Deep Q-Network Based Power Allocation Meets Reservoir Computing in Distributed Dynamic Spectrum Access Networks

Hao Song, Lingjia Liu, Hao-Hsuan Chang, Jonathan Ashdown, and Yang Yi

***Abstract*—Dynamic spectrum access (DSA) is regarded as one of the key enabling technologies for future communication networks. In this paper, we introduce a power allocation strategy for distributed DSA networks using a powerful machine learning tool, namely deep reinforcement learning. The introduced power allocation strategy enables DSA users to conduct power allocation in a distributed fashion without relying on channel state information and cooperations among DSA users. Furthermore, to capture the temporal correlation of the underlying DSA network environments, the reservoir computing, a special class of recurrent neural network, is employed to realize the introduced deep reinforcement learning scheme. The combination of reservoir computing and deep reinforcement learning significantly improves the efficiency of the introduced resource allocation scheme. Simulation evaluations are conducted to demonstrate the effectiveness of the introduced power allocation strategy.**

I. Introduction

It is predicted that mobile traffic will increase sevenfold between 2016 and 2021 with a compound annual growth rate (CAGR) of 46% [1]. This explosive growth in mobile traffic imposes a huge challenge to future mobile broadband networks searching for methods to increase the underlying spectrum utilization for mobile communications. Meanwhile, various experimental tests and measurements reveal the fact that much licensed spectrum is under-utilized. This motivated Federal Communication Commission (FCC) to re-investigate spectrum access related techniques and introduced the concept of dynamic spectrum access (DSA) to enhance spectrum utilization [2].

To support DSA, many frequency bands are opened up for unlicensed spectrum access, such as industrial, scientific and medical (ISM) bands. LTE-Advanced is also extended to the 5.8 GHz ISM band through licensed-assisted access (LAA) [3]. The over utilization caused mainly by Wi-Fi access makes these bands extremely congested. To cope with that, the FCC further exploits under-utilized licensed frequency bands for DSA opportunities. For example, in 2015, the FCC held an auction on the licenses of AWS-3 bands, including 1695-1710 MHz, 1755-1780 MHz, and 2155-2180 MHz bands. Besides, 3.5 GHz bands (3550-3700 MHz) will also be opened up for DSA [4].

Despite tremendous benefits of using opened licensed bands, many research challenges have to be addressed first. In most of opened licensed bands, there would exist primary users (PUs) with high priorities, which are normally incumbents on licensed bands and should be protected from detrimental interference. For example, according to relevant policies, the DSA network operators that attempt to access AWS-3 bands need to treat federal systems, like the federal Meteorological-Satellite (MetSat) systems, as PUs [4]. On the other hand, DSA users may be of heterogenous nature with limited cooperations among themselves. In this case, global channel state information (CSI) may be unavailable.

In this paper, we investigate resource allocations of DSA networks by introducing efficient and robust wireless resource management strategies, including spectrum access and power allocation. Our earlier work [5] introduces a framework of an artificial intelligence-enabled spectrum access strategy using Deep Q-network (DQN) and reservoir computing (RC). The introduced resource allocation strategy focuses on spectrum access without considering the underlying power allocation. To further address the challenges and improve performance, we focus on studying power allocation strategies for distributed DSA networks in this work.

II. System Model

In DSA, multiple DSA users share spectrum resources with PUs. Without loss of generality, we assume each DSA user consists of a DSA pair: a transmitter

H. Song, L. Liu, H. Chang and Y. Yi are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech., Blacksburg, VA, USA, 26041. J. Ashdown is with the Information Directorate, Air Force Research Laboratory (AFRL), Rome, NY, 13441, USA. This work was supported in part by the U.S. National Science Foundation (NSF) under Grants NSF/ECCS-1802710, NSF/ECCS-1811497, and NSF/CNS-1811720. Approved for public release (reference number: 88ABW-2019-0210).

978-1-7281-1878-9/19/$31.00 ©2019 IEEE

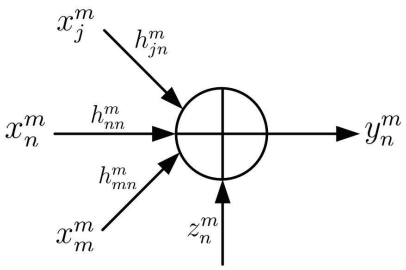

Fig. 1. Received signals of a DSA user.

and a receiver. Furthermore, for simplification we assume that each PU occupies one particular wireless channel. For DSA users, we consider the general case where DSA users simultaneously utilize multiple channels to conduct transmissions. In this way, different DSA users may access the same channel causing interference among themselves. In this paper we focus on power allocations assuming channel allocations are given. Notations adopted in this paper are the following: $\mathbf{N} = \{n | n = 1, 2, \cdots, N\}^T$ represents the set of DSA users. We use $\mathbf{M} = \{m | m = 1, 2, \cdots, M\}^T$ to represent the sets of PUs and wireless channels associated with corresponding PUs. $\mathbf{\Omega}_n = \{m | m = 1, 2, \cdots, M_n\}^T$ and $\mathbf{\Phi}_n = \{n | n = 1, 2, \cdots, N_m\}^T$ denote the set of channels allocated to DSA user n and the set of users sharing channel m, respectively.

In DSA networks, DSA users receives interference from other DSA users and PUs, as shown in Fig. 1. The received signal of DSA user n on channel m is

$$y_n^m = x_n^m \cdot h_{nn}^m + x_m^m \cdot h_{mn}^m + \sum_{j \in \mathbf{\Phi}_m, j \neq n} x_j^m \cdot h_{jn}^m + z_n^m,$$

where x_n^m denotes the desired signal of DSA user n. x_m^m and x_j^m stand for interference caused by PU m and DSA user j, respectively. Accordingly, h_{nn}^m, h_{mn}^m, and h_{jn}^m represent channel gains of links from the transmitter to the receiver of DSA user n, from PU m to DSA user n, and from DSA user j to DSA user n, respectively. z_n^m is the received additive white Gaussian noise (AWGN).

The corresponding signal to interference-plus-noise-ratio (SINR) can be expressed as

$$r_n^m = \frac{p_n^m \cdot |h_{nn}^m|^2}{\underbrace{p_m^m \cdot |h_{mn}^m|^2}_{\substack{Interference\ from \\ PU\ m}} + \underbrace{\sum_{j \in \mathbf{\Phi}_m, j \neq n} p_j^m \cdot \left|h_{jn}^m\right|^2}_{\substack{Interference\ from \\ other\ DSA\ users}} + \underbrace{B \cdot N_0}_{noise}},$$

where p_n^m, p_m^m, and p_j^m denote transmit power of n, m, and j on channel m. B and N_0 are channel bandwidth and noise spectral density, respectively.

Due to the lack of cooperations, each DSA user can only obtain the CSI of the link between its own transmitter and receiver. For DSA networks, it is important to protect PUs from harmful interference. Therefore, PUs should provide basic feedback on the received interference to facilitate DSA users to adjust their transmit power. As shown in Fig. 2, it is assumed that each PU is capable of detecting its interference and is able to feedback the interference to DSA users.

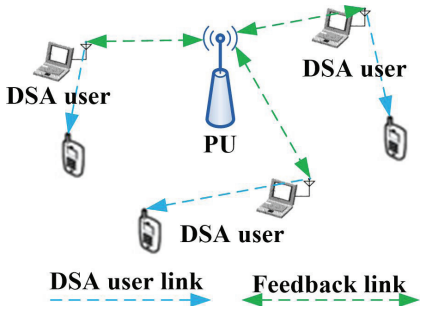

Fig. 2. Feedback specific interference level to each DSA user.

III. Deep Q-network based Distributed Power Allocation Scheme

Without centralized control, DSA users have to make the decision of power allocations individually through limited CSI. In this paper, we utilize reinforcement learning (RL) and deep neural network (DNN) to enable intelligent distributed power allocations.

A. Reinforcement learning

Reinforcement learning is a promising machine learning paradigm. With reinforcement learning, agents are able to learn which actions should be taken to yield the maximum reward without relying on labels, the acknowledged correct actions provided by authoritative external supervisor. Instead, agents need to try a variety of actions to accumulate the knowledge of rewards. Furthermore, each action should be tried many times to obtain the reward knowledge associated with different states. By exploiting accumulated reward knowledge, agents will take the actions that are expected to bring in maximum rewards [6].

Q-learning is a type of reinforcement learning, which is widely used in various applications due to its model-free nature. The model-free nature makes agents learn optimal action policy directly through interacting with environments rather than investigating environment models, such as transition probability. Since Q-learning uses

iterative approach to update Q-value of each state and each action, a big challenge that has to be addressed is the tradeoff between exploitation and exploration. For exploitation, it is better for agents to select actions that has been tried and found to be optimal for the current state, aiming at gaining high reward. The optimal policy π^* could be employed to guarantee exploitation:

$$A_t^* = \arg \max_{A_t} Q^{\pi^*}(S_t, A_t) \quad (1)$$

where S_t and A_t represent an initial state and the action selected following the optimal policy π^*.

However, to obtain higher reward in the future, agents need to try actions that have not been experienced to accumulate reward knowledge, so-called exploration. The exploration in Q-learning would be more important and meaningful in distributed DSA networks with dynamic environments, enabling Q-values to keep updating to adapt to the variations of wireless environments. In this paper, the ε-greedy method is applied to take into account both exploitation and exploration, where $\varepsilon \in [0, 1]$ is the probability that agents randomly select actions regardless of Q-values. The corresponding policy used for action selections is shown as

$$A_t^* = \begin{cases} \arg\max_{A_t} Q^{\pi^*}(S_t, A_t), \text{ with the probability of } 1-\varepsilon, \\ \text{Randomly select actions, with the probability of } \varepsilon, \end{cases}$$

Accordingly, an online Q-value update method is adopted, which is defined by

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \cdot \left[R_{t+1} + \gamma \cdot \max_{A_{t+1}} Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

where R_{t+1}, $\alpha \in (0, 1)$, and $\gamma \in [0, 1]$ denote the obtained reward, the learning rate, and the discounted rate, respectively. It is noticeable that γ could be deemed as a factor to adjust the weights of immediate rewards and future rewards. If the future reward is considered to be more important than the immediate reward, γ should be set to a relatively large value.

B. Deep Q-network for distributed power allocations

A DQN based distributed power allocation scheme is introduced where powerful DNNs are employed to efficiently perform Q-learning especially for large-scale model. For the feasibility of using DQN in power allocations, basic configurations of Q-learning need to be designed, which is described as the following:

1) Each DSA user possesses independent DNNs to perform Q-learning, including updating Q-values and fulfilling action selections based on the policy.

2) The state in the Q-table of DSA user n is defined as a transmit power vector expressed by $S = \left(p_1, p_2, \cdots, p_{|\mathbf{\Omega}_n|}\right)^T$, where p_i, $i = 1, 2, \cdots, |\mathbf{\Omega}_n|$, denotes the transmit power on i^{th} channel, and $|\mathbf{\Omega}_n|$ is the number of the channels used by DSA user n. To limit the scale of the Q-table, the transmit power should be discretized properly. For example, if the total transmit power constraint of a user is 300 mW, its transmit power on one channel could be discretized into 4 levels: 0 mW, 100 mW, 200 mW, and 300 mW.

3) The action in the Q-table is defined as a vector, indicating transmit power adjustments of all channels used by a DSA user. The vector can be expressed as $A = \left(a_1, a_2, \cdots, a_{|\mathbf{\Omega}_n|}\right)^T$, where a_i, $i = 1, 2, \cdots, |\mathbf{\Omega}_n|$, stands for the transmit power change of i^{th} channel. Considering the scale of the Q-table, the number of possible actions should be restricted, therefore, we only consider 3 kinds of transmit power adjustments: increasing transmit power to the next level, decreasing transmit power to the next level, and no change, represented by In, De, and Un, respectively.

Based on the aforementioned configurations, a design example of a DSA user's Q-table is given as shown in Table I under the condition of 2 wireless channels, 300 mW transmit power constraint, and 4 transmit power levels. It should be noticed that when a user is in a specific state, some actions should not be chosen. For example, at state 8 (100 mW, 200 mW), taking actions 5, 6, and 9 will make the transmit power exceed the maximum power constraint. As for state 2 (100 mW,0 mW), actions 2, 4, 8 should not be chosen, since they will make the transmit power become negative. To resolve these issues, an operating mechanism is designed that the state will kept the same if taking an action will make the transmit power vector out of the scope defined in the Q-table. For example, if the initial state is state 7 (200 mW, 100 mW) while action 6 is chosen, in this case the transmit power vector will turn to (300 mW, 100 mW), which is not defined in the Q-table. As a result, state 7 will be kept.

From Table I, it is easy to see that the size of Q-table will exponentially increase with the growth of the number of channels and the number of transmit power levels. A large Q-table will make it hard or even impossible to train [7]. Thus, DNNs are utilized to provide efficient Q-learning operations, so-called deep Q-network [8]. An iteration of DQN is depicted in Fig. 3. There are two neural networks (NNs) in DQN. The first one, named Evaluated NN (ENN), is used to generate Q-values of the initial state S_t on each action, $Q(S_t, A1), Q(S_t, A2), \cdots, Q(S_t, AL)$, where L is the total number of the actions defined in Q-table. An action A_t is selected based on generated Q-values and a predefined policy. After taking action A_t in environments, the corresponding reward and the next

TABLE I
Q-TABLE IN A DSA USER

	$A1:(Un,Un)$	$A2:(Un,De)$	$A3:(De,Un)$	$A4:(De,De)$	$A5:(Un,In)$	$A6:(In,Un)$	$A7:(De,In)$	$A8:(In,De)$	$A9:(In,In)$
$S1:(0mW,0mW)$	$Q(S1,A1)$	$Q(S1,A2)$	$Q(S1,A3)$	$Q(S1,A4)$	$Q(S1,A5)$	$Q(S1,A6)$	$Q(S1,A7)$	$Q(S1,A8)$	$Q(S1,A9)$
$S2:(100mW,0mW)$	$Q(S2,A1)$	$Q(S2,A2)$	$Q(S2,A3)$	$Q(S2,A4)$	$Q(S2,A5)$	$Q(S2,A6)$	$Q(S2,A7)$	$Q(S2,A8)$	$Q(S2,A9)$
$S3:(0mW,100mW)$	$Q(S3,A1)$	$Q(S3,A2)$	$Q(S3,A3)$	$Q(S3,A4)$	$Q(S3,A5)$	$Q(S3,A6)$	$Q(S3,A7)$	$Q(S3,A8)$	$Q(S3,A9)$
$S4:(100mW,100mW)$	$Q(