Sensing Algorithm Based on Multi-Agent DQN With UCB-H

```
Initial Current network and Target network for i ∈ N_s;
for t = 1 : T do
    for i ∈ N_s do
        Sense the spectra according to a_{i,t}, get the
        sensing result;
    end
    for i ∈ N_s do
        Send the local sensing result to other SUs and
        receive the sensing results from other SUs,
        obtain R_{i,t} and s_{i,t+1};
    end
    for i ∈ N_s do
        Use some access policy to access the spectrum;
    end
    # Store the training samples:
    for i ∈ N_s do
        Store (s_{i,t}, a_{i,t}, R_{i,t}, s_{i,t+1}) to the relay memory;
    end
    # Training when samples are enough:
    for i ∈ N_s do
        if t > T_s then
            Sample a training sample (s, a, R, s′) from
            the experience relay memory arbitrarily;
            Input training sample to neural networks to
            obtain Q_i^Target(s′, a) and Q_i^Current(s, a);
            N_i(s, a) = N_i(s, a) + 1, τ = N_i(s, a),
            b_τ = c√(ι/τ);
            Conducting gradient descent to the loss
            function (5);
            Update the Current network;
            # Update Target network every T_u slots:
            if mod (t, T_u) == 0 then
                Target network ← Current network;
            end
            a_{i,t+1} = arg max_{a′} Q_i^Current(s_{i,t+1}, a′);
        else
            Select a_{i,t+1} randomly;
        end
    end
end
```

function can be expressed as follows:

$$L = (R + b_\tau + \lambda Q^{\text{Target}}(s', a) - Q^{\text{Current}}(s, a))^2. \quad (5)$$

And the value of the Current network will be directly copied to the Target network every T_u slots. In the first T_s slots, because the neural networks had not been trained, the actions are selected randomly. The action with the largest Q-value is selected after T_s slots.

V. SIMULATION AND ANALYSIS

Without loss of generality, we assume that each SU can sense one spectrum each time slot in the simulations, namely, $K = 1$, and the proposed algorithm also works in scenarios where the SUs can sense multiple spectra. The channel models have significant impact on the false alarm and missed detection probability of the energy detection algorithm. For the sake of simplicity, we consider the effects of channel models, signal-to-noise ratio and energy detection algorithm comprehensively to assume that the probability of missed detection can be controlled at a very low level with a sufficiently long detection time, i.e., $P_m \approx 0$, and the probability of false alarm is set as $P_f = 0.1$ in the first simulation experiment. Since the probability of missed detection is negligible, the decision of cooperative spectrum sensing results can adopt the OR-rule. As long as some SUs sensed the spectrum as idle, the cooperative sensing result of this spectrum is idle in this slot.

In simulations to verify the performance of the proposed algorithm in Fig. 5, we first set the number of SUs $N_S = 5$ and the number of spectra $M = 6$ in the CR network. The size of action space is $|\mathcal{A}| = 6$ owing to $K = 1$. Besides, according to this assumption, $s^m_{i,t} = 3$ is equivalent to $s^m_{i,t} = 0$ because each SU will try to avoid sensing the same spectrum with other SUs. There is a PU which occupies one spectrum per time slot for its own data transmission (so the number of idle spectra in each time slot is 5). For intuitive understanding, we assume that the PU adopts a type of scanning method. At the beginning of each time slot, the PU selects another spectrum to occupy with an 80% probability, and stays in the previous spectrum with a 20% probability. The proposed method also works when the PU occupies spectra with other patterns. As mentioned above, the false alarm probability and missed detection probability of the sensing result are $P_f = 0.1$ and $P_m = 0$, respectively. The SUs adopt the first time slot structure. We conducted 100 simulation experiments and averaged the results to verify the performance of the algorithms based on Q-learning with UCB-H, based on Q-learning with ε-greedy, based on DQN with UCB-H and based on DQN with ε-greedy, respectively. Each simulation experiment runs $T = 5000$ time slots and the rewards of each SU is recorded to calculate the average reward. The discount factor $\lambda = 0.9$ in above algorithms. In the algorithms with ε-greedy, $\varepsilon = 0.1$ and α decades with time t as $\alpha = T/(T + 10t)$. In the algorithms with UCB-H, $p = 0.01$, $c = 0.5\sqrt{\log(|\mathcal{S}||\mathcal{A}|T/p)}$ and α' decades with the times τ the state-action pair (s, a) has been visited as $\alpha' = 2/(1 + \tau)$

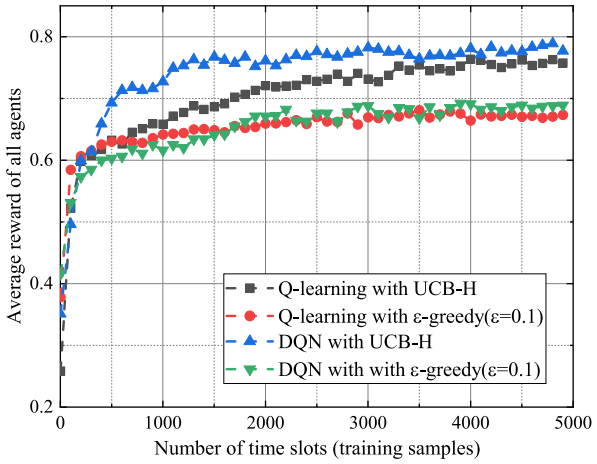

FIGURE 5. The average reward performance of different algorithms ($N_S = 5$, $M = 6$).

(carefully adjust c may further improve the performance of the algorithms with UCB-H). In the two algorithms based on DQN, $T_s = 50$, $T_u = 100$ and the neural networks adopt the structure of convolutional neural networks. In our simulations, we only build up two convolution layers, each with 10 neurons, and the activation function is the ReLu function. The implementation of the convolutional layers is simple, and the sparsity of the inter-layer connection reduces the total amount of weight parameters, which is beneficial to the rapid convergence speed of the neural network and reduces the memory overhead in the calculation. When the problem dimension is larger, small neural networks may also fail to meet the approximation accuracy requirements, and it will require more layers in the neural network structures.

It can be seen from Fig. 5 that the two algorithms with UCB-H can obtain a higher average reward than the two algorithms with ε-greedy, which confirm the better exploring efficiency of UCB-H exploration. When the exploration method is fixed, the algorithms based on DQN can achieve better performance than those based on Q-learning. This is because the DQN can reduce the correlation between the training data, sequentially reduce the probability that the learning result converged to the local optimal solution. Especially when using UCB-H exploration, the algorithm based on DQN can achieve faster convergence speed, for example under our simulation settings, the algorithm based on DQN almost converges within 2000 slots, and the algorithm based on Q-learning needs more than 3000 slots to converge.

In Fig. 6, we show the performance of different algorithms in the early stage of learning. The two algorithms based on DQN need to collect some training samples to form the experience relay memory in the first few time slots, so they start slower than the two algorithms based on Q-learning. Besides, the two algorithms with UCB-H will first explore a part of state-action pairs, resulting in their selections of many worse actions at the outset. Nevertheless, the algorithms ε-greedy using small ε (such as $\varepsilon = 0.1$) prefer to choose actions with

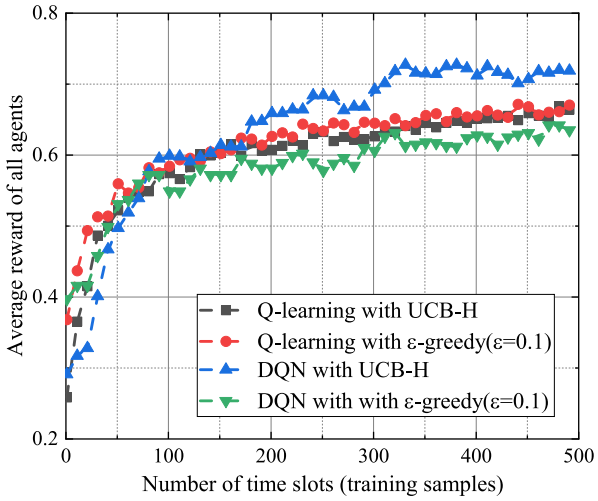

FIGURE 6. The convergency of different algorithms with small number of training samples ($N_S = 5$, $M = 6$).

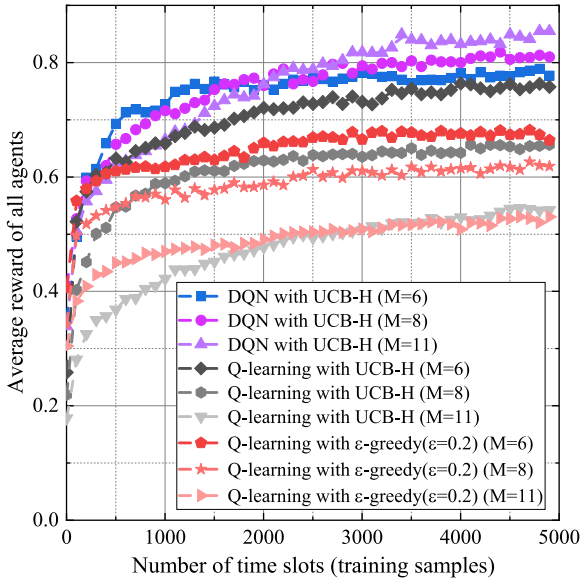

FIGURE 7. The average reward performance of different methods with different N_S and M.

the highest expected reward based on current Q-values, hence their average reward will be higher than the algorithms with UCB-H in the early stage of simulation experiments.

For different N_S and M, we reveal the simulation results in Fig. 7. For the algorithms based on Q-learning, the number of state-action pairs increase exponentially with the number of spectrums. For example, when $N_S = 20$ with $M = 21$, the number of state-action pairs is $3^{21} \times 21 \approx 10^{11}$, and the size of the Q-table will exceed the memory of the general computers, so the simulation experiments of the algorithms based on Q-learning can hardly be performed when M is large. This is also the main purpose we need to use DQN to implement the cooperative spectrum sensing algorithm. Therefore, we only give the comparison results of the performance of these algorithms in the cases that $N_S = 5$ with $M = 6$, $N_S = 7$ with $M = 8$ and $N_S = 10$ with

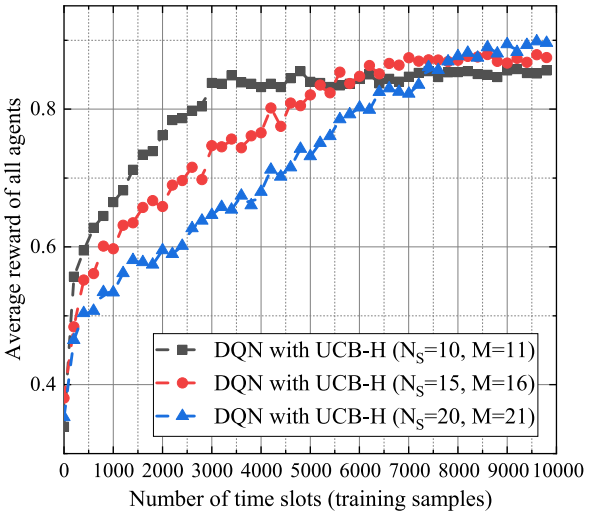

FIGURE 8. The performance of the proposed algorithm with different numbers of SUs and spectra ($P_f = 0.1$).

$M = 11$, respectively. We set $\varepsilon = 0.2$ in the ε-greedy to enhance exploration in the early phase of learning. Fig. 7 showed the trend that the proposed algorithm based on DQN with UCB-H outperforms the conventional algorithms based on Q-learning. The convergence of the two algorithms based on Q-learning significantly decreases when M is large, whereas the convergence of the proposed algorithm is less affected by the increase of M. This is because DQN estimates the Q-values of different state-action pairs by extracting features using neural networks, so that the estimated value of Q-values for any (s, a) can be obtained after the neural networks are trained with a certain amount of training samples. The two algorithms based on Q-learning still need to explore the Q-values of most state-action pairs to achieve better performance, which cannot achieve convergence in a short time and become more easily to converge to the local optimal solutions when M is large, causing the significant degradation of their overall performance. In addition, the PU only occupies one spectrum each slot in the simulation, so the larger M, the smaller the probability that the SUs sensed a spectrum which is occupied by the PU. Thereupon, the SUs obtain a larger average reward after convergence.

We carry out simulation experiments with 10000 time slots to show the performance of the proposed algorithm with larger M in Fig. 8. It can be seen that superior reward performance can be exhibited even when M is large. However, for a network with many SUs and spectra, the large state-action space also results in the difficulty for the proposed algorithm to converge. Each SU regards other SUs as parts of the environment, and each SU learns its optimal strategy from its observation of the environment. In the proposed algorithm, when the collision happened that two SUs sense the same spectrum, the later SU changes its strategy to avoid collision with the former SU, so the strategy of the later SU converges after the former one converged. As a result, the convergence speed of the overall reward performance decreases when the

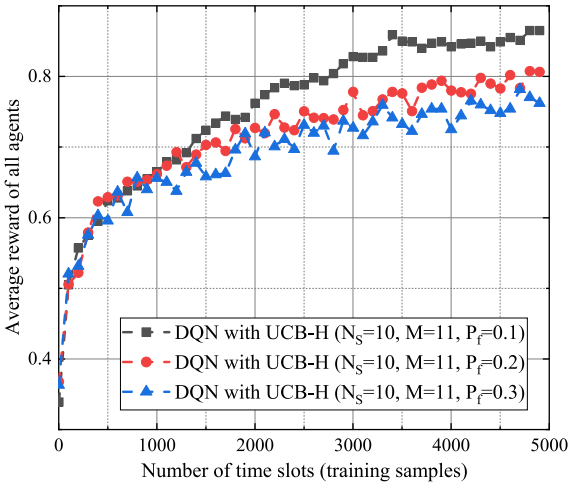

FIGURE 9. The performance of the proposed algorithm with different false alarm probabilities ($N_S = 10$, $M = 11$).

number of SUs increases. For example, when $N_S = 20$ with $M = 21$, the proposed algorithm needs about 8000 time slots to converge.

Besides, in practical scenarios, the channel condition may be poor and the signal-to-noise ratio is low, causing the high false alarm probability. In Fig. 9, we show the performance of the proposed algorithm with different false alarm probabilities. It can be seen from the simulation results that the larger false alarm probability leads to the performance degradation of the proposed algorithm. Because the SUs cannot confirm the correctness of their observation of the environment, the more wrong sensing results were used for training, the larger probability that the proposed algorithm converged to a sub-optimal strategy. So the average reward performance decreases and the stability of the learning process is weakened with larger false alarm probability, but the proposed algorithm still outperforms the conventional algorithms based on Q-learning in the same situation.

In the aforementioned simulations, each SU can only sense one spectrum each slot and the total number of spectra sensed by the SUs is not more than the number of spectra in the network. Consequently, the main purpose of the cooperative spectrum sensing problem is to find more idle spectra and the SUs will avoid sensing the same spectra as other SUs. If the total number of spectra that the SUs can sense is greater than the number of spectra in the network when $P_m > 0$, then each spectrum can be sensed by more than one SUs. As a result, a tradeoff should be made on the number of sensed spectra and the reliability of the sensing result. The optimal number of SUs to sense each spectrum will be analyzed in our future work.

VI. CONCLUSION

In this paper, we proposed a novel cooperative spectrum sensing algorithm for CR networks. By implementing DQN with UCB-H to improve the exploration efficiency, the proposed algorithm can achieve better reward performance with faster convergence speed than the conventional algorithms based on Q-learning with ε-greedy, especially in large networks. The simulation results verify the superior performance of the proposed algorithm, which can promote the development of smarter CR network technology to achieve more efficient utilization of spectra. We will study the cooperative sensing strategy to deal with the relevant sensing results due to correlated shadow, sequentially making the fusion of sensing results more reliable based on multi-agent deep reinforcement learning in future work.

REFERENCES

[1] P. Kolodzy and I. Avoidance, "Spectrum policy task force," Federal Commun. Commission, Washington, DC, USA, Tech. Rep. 02-135, 2002.
[2] V. Blaschke, H. Jaekel, T. Renk, C. Kloeck, and F. Jondral, "Occupation measurements supporting dynamic spectrum allocation for cognitive radio design," in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw. Commun.*, Aug. 2007, pp. 50–57.
[3] J. Mitola and G. Q. Maguire, Jr., "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Apr. 1999.
[4] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 116–130, 1st Quart., 2009.
[5] S. Mishra, A. Sahai, and R. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. IEEE Int. Conf. Commun.*, vol. 4, Jun. 2006, pp. 1658–1663.
[6] W. Ejaz, H. Hassan, and A. Azam, "Cooperative spectrum sensing for cognitive radio networks application: Performance analysis for realistic channel conditions," in *Advances in Computational Science, Engineering and Information Technology*. Berlin, Germany: Springer, 2013.
[7] J. Zhu, Z. Xu, F. Wang, B. Huang, and B. Zhang, "Double threshold energy detection of cooperative spectrum sensing in cognitive radio," in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw. Commun.*, Singapore, 2008, pp. 1–5.
[8] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *Proc. IEEE Int. Symp. New Frontiers Dyn. Spectr. Access Netw.*, Baltimore, MD, USA, Nov. 2005, pp. 137–143.
[9] X. Liu, M. Jia, Z. Na, W. Lu, and F. Li, "Multi-modal cooperative spectrum sensing based on dempster-shafer fusion in 5G-based cognitive radio," *IEEE Access*, vol. 6, pp. 199–208, 2018.
[10] J. Tong, M. Jin, Q. Guo, and Y. Li, "Cooperative spectrum sensing: A blind and soft fusion detector," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2726–2737, Apr. 2018.
[11] W. Ejaz, G. A. Shah, N. U. Hasan, and H. S. Kim, "Energy and throughput efficient cooperative spectrum sensing in cognitive radio sensor networks," *Trans. Emerg. Telecommun. Technol.*, vol. 26, no. 7, pp. 1019–1030, Jul. 2015.
[12] Z.-H. Wei, B.-J. Hu, E.-J. Xia, and S.-H. Lu, "A contention-free reporting scheme based MAC protocol for cooperative spectrum sensing in cognitive radio networks," *IEEE Access*, vol. 6, pp. 38851–38859, 2018.
[13] R. Akhtar, A. Rashdi, and A. Ghafoor, "Grouping technique for cooperative spectrum sensing in cognitive radios," in *Proc. Int. Workshop Cogn. Radio Adv. Spectr. Manage.*, Aalborg, Denmark, May 2009, pp. 80–85.
[14] S. Pradeep and T. Sudha, "Optimal sensing scheduling for green cognitive radio," in *Proc. Int. Conf. Control Commun. Comput. India*, Trivandrum, India, Nov. 2015, pp. 424–428.
[15] B. F. Lo and I. F. Akyildiz, "Reinforcement learning for cooperative sensing gain in cognitive radio ad hoc networks," *Wireless Netw.*, vol. 19, no. 6, pp. 1237–1250, Aug. 2013.
[16] Y. Teng, Y. Zhang, F. Niu, C. Dai, and M. Song, "Reinforcement learning based auction algorithm for dynamic spectrum access in cognitive radio networks," in *Proc. IEEE Veh. Technol. Conf.-Fall*, Ottawa, ON, Canada, Sep. 2010, pp. 1–5.
[17] M. A. Aref, S. K. Jayaweera, and S. Machuzak, "Multi-agent reinforcement learning based cognitive anti-jamming," in *Proc. IEEE Wireless Commun. Netw. Conf.*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
[18] J. Lundén, V. Koivunen, S. R. Kulkarni, and H. V. Poor, "Reinforcement learning based distributed multiagent sensing policy for cognitive radio networks," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw.*, Aachen, Germany, May 2011, pp. 642–646.

[19] V. Mnih, K. Kavukcuoglu, and D. Silver, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[20] 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.
[21] 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, Apr. 2018.
[22] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, New Orleans, LA, USA, Mar. 2017, pp. 2087–2091.
[23] C. Jin, Z. Allen-Zhu, and S. Bubeck, "Is q-learning provably efficient?" in *Proc. Adv. Neural Inf. Process. Syst.*, Montréal, QC, Canada, 2018, pp. 4868–4878.
[24] J. Hu and P. Wellman, "Multiagent reinforcement learning: Theoretical framework and an algorithm," in *Proc. Int. Conf. Mach. Learn.*, Madison, WI, USA, vol. 98, 1998, pp. 242–250.
[25] C. J. C. H. Watkins and P. Dayan, "*Q*-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

YU ZHANG (M'07–SM'12) received the B.E. and M.S. degrees in electronics engineering from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from Oregon State University, Corvallis, OR, USA, in 2006. In 2007, he was an Assistant Professor with the Research Institute of Information Technology, Tsinghua University, for eight months, where he is currently an Associate Professor with the Department of Electronic Engineering. His current research interests include the performance analysis and detection schemes for MIMO-OFDM systems over doubly selective fading channels, transmitter and receiver diversity techniques, and channel estimation and equalization algorithm.

PEIXIANG CAI received the B.E. degree from Tsinghua University, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include communication systems, intelligent transportation systems, information theory, and signal processing.

CHANGYONG PAN born in Anhui, China, in 1975. He is currently a Full Professor with the Research Institute of information Technology and the Deputy Director of the DTV R&D Center, Tsinghua University. He has authored or coauthored more than 180 technical articles and published eight technical books. He holds 34 patents. He was a recipient of the National Technical Award three times. He is also the winner of numerous other awards.

SUBING ZHANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2001. He is currently a full-time Researcher with the China Electronics Standardization Institute.