Power Control Based on Deep Reinforcement Learning for Spectrum Sharing

Haijun Zhang, *Senior Member, IEEE*, Ning Yang, Wei Huangfu, Keping Long, *Senior Member, IEEE*, and Victor C. M. Leung, *Fellow, IEEE*

***Abstract*— In the current researches, artificial intelligence (AI) plays a crucial role in resource management for the next generation wireless communication network. However, traditional RL cannot solve the continuous and high dimensional problems. To handle these problems, the concept of deep neural network (DNN) is introduced into RL to solve high dimensional problems. In this paper, we first construct an information interaction model among primary user (PU), secondary user (SU) and wireless sensors in a cognitive radio system. In the model, the SU is unable to get the power allocation information of the PU, and needs to use the received signal strengths (RSSs) of the wireless sensors to adjust its own power. The PU allocates transmit power relying on its power control scheme. We propose an asynchronous advantage actor critic (A3C)-based power control of SU that is a parallel actor-learners framework with root mean square prop (RMSProp) optimization. Multiple SUs learn power control scheme simultaneously on different CPU threads, reducing neural network gradient update interdependence. To further improve the efficiency of spectrum sharing, the distributed proximal policy optimization (DPPO)-based power control is proposed which is an asynchronous variant of actor-critic with adaptive moment (Adam) optimization. It enables the network to converge quickly. After several power adjustments, the PU and the SU meet quality of service (QoS) requirements and achieve spectrum sharing.**

***Index Terms*— Deep reinforcement learning (DRL), cognitive radio network, spectrum sharing, power control.**

Manuscript received August 1, 2019; revised December 30, 2019; accepted March 9, 2020. Date of publication March 24, 2020; date of current version June 10, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61822104 and Grant 61771044, in part by the Beijing Natural Science Foundation under Grant L172025 and Grant L172049, in part by 111 Project under Grant B170003, in part by the Fundamental Research Funds for the Central Universities under Grant FRF-TP-19-002C1, RC1631, and in part by the Beijing Top Discipline for Artificial Intelligent Science and Engineering, University of Science and Technology Beijing. The associate editor coordinating the review of this article and approving it for publication was X. Chen. *(Corresponding authors: Wei Huangfu; Haijun Zhang.)*

Haijun Zhang, Ning Yang, Wei Huangfu, and Keping Long are with Institute of Artificial Intelligence, the Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing 100083, China (e-mail: haijunzhang@ieee.org; b20170322@xs.ustb.edu.cn; huangfuwei@ustb.edu.cn; longkeping@ustb.edu.cn).

Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: vleung@ieee.org).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.2981320

I. Introduction

As THE rapid growth of data traffic, how to meet the higher quality of service (QoS) of users with limited spectrum has become a crucial problem [1], [2]. Artificial intelligence (AI) is an important research topic for next generation wireless communication [3]–[5]. It has been widely used for resource management in wireless communication, such as spectrum sharing [6], access control [7], and transmit power control [8]. However, there still exists several issues on spectral efficiency, power efficiency, network convergence, etc., which call for more efficient spectrum sharing technologies.

Meanwhile, in academia, the role of the primary user (PU) is divided into active user model and passive user model for the spectrum sharing. In the active model, there was a cooperative [9] or non-cooperative [10] relationship between the PU and the secondary user (SU). The information interaction was performed to improve the system transmission performance. However, the passive user model in literature, SU performed spectrum sensing to find the idle spectrum or power allocation [11], [12]. When PU played the role of the passive PU, the PU allocates its transmit power relying on its power control scheme.

In current works, there are many ways to address power allocation and power control problem, such as optimization theory, game theory and machine learning. Optimization-based algorithms have been studied, including difference of convex programming, interior point methods, Lagrangian multiplier. In [13], the optimization problem was non-convex which was transform into convex optimization problem using difference of convex. In [14], the secure power allocation algorithm was converted to a convex geometric programming problem, using interior point method to solve it. In [15], the power optimization problem was treated as mixed-integer programming problem that was solved by the Lagrangian dual decomposition.

Deep reinforcement learning (DRL) is the method that uses policy function or deep neural network (DNN) approximation function. The policy function includes policy gradient method and DNN approximation function includes deep Q learning. The agents learn policies and maximize their rewards during the interaction with the environment. The DRL methods, such as, deep Q network (DQN) [20], asynchronous advantage actor critic (A3C) [21] and distributed proximal policy optimization (DPPO) [22], are suitable for solving problems with high-dimensional. The traditional DQN needs the memory replay and fixed Q-target to work well. The A3C method is based on

actor-critic mechanism and uses parallel computing to break the data dependency. The DPPO method can limit the update steps of the policy and operate on a multi-threaded CPU in a distributed form. The objective functions of DRL are complex, because they involve the sum of a large number of data likelihood functions. To deal with these objective functions, many optimization algorithms were investigated, including stochastic gradient descent (SDG) [23], root mean square prop (RMSprop) [24] and adaptive moment (Adam) [25].

Many resource management problems, such as, caching, transmission scheduling, spectrum access and power management in wireless communication are modeled as markov decision processes (MDPs). A novel DRL approach was studied for cache-enabled wireless networks, which used DQN to approximate Q value-action function [26]. A deep learning (DL) transmission scheduling mechanism based on Q learning was proposed to maximize system throughput according to transmitting different buffer data packets through multiple channels [27]. A distributed dynamic spectrum access algorithm was proposed [28]. The current state of user was mapped to the action of spectrum access to maximize network revenue. For cloud resource allocation problem, a DRL based power management strategy was proposed to reduce power consumption [29].

Although several DRL-based researches have been studied for resource management [6], [30]–[32]. However, there are continuous and high dimensional problems in the wireless communication network model, and the Q-Learning algorithm cannot solve the continuous domain problems since the state is infinite. Besides, the gradient update of neural network depends on each other in DQN method, the network cannot achieve the expected learning purpose. The power control for spectrum sharing is related to continuous domain. Therefore, we focus on DRL algorithms based on the actor-critic framework for the power control, such as, A3C and DPPO, combining the methods of policy-based and value-based. Power control with A3C and DPPO can significantly reduce the relevance of updates and improve network convergence.

The contributions of this paper mainly have the following aspects:

- The power control is investigated in cognitive radios for spectrum sharing, guaranteeing the QoS requirements of PU and SU. We first construct an information interaction model among PU, SU and wireless sensors. In the model, the RSSs of the wireless sensors are spatially distributed to help the PU obtain the power allocation information of SU.
- The A3C-based power control is proposed where PUs and SUs share the same spectrum according to adjusting the power allocation. The proposed scheme is a parallel actor-learners framework with RMSProp optimization. Multiple SUs learn simultaneously power allocation scheme on different CPU threads. It reduces the relevance of update and has a stabilizing process for power control.
- The DPPO-based power control is investigated. It optimizes the power allocation function using Adma optimization. The traditional policy gradient method performs a gradient update for each data sample, whereas the proposed scheme can achieve minibatch updates for multiple epochs. It enables the network to converge quickly and meet the QoS of users.

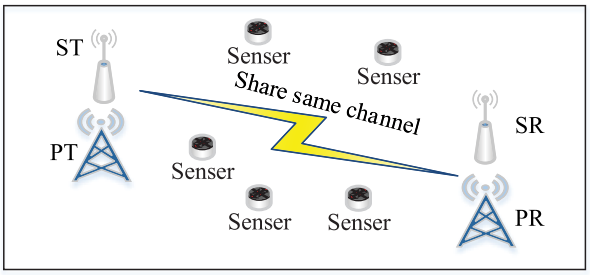

Fig. 1. System model.

The rest of this paper is organized as follows. The system model of cognitive radio network is discussed in Section II. Section III describes the A3C-based power control and the DPPO-based power control schemes for the system. The simulation results are presented in Section V, and finally the paper is concluded in Section VI.

II. System Model

A. Problem Formulation

We investigate a power control mechanism between PU and SU in the underlay access mode for the cognitive radio network. The system model includes a primary transmitter (PT), a primary receiver (PR), a secondary transmitter (ST) and a secondary receiver (SR). The system model is shown in Fig. 1. The aim of the PU is to share the same spectrum resource with the SU, both of them can transmit their signals successfully with the demand of QoS.

The transmit power of them synchronously update on a time frame. The transmit power for the primary transmitter and secondary transmitter is divided into the set of discrete powers, which is represented as

$$\mathcal{P} = \{p_i^1, p_i^2, \ldots, p_i^L\}, \tag{1}$$

where $i \in \{1, 2\}$, $p_i^1 < p_i^2 < \ldots < p_i^L$ and L is the number of the power level. Suppose there exists at least a pair of transmit power $\{p_1^{l_1}, p_2^{l_2}\}$ can make the PU and the SU meet their QoS requirements, where $p_1^{l_1} \subset \mathcal{P}$ and $p_2^{l_2} \subset \mathcal{P}$.

The PU and the SU work in non-cooperative ways. The transmit power of the PU is adjusted by its own power control policy. However, the SU is not aware of the power allocation of the PU. Obviously, the SU needs to obtain the information of the PU power allocation. In order to get the information, we utilize the received signal strengths (RSSs) for the set of spatially distributed wireless sensors. The RSSs are related to the transmit power of the PU and the SU, which reflects the channel state information. The SU can get the timely feedback information of the RSSs from the wireless sensors by conventional technologies [16].

We assume signal-to-interference-plus-noise ratios (SINR) can represent the QoS of the users. The SINR for the receiver i

at the kth slot is

$$\mathrm{SINR_i(k)} = \frac{|h_{ii}|^2 p_i(k)}{\sum\limits_{j\neq i} |h_{ij}|^2 p_j(k) + \sigma^2}, \tag{2}$$

where $i \in 1,2$, $j \in 1,2$, $p_1(k) \subset \mathcal{P}$ and $p_1(k) \subset \mathcal{P}$. The powers for the transmitter i and the transmitter j at the kth slot are $p_i(k)$ and $p_j(k)$, respectively. h_{ii} and h_{ij} represent the channel gains from the transmitter i to the receiver i and j, respectively. The noise variance of the system is σ_0^2.

For the PU, there are two power control schemes investigated. The PU allocates its transmit power relying on its power control scheme. In the first strategy, the formulation of the PU power update is given by

$$p_1(k) = f(\frac{\mathrm{SINR}_1^{\min} p_1(k-1)}{\mathrm{SINR}_1(k-1)}), \tag{3}$$

where $f(.)$ is a discretization operation which maps continuous powers into a set of discrete powers. $f(x)$ is equal to the nearest discrete power which means it is not more than x and $f(x) = p_1^l$ if $x > p_1^{l_1}$, where $l_1 \subset \{1, 2, \ldots, L\}$. The minimum QoS requirements for the PU and the SU are $\mathrm{SINR}_1^{\min}$ and $\mathrm{SINR}_2^{\min}$. In the second strategy, t is defined as

$$t = \frac{\mathrm{SINR}_1^{\min} p_1(k-1)}{\mathrm{SINR}_1(k-1)}. \tag{4}$$

The formulation of the PU power update is given by

$$p_1(k) = \begin{cases} p_1^{l_1+1} & p_1^{l_1} \leq t \leq p_1^{l_1+1}, l_1 + 1 \leq L \\ p_1^{l_1-1} & t \leq p_1^{l_1-1}, l_1 - 1 \geq 1 \\ p_1^{l_1} & other. \end{cases} \tag{5}$$

The predicted SINR of PU at the kth slot is defined as:

$$\mathrm{SINR}_1^{\mathrm{pre}}(k) = \frac{\mathrm{SINR}_1(k-1) p(k)}{p(k-1)}. \tag{6}$$

If $\mathrm{SINR}_1^{\min} \geq \mathrm{SINR}_1(k-1)$ and $\mathrm{SINR}_1^{\mathrm{pre}}(k) \geq \mathrm{SINR}_1^{\min}$, the power of PU should increase. If $\mathrm{SINR}_1^{\min} < \mathrm{SINR}_1(k-1)$ and $\mathrm{SINR}_1^{\mathrm{pre}}(k) \geq \mathrm{SINR}_1^{\min}$, the power of PU should decrease. Otherwise, the power of PU should stay the previous level.

For the SU, it needs to learn an effective power control scheme using the information of RSSs. The SU can satisfy its own QoS requirements after several rounds of power allocation adjustment. The wireless sensors are uniformly deployed between the PU and the SU. The set of wireless sensors is represented as $\mathcal{N} = \{1, 2, \ldots, N\}$ and the SU samples the RSSs information from the wireless sensors. The receive power at the kth slot for wireless sensor n is given by

$$P_n(k) = p_1(k) g_{1n} + p_2(k) g_{2n} + z_n(k), \tag{7}$$

where g_{1n} and g_{2n} are the path loss from wireless sensor n to primary transmitter and secondary transmitter, respectively. $z_n(k) \sim cN(0, \sigma^2)$ represents the shadowing effect and estimation errors for the system.

B. System Model Based on DRL Framework

The power control framework can be modeled as a markov decision process (MDP). *Proof*: Please refer to Appendix A. An agent-environment interaction modeling of the MDP is constructed, including an agent, an environment, state $\mathcal{S}$, action $\mathcal{A}$ and a reward function $\mathcal{R}$. In the model, the agent is SU and the environment consists of the PU and wireless sensors. The model is defined as below:

State Space: In the MDP model, the state of the environment is the RSSs of the wireless sensors. The size of state space is equal to the number of wireless sensors. The state of the MDP is shown as

$$S(k) = [P_1(k), P_2(k), \ldots, P_n(k)]. \tag{8}$$

Action Space: The action of the MDP agent is the power of the SU. The size of action space is equal to the number of the power level L. The action of the MDP can be given by

$$\mathcal{A} \subset \{a | a \in \{p_2^1, p_2^2, \ldots, p_2^L\}\}. \tag{9}$$

Reward: The reward of the system is related to whether the QoS of the PU and the SU are satisfied. Both of them satisfy QoS requirements that the reward is 10, otherwise the reward is 0. The reward function of the MDP is defined as

$$r(k) = \begin{cases} 10 & \mathrm{SINR_1(k+1)} \geq \mathrm{SINR}_1^{\min}, \\ & \mathrm{SINR_2(k+1)} \geq \mathrm{SINR}_2^{\min} \\ 0 & other \end{cases} \tag{10}$$

The observations of the RSS is random variation and the number of states is infinite. Therefore, it is not realistic for the SU to control power using Q-learning scheme. In order to handle this issue, the DNN is introduced in the RL framework. The action value function can be approximated using the DNN instead of using Q-table in the Q-learning. In value-based RL framework, the action value function is approximated by DNN. i.e., $Q^*(s, a) \approx Q(s, a; \theta)$, θ is the weights of the DNN. The parameters θ are updated according to minimizing the loss function, which is defined as

$$L_i(\theta_i) = \mathbb{E}(Q'(s(k), a(k); \theta_i) - Q(s(k), a(k); \theta_i))^2, \tag{11}$$

where $Q'(s(k), a(k); \theta_i)$ is the estimation value of the action. The action value function converges to the optimal action value function after taking some actions. For the Q-learning, the action-value function is estimated according to the Q-table for each state. The rows and columns separately represent the number of states and the number of actions. For the one-step Q-learning the $Q'(s(k), a(k); \theta_i)$ can be given by

$$\begin{aligned} &Q'(s(k), a(k); \theta_i) \\ &\quad = r(s(k), a(k)) \\ &\qquad + \gamma \max_{a(k+1)} Q(s(k+1), a(k+1); \theta_{i-1}). \end{aligned} \tag{12}$$

The drawback of one-step Q-learning is that the reward $r(k)$ only influences current state $s(k)$ and current action $a(k)$. Other states and actions are influenced through updating the action value $Q'(s(k), a(k); \theta_i)$. All the updates need to propagate the rewards to the relevant previous states and actions,

so the speed is quite slow. In the $n-$step Q-learning [18], the $Q'(s(k), a(k); \theta_i)$ is shown as.

$$Q'(s(k), a(k); \theta_i) = \sum_{i=0}^{n-1} \gamma^i r_{k+i} + \gamma^n \max_{a(k+n)} Q(s(k+n), a(k+n); \theta_{i-1}). \tag{13}$$

It can be seen from the formula that a single reward $r(k)$ can affect the values of n previous states and actions. Therefore, the rewards are propagated to relevant states and actions should be more effective in the $n-$step Q-learning. The optimal value function is the maximum action value

$$Q^*(s(k), a(k); \theta_i) = \max_{a(k)} Q(s(k), a(k); \theta_i). \tag{14}$$

In the training process, the SU constantly interacts with the wireless sensors and the PU. The SU chooses a power from the set of actions $\mathcal{A}$ with the largest action value $Q^*(s(k), a(k); \theta_i)$ at the time slot k. The wireless sensors and the PU receive the power allocation of the SU and update its next state $s(k+1)$ and provide reward $r(k)$ to the SU. The reward feedback mechanism can be used by the agent to learn about the optimal policy. The agent aims to maximize the cumulative rewards $R(k) = \sum_{k=0}^{\infty} r(k)$ in the process. This process is terminated until the total reward is not growing.

III. Asynchronous DRL Framework

There exists many DRL algorithms which can perform user power allocation. In this paper, there are two DRL frameworks that are asynchronous variant of actor-critic, such as, A3C-based power allocation scheme and DPPO-based power allocation scheme.

There are differences between asynchronous DRL framework and DQN. There exists a replay memory D in DQN algorithm that stores the transition $\{s(k), a(k), r(k), s(k+1)\}$. The training process will start once the replay memory has sufficient transition. The collected transitions are first placed in the replay memory, and then a minibatch transition is randomly selected from the replay memory for the training network. It breaks the association between the transitions and makes the transitions independent of each other. For asynchronous DRL algorithm, the parallel actor-learners learn power control scheme in different threads, breaking the interdependence of gradient updates. Therefore, the A3C and DPPO are online algorithms, while DQN is an offline value-based algorithm.

For our power control model, the asynchronous DRL framework with parallel computing is shown in Fig. 2. Unlike DQN, where a single SU represented by a single neural network (NN) interacts with a single PU and wireless sensors. The asynchronous DRL schemes utilize multiple incarnations of the above in order to learn more efficiently. From the Fig. 2, there exists a global network, and multiple SUs (agents) that each of them has their own network parameters. The SUs are distributed on different CPU threads of the same machine to learn the scheme. Each of SU interacts with it's own copy of the PU and wireless sensors (environment) at the same time as the other PUs are interacting with their environments. The power allocation experience of each SU is independent of the experience of the others. Therefore, the training of asynchronous DRL becomes more diverse and faster.

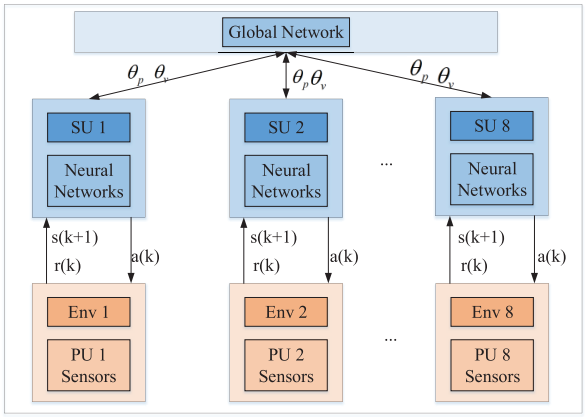

Fig. 2. Asynchronous DRL framework.

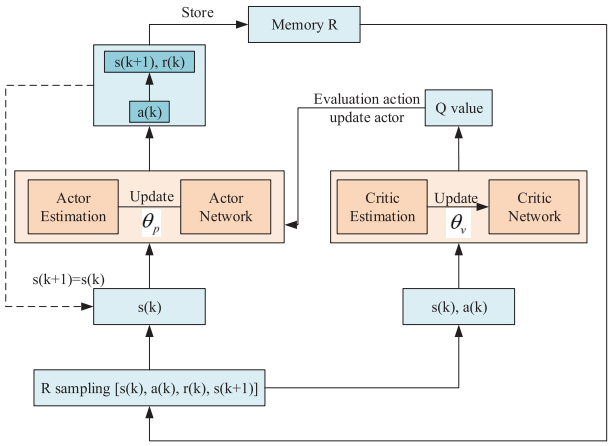

Fig. 3. Actor-Critic Mechanism.

There exists an actor-critic mechanism on each thread, which is shown in Fig. 3. The actor-critic mechanism combines the benefits of value-iteration methods and policy-iteration methods. For the actor network, the input is the state and the output is policy that is the probability of action. For the critic network, the output is the action value function that represents scores of all actions. The actor modifies the probability of an action based on the score of the critic. Importantly, the PU uses the value estimate (the critic) to update the power allocation policy (the actor) more intelligently than traditional policy gradient methods.

A. Asynchronous Advantage Actor-Critic

The A3C is a method for asynchronously parallel learning of multiple actor-learners on multiple CPU threads. The principle of RMSProp optimization is in Appendix B. Multiple actor-learners are distributed on different CPU threads in the same machine. The training method of A3C is similar to Hogwild [23] that reduces the cost of network parameters.

Another advantage of the method is that it can train the policy more reliably and with small resource requirements.

The update rule tells the agent which behavior is "good" and which behavior is "bad". This update network can properly encourage and discourage actions. Instead of using discounted returns update rule in policy gradient, the A3C method utilizes advantage estimates for updating. For update rule of A3C, the agent not only learns how good the action is, but also learn how much better than expected. The discounted returns as an estimate of action value function to allow us to generate an estimate of the advantage. The estimate of the advantage function is given by.

$$A(s(k),a(k);\theta_p,\theta_v)=\sum_{i=0}^{n-1}\gamma^i r_{k+i}+\gamma^n V(s(k+n);\theta_v) - V(s(k);\theta_v), \quad (15)$$

where $V(s(k+n);\theta_v)$ and $V(s(k);\theta_v)$ are the state value functions in the state $s(k+n)$ and $s(k)$, respectively.

All the actor-learners update the policy $\pi(a(k),s(k);\theta_p{}')$ and the state value function $V(s(k);\theta_v)$ according to gradient loss. The gradient loss function is regarded as gradient estimator that is given by

$$L(\theta_p{}')=\mathbb{E}[\nabla_{\theta_p{}'}\log\pi(a(k)|s(k);\theta_p{}')A(s(k),a(k);\theta_p,\theta_v)], \quad (16)$$

where θ_p and θ_v are the policy parameters of the actor network and action value parameters of the critic network. Both of them are updated until the terminal state is reached. $E_k[.]$ is the expectation which shows the empirical average in the batch of samples. The update formulas for θ_p and θ_v are

$$d\theta_p=d\theta_p+\nabla_{\theta_p{}'}\log\pi(a(k),s(k);\theta_p{}')(R-V(s(k);\theta_v{}')), \quad (17)$$

$$d\theta_v=d\theta_v+\nabla_{\theta_v{}'}(R-V(s(k);\theta_v{}')). \quad (18)$$

B. Proximal Policy Optimization Algorithms

The proximal policy optimization (PPO) algorithm is a family of the policy gradient methods. It optimizes the objective function. The principle of Adam optimization is in Appendix C. The traditional policy gradient method can only perform a gradient update on a single sample, and the PPO objective function can update multiple small batch gradients. It is easier to implement and more general.

The probability ratio is defined as

$$r_k(\theta_p{}')=\frac{\pi(a(k)|s(k);\theta_p{}')}{\pi(a(k)|s(k);\theta_p)}. \quad (19)$$

The clipped probability ratio that eliminates the motivation to move $r_k(\theta_p{}')$ outside the interval $[1-\varepsilon,1+\varepsilon]$. The clipped probability ratio is given by

$$clip(r_k(\theta_p{}'),1-\varepsilon,1+\varepsilon)A(s(k),a(k);\theta_p,\theta_v). \quad (20)$$

The gradient estimator with clipped objective function is given by

$$L^{\mathrm{CLIP}}(\theta_p{}') = \mathbb{E}[\min(r_k(\theta_p{}'),clip(r_k(\theta_p{}'),1-\varepsilon,1+\varepsilon))A(s(k),a(k);\theta_p,\theta_v)]. \quad (21)$$

Instead of using clipping for objective function, the penalty on KL divergence can be used in the objective function. The adaptive KL penalty coefficient is

$$\beta KL[\pi_{\theta_p}(.|s(k);\theta_p),\pi_{\theta_p{}'}(.|s(k);\theta_p{}')]. \quad (22)$$

The gradient estimator with KL-penalized objective function is shown as

$$L^{\mathrm{KLPEN}}(\theta_p{}') = \mathbb{E}[r_k(\theta_p{}')A(s(k),a(k);\theta_p,\theta_v)-\beta KL[\pi_{\theta_p},\pi_{\theta_p{}'}]], \quad (23)$$

where β is the coefficient for the policy update.

C. Algorithm Description

The A3C and DPPO methods are based on the actor-critic mechanism for the power control. The actor network optimizes power control policy to make spectrum sharing better. The critic network tries to estimate the value function to make spectrum sharing more accurate. The actor-critic mechanism is put in multiple threads for synchronous training. Each SU performs power control at the same time, and power control experience of SUs is simultaneously uploaded to a center. Then SUs obtain the latest power control strategy from the center. The center brings together the experience of each SU, and SU can obtain information from the center and use it in his own network. The center has global network and its parameters, each SU has a global network and a copy of local network, which can periodically push updates to global network, and then periodically from global network to get the comprehensive version of the update.

The A3C-based power control of the SU is given in Algorithm 1. The detailed steps of the Algorithm 1 are as follows. 1) We first initialize the global network parameters θ_p and θ_v and thread parameters $\theta_p{}'$ and $\theta_v{}'$. 2) Each SU interacts with wireless sensors and the PU. The SU gets $r(k)$ and $s(k+1)$ from environment and the environment obtains $a(k)$ from SU. The state is updated until the terminal $s(k)$ is reached or the number of iterations is greater than the maximum number of iterations on the thread. 3) The SU calculates update the policy $\pi(a(k),s(k);\theta_p{}')$ and the state value function $V(s(k);\theta_v)$. 4) The SU gets gradients $\theta_p{}'$ and $\theta_v{}'$ from the function (17) and (18), respectively. 5) The gradients $\theta_p{}'$, $\theta_v{}'$ on each thread are passed to the global network, and then the gradients of the global network θ_p, θ_v are updated. The gradients θ_p, θ_v of the global network are passed to each thread separately. 6) The network repeats the above steps until the number of iterations reaches maximum global shared counter $K_{\max}$. Finally, the SU learns an efficient power control. The QoS of SU can be satisfied according to adjusting its transmit power.

To further improve the efficiency of spectrum sharing, the DPPO-based power control of the SU is described in detail in Algorithm 2. 1) Similar to Algorithm 1, we initialize the global network parameters θ_p and θ_v and thread parameters $\theta_p{}'$ and $\theta_v{}'$. 2) Each actor-learner collects K time slots of data. The policy $\pi(a(k)|s(k);\theta_p{}')$ is obtained from actor and the state value function $V(s(k),\theta_v{}')$ and an estimation of advantage $A(s(k),a(k);\theta_p,\theta_v)$ are obtained from critic. 3) On

Algorithm 1 A3C-Based for Power Control

Initialize the global parameters θ_p and θ_v and thread parameters $\theta_p{}'$ and $\theta_v{}'$.
Initialize global shared counter $K = 0$.
Initialize maximum global shared counter $K_{\max}$ and maximum thread counter $k_{\max}$.
repeat
  Reset the global gradients: $d\theta_p = 0, d\theta_v = 0$.
  Reset the thread parameters: $\theta_p{}' = \theta_p, \theta_v{}' = \theta_v$.
  Initialize thread counter $k = 0$.
  $t_{start} = t$.
  Get the RSSs of the sensors $s(k)$.
  repeat
    Get power $a(k)$ by policy $\pi(a(k)|s(k);\theta_p{}')$.
    Get reward $r(k)$ and next state $s(k+1)$.
    $k = k + 1, K = K + 1$.
  until terminal $s(k)$ or $k_{\max} > k - k_{start}$.
  $R = \begin{cases} 0 & terminal\ s(k) \\ V(s(k),\theta_v{}') & non-terminal\ s(k) \end{cases}$.
  Update $\pi(a(k)|s(k);\theta_p{}')$.
  while $k < k_{start}$ **do**
    $R = r(k) + \gamma R$.
    Update thread parameters $\theta_p{}'$ using (17).
    Update thread parameters $\theta_v{}'$ using (18).
  end while
  Asynchronous update global parameters θ_p according to (17).
  Asynchronous update global parameters θ_v according to (18).
until $K \geq K_{\max}$.

each thread, the thread parameters $\theta_p{}'$ and $\theta_v{}'$ are updated according to minimizing the gradient estimator with clipped objective function (21) or the KL-penalized objective function (23). 4) The global network can get all threads parameters $\theta_p{}'$ and $\theta_v{}'$, and the global parameters gradients θ_p, θ_v are passed to each thread separately. In this way, the overall experience available for training becomes more diverse, assisting the SU makes better decision. The Algorithm 2 eliminates excessive changes in SU power values and also makes network functions converge faster.

The A3C-based and DPPO-based power control schemes both outperform existing DQN-based method. There are main two reasons. Firstly, the A3C-based and DPPO-based methods are based on actor-critic mechanism and DQN method is based on Q-learning method. The actor-critic mechanism combines the benefits of value-iteration methods and policy-iteration methods. The action can be made by the actor network and action values are produced by the critic network. Both of them separate action selection from value estimation, avoiding overestimation of value. However, Q-learning belongs to value-iteration method. Secondly, multiple agents are considered in the A3C-based and DPPO-based schemes and single agent is investigated in the DQN-method scheme. The parallel agents learn the strategies in different threads and they can obtain the learned information of each other from the global network, breaking the interdependence of gradient updates.

Algorithm 2 DPPO-Based for Power Control

Initialize the global parameters θ_p and θ_v and thread parameters $\theta_p{}'$ and $\theta_v{}'$.
Initialize global shared counter $K = 0$.
Initialize maximum global shared counter $K_{\max}$.
repeat
  for $thread = 1$ to N **do**
    for $k = 1$ to $k_{\max}$ **do**
      Get $a(k)$ by policy $\pi(a(k)|s(k);\theta_p{}')$.
      Get advantage estimates $A(s(k), a(k); \theta_p, \theta_v)$.
    end for
    Update thread parameters $\theta_p{}'$ and $\theta_v{}'$ according to minimizing gradient estimator (21) or (23).
  end for
  Update global parameters θ_p and θ_v.
until $K \geq K_{\max}$

IV. Simulation Results

A. Simulation Setup

In this section, simulations verify the performance of the A3C-based power control algorithm and the DPPO-based power control algorithm scheme. For the simulation, the PU adopts second strategy to update the power. The transmit power of SUs is selected from the set of power P. We set the noise power for receivers to 0.01 W, the minimum QoS requirements for the PU and the SU are set to 1.2 bps/Hz and 0.7 bps/Hz. The wireless sensors are evenly distributed in a circle with a radius of 300 m centered on the transmitters. The Adam optimizer and the RMSProp optimizer are applied for updating DPPO network and A3C network, respectively. The proportion of success is the ratio of the number of successful trials to the total number of trials. Assuming that the agent adjusts the SU power to the target state within 20 time slots, the test trial considered successful. The average exploration step is the average time slot required to reach the target state when the test is successful. The Hyperparameters for DQN, A3C and DPPO are shown as

The proposed scheme is performed in Python 3.5 with TensorFlow 1.8.0 with Intel(R) Core(TM)i7-7700@3.6GHz, NVIDIA GeForce GTX 1050.

Fig. 4 shows the total reward versus the number of iteration with DPPO, A3C and DQN. It can be observed that the DPPO-based power control scheme can obtain higher reward than A3C scheme and DQN scheme. The proposed DPPO-based algorithm converges to the maximum value of the reward within 100 iterations. In addition, the A3C-based algorithm has better performance than the DQN-based algorithm for power control. As expected, DPPO and A3C are better than DQN power control since they reduce the strong correlation and gradient dependence of neural network value estimation.

Fig. 5 demonstrates the average exploration step versus the number of iteration with DPPO, A3C and DQN. The average

TABLE I
DRL HYPERPARAMETERS OF SYSTEM

Parameter	Value
Learning rate of DQN	0.0001
Learning rate of A3C critic network	0.0001
Learning rate of A3C actor network	0.001
Learning rate of DPPO critic network	0.0001
Learning rate of DPPO actor network	0.0001
Initial exploration	0.8
Final exploration	0
Discount rate	0.8
Decay rate	0.01
Replay memory	400
Input dimension	10
Output dimension	8
Hidden layer 1 of DQN	256
Hidden layer 2 of DQN	256
Hidden layer 3 of DQN	512
Hidden layer of A3C actor network	200
Hidden layer of A3C critic network	100
Hidden layer of DPPO actor network	200
Batch size	256
Hidden layer of DPPO critic network	200

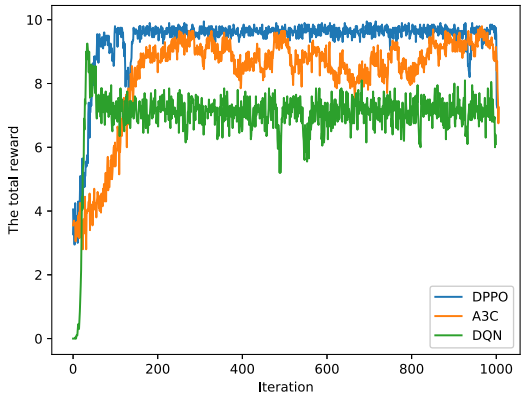

Fig. 4. The total reward vs. iteration.

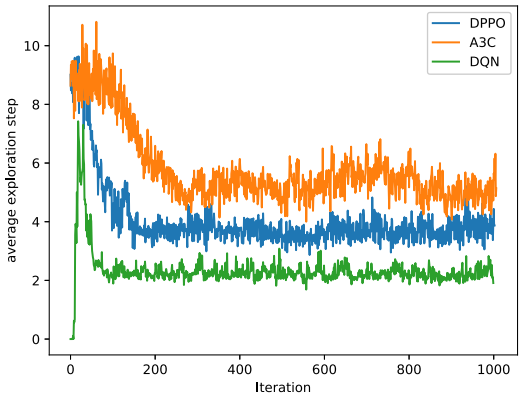

Fig. 5. Average exploration step vs. iteration for different Algorithms.

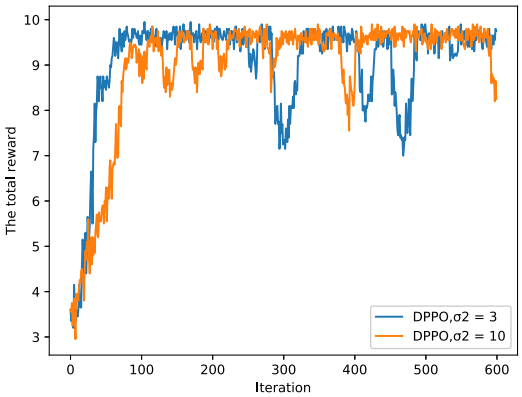

Fig. 6. The total reward vs. iteration for different σ^2 for DPPO.

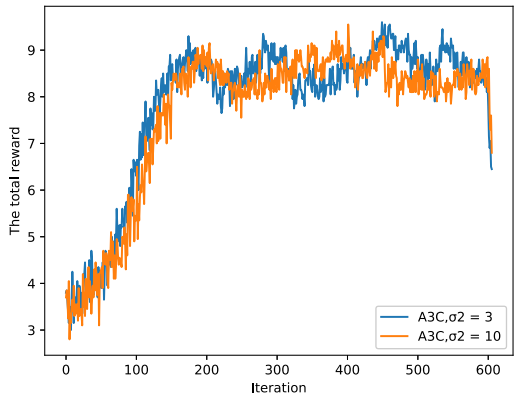

Fig. 7. The total reward vs. iteration for different σ^2 for A3C.

exploration step reflects the average time slot required to reach the target state when the trail is successful. From Fig. 5, when the number of iteration within 300, the average exploration step is close to a steady level. Besides, the average exploration step of the three algorithms is less than 6. Therefore, the agent can quickly adjust the power of the SU to the target power.

Fig. 6 depicts the total reward versus the number of iteration for different noise power σ^2 in the receivers for the DPPO algorithm. Fig. 7 depicts the total reward versus the number of iteration for different noise power σ^2 in the receivers for the A3C algorithm. For the DPPO algorithm and the A3C algorithm, when noise power is 3 or 10, their reward values are very close.

Fig. 8 demonstrates the total reward versus the number of iteration for different number of state for DPPO. Fig. 9 demonstrates the total reward versus the number of iteration for different number of state for A3C. When the state number of the model uses different values, such as 5 or 10, the system return value is still very close for the DPPO algorithm and the A3C algorithm. Therefore, the proposed two algorithms are very robust, and the variables and state changes in the model will hardly affect the performance of the system.

Fig. 10 shows the average exploration step versus the number of iteration for DPPO scheme with different set of actions A. The action of the agent is the power of the SU. The dimension of the action is the number of power values that the SU can adjust. The larger the action dimension, the more average exploration steps the network needs to reach the target power. Besides, the smaller action dimension, the faster the DPPO network converges. Fig. 11 shows the average exploration step versus the number of iteration for

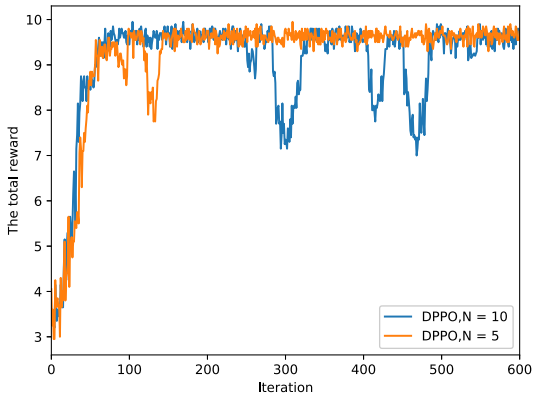

Fig. 8. The total reward vs. iteration for different $\mathcal{S}$ for DPPO.

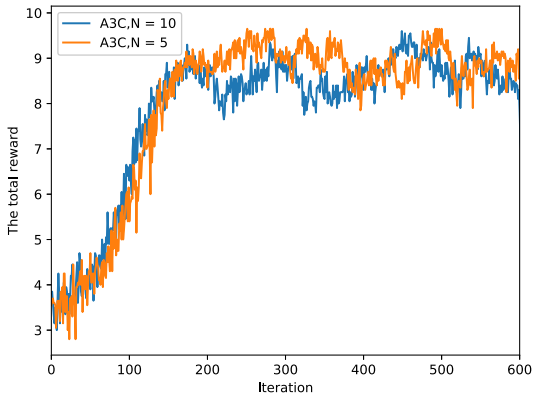

Fig. 9. The total reward vs. iteration for different $\mathcal{S}$ for A3C.

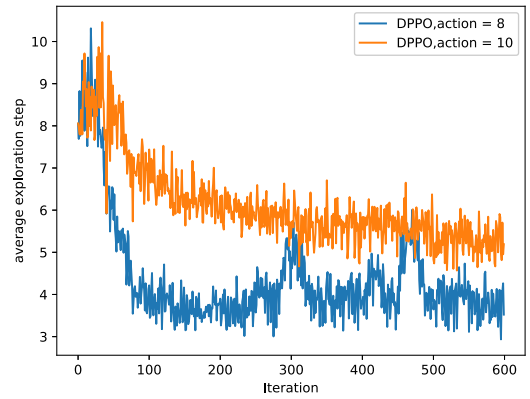

Fig. 10. Average exploration step vs. iteration for different $\mathcal{A}$.

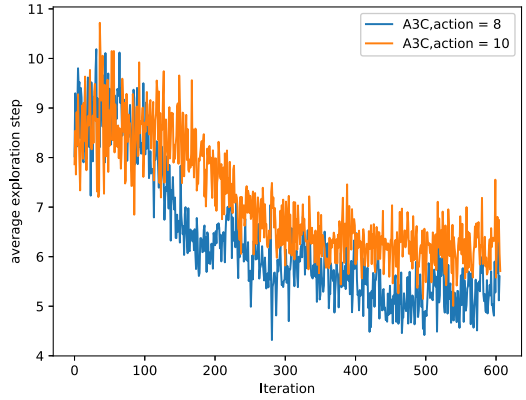

Fig. 11. Average exploration step vs. iteration for different $\mathcal{A}$.

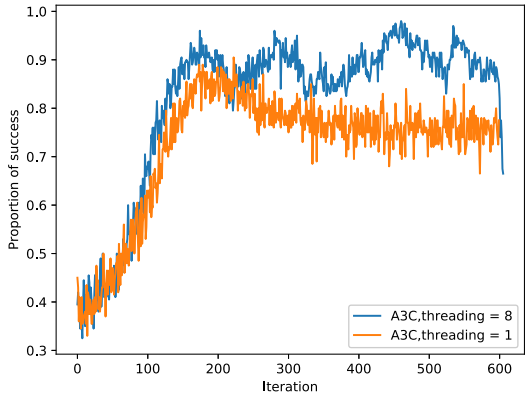

Fig. 12. Average exploration step vs. iteration for different threads.

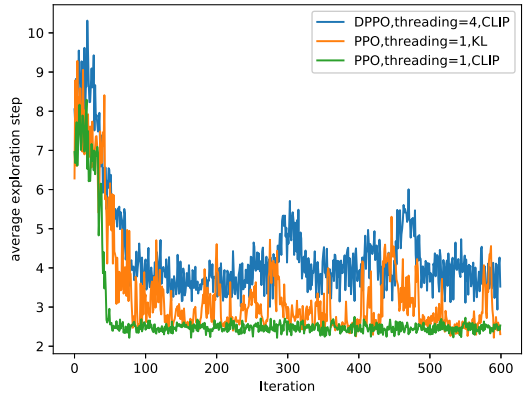

Fig. 13. Proportion of success vs. iteration for different threads.

A3C scheme with different set of actions A. Similar to Fig. 10, the more power values a SU can choose, the more exploration steps the A3C network needs.

Fig. 12 shows the proportion of success versus the number of iteration for A3C scheme with different number of thread. All threads learn the power control strategy in parallel and then pass the learned parameters to the global network. The global network issues instructions to each thread based on the data of all threads. Therefore, multiple threads have a higher success rate than a single thread. Normally the number of the thread is, the better the neural network learns.

Fig. 13 shows the average exploration step versus the number of iteration for PPO scheme with different number of thread and different gradient estimator. In PPO-based power control with single thread, the thread parameters $\theta_p{}'$ and $\theta_v{}'$ are updated according to minimizing the gradient estimator with clipped objective function and the KL-penalized objective function. For single thread PPO, both of them converge to the same value almost, and the average exploration step is less than 3 times. However, the gradient estimator with clipped objective function converges faster and more stably than the gradient estimator with clipped objective function.

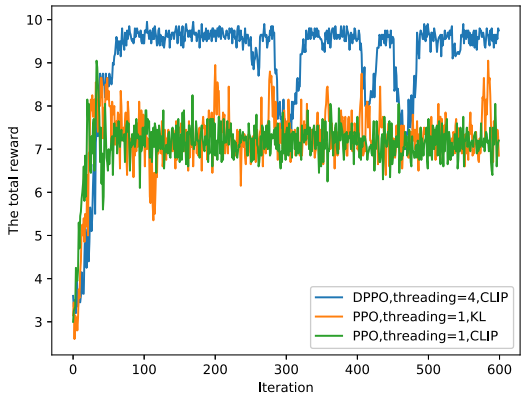

Fig. 14. The total reward vs. iteration for different threads.

Comparing DPPO-based power control with four threads and PPO-based power control with single thread, DPPO method requires more average exploration steps than PPO method.

Fig. 14 shows the total reward versus the number of iteration for PPO scheme with different number of threads and different gradient estimators. The PPO-based power control with clipped objective function and the PPO-based power control with KL-penalized objective function converge to the same performance in system reward. Similar to Fig. 13, multiple threads have better system reward than single threads.

V. Conclusion

The spectrum sharing problem was investigated in a cognitive radio system consisting of a PU, a SU and wireless sensors, according to adjusting their power. The PU and the SU work in a non-cooperative way that PU cannot obtain the power allocation about the PU. The PU adjusts its transmit power relying on its power control scheme. The A3C-based power control and DPPO-based power control were proposed for the SU to learn how to adjust its transmit power. Both of the schemes were asynchronous variant of actor-critic with policy-based and value-based methods. Finally, the system can meet the QoS of PU and SU. The results showed that the proposed schemes have better performance than the DQN-based power allocation in power efficiency and network convergence.

Appendix A

Proof: The power control framework can be modeled as a MDP.

The SU selects an action $a(k) = p_2(k+1)$ under the state $s(k)$, the environment gets the next state $s(k+1)$. The next state $s(k+1)$ only depends on the action $a(k)$ and the current state $s(k)$ and is not relate to previous states and actions. Therefore, the power control framework can be modeled as a MDP.

Appendix B

Proof: The principle of RMSProp optimization

In the DRL-based asynchronous framework, RMSProp optimization and Adma optimization algorithms are used to learn power control scheme. The RMSProp algorithm uses a differential squared weighted average for the gradients of weight W and bias b, which is represented as

$$W = W - \alpha \frac{dW}{\sqrt{g_{dw}} + \varepsilon}$$
$$b = b - \alpha \frac{db}{\sqrt{g_{db}} + \varepsilon}. \tag{24}$$

The RMSProp algorithm calculates the differential squared weighted average for the gradient. g_{dw} and g_{db} are gradient momentum which are accumulated by loss function. gradient momentum g_{dw} and g_{db} are given by

$$g_{dw} = \beta g_{dw} + (1-\beta) dW^2$$
$$g_{db} = \beta g_{db} + (1-\beta) db^2. \tag{25}$$

β is gradient accumulation index.

Appendix C

Proof: The principle of Adam optimization

Adam optimization not only calculates the first-order moment of the gradient, but also makes full use of the second-order moment. Specifically, it calculates the exponential moving average of the gradient, and the hyperparameters β_1 and β_2 control the decay rate of these moving averages. We can first calculate the parameter updates for first-order moment and second-order moment of gradient:

$$c_{dw} = \beta_1 g_{dw} + (1-\beta_1) dW$$
$$c_{db} = \beta_1 g_{db} + (1-\beta_1) db. \tag{26}$$
$$g_{dw} = \beta_2 g_{dw} + (1-\beta_2) dW^2$$
$$g_{db} = \beta_2 g_{db} + (1-\beta_2) db^2. \tag{27}$$

The average of the moving index is very different from the initial value at the beginning of the iteration, so the deviations of the several values obtained above are corrected as follows

$$c_{dw}^r = \frac{c_{dw}}{1-\beta_1^i}$$
$$c_{db}^r = \frac{c_{db}}{1-\beta_1^i}$$
$$g_{dw}^r = \frac{g_{dw}}{1-\beta_2^i}$$
$$g_{db}^r = \frac{g_{db}}{1-\beta_2^i}. \tag{28}$$

Next, the update of weights and bias are represented as.

$$W = W - \alpha \frac{c_{dw}^r}{\sqrt{g_{dw}^r} + \varepsilon}$$
$$b = b - \alpha \frac{c_{db}^r}{\sqrt{g_{db}^r} + \varepsilon}. \tag{29}$$

References

[1] P. Liu, S. R. Chaudhry, T. Huang, X. Wang, and M. Collier, "Multi-factorial energy aware resource management in edge networks," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 45–56, Mar. 2019.

[2] L. A. Fletscher, L. A. Suarez, D. Grace, C. V. Peroni, and J. M. Maestre, "Energy-aware resource management in heterogeneous cellular networks with hybrid energy sources," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 279–293, Mar. 2019.

[3] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
[4] B. Zheng *et al.*, "Design of multi-carrier LBT for LAA&WiFi coexistence in unlicensed spectrum," *IEEE Netw.*, vol. 34, no. 1, pp. 76–83, Jan. 2020.
[5] M. Z. Khan, S. Harous, S. U. Hassan, M. U. G. Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72622–72633, 2019.
[6] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu, "Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1938–1948, Apr. 2019.
[7] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning based multi-access control with energy harvesting," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.
[8] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.
[9] V.-D. Nguyen and O.-S. Shin, "Cooperative prediction-and-sensing-based spectrum sharing in cognitive radio networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 1, pp. 108–120, Mar. 2018.
[10] Z. Xiao *et al.*, "Spectrum resource sharing in heterogeneous vehicular networks: A noncooperative game-theoretic approach with correlated equilibrium," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9449–9458, Oct. 2018.
[11] I. Mitliagkas, N. D. Sidiropoulos, and A. Swami, "Joint power and admission control for ad-hoc and cognitive underlay networks: Convex approximation and distributed implementation," *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4110–4121, Dec. 2011.
[12] P. Wang, J. Fang, N. Han, and H. Li, "Multiantenna-assisted spectrum sensing for cognitive radio," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1791–1800, May 2010.
[13] F. Fang, H. Zhang, J. Cheng, and V. C. M. Leung, "Energy-efficient resource allocation for downlink non-orthogonal multiple access network," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3722–3732, Sep. 2016.
[14] H. Zhang, N. Yang, K. Long, M. Pan, G. K. Karagiannidis, and V. C. M. Leung, "Secure communications in NOMA system: Subcarrier assignment and power allocation," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 7, pp. 1441–1452, Jul. 2018.
[15] H. Zhang, S. Huang, C. Jiang, K. Long, V. C. M. Leung, and H. V. Poor, "Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1936–1947, Sep. 2017.
[16] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.
[17] S. A. Grandhi, J. Zander, and R. Yates, "Constrained power control," *Wireless Pers. Commun.*, vol. 1, no. 4, pp. 257–270, Dec. 1994.
[18] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 1989.
[19] H. Zhang, H. Liu, J. Cheng, and V. C. M. Leung, "Downlink energy efficiency of power allocation and wireless backhaul bandwidth allocation in heterogeneous small cell networks," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1705–1716, Apr. 2018.
[20] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[21] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1928–1937.
[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347
[23] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 693–701.
[24] T. Tijmen and H. Geoffrey, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *Neural Netw. Mach. Learn.*, vol. 4, pp. 26–31, Oct. 2012.
[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015, pp. 1–15.
[26] Y. He *et al.*, "Deep-Reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.
[27] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2375–2385, Aug. 2018.
[28] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.
[29] N. Liu *et al.*, "A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Atlanta, GA, USA, Jun. 2017, pp. 372–382.
[30] Z. Xu, J. Tang, C. Yin, Y. Wang, and G. Xue, "Experience-driven congestion control: When multi-path TCP meets deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1325–1336, Jun. 2019.
[31] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for Internet of Things: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2074–2083, Apr. 2019.
[32] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

Haijun Zhang (Senior Member, IEEE) is currently a Full Professor with the University of Science and Technology Beijing, China. He received the IEEE CSIM Technical Committee Best Journal Paper Award, in 2018, the IEEE ComSoc Young Author Best Paper Award, in 2017, and the IEEE ComSoc Asia–Pacific Best Young Researcher Award, in 2019. He serves as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS NETWORKING, and the IEEE COMMUNICATIONS LETTERS.

Ning Yang is currently pursuing the Ph.D. degree with the University of Science and Technology Beijing, China. Since September 2019, she has been visiting the Communications and Networking Lab, Northwestern University, Chicago, IL, USA, as a Visiting Research Associate. Her research interests are in resource management based on deep reinforcement learning in wireless communication. More specifically, she focuses on developing analytic techniques in stochastic modeling, reinforcement learning, and game theory, optimization, and applying these techniques to wireless resource management and optimization.

Wei Huangfu received the M.S. and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1998 and 2001, respectively. He is currently an Associate Professor with the School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB). His main research interests include statistical signal processing, cooperative communications, the Internet of Things, and wireless sensor network.

Keping Long (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), in 1995 and 1998, respectively. From July 2001 to November 2002, he was a Research Fellow at the ARC Special Research Centre for Ultra Broadband Information Networks (CUBIN), University of Melbourne, Australia. He is currently a Professor and Dean of the School of Computer and Communication Engineering (CCE), USTB. He is/was a member of the Editorial Committee of Sciences in China Series F and China Communications. He is/was also a TPC and ISC member for COIN, IEEE IWCN, ICON, and APOC, and the Organizing Co-Chair of IWCMC'06, the TPC Chair of COIN'05/'08, and the TPC Co-Chair of COIN'08/'10. He was awarded the National Science Fund Award for Distinguished Young Scholars of China in 2007 and selected as the Chang Jiang Scholars Program Professor of China in 2008.

Victor C. M. Leung (Fellow, IEEE) is currently a Distinguished Professor of computer science and software engineering with Shenzhen University. He is also an Emeritus Professor of electrical and computer engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems, The University of British Columbia (UBC). His research interests are in the broad areas of wireless networks and mobile systems. He has coauthored more than 1300 journals/conference papers and book chapters. He received the IEEE Vancouver Section Centennial Award, the 2011 UBC Killam Research Prize, the 2017 Canadian Award for Telecommunications Research, and the 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award. He has coauthored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the 2017 IEEE Systems Journal Best Paper Award, the 2018 IEEE CSIM Best Journal Paper Award, and the 2019 IEEE TCGCC Best Journal Paper Award. He is a fellow of the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of "Highly Cited Researchers". He is serving on the editorial boards of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE ACCESS, IEEE NETWORK, and several other journals.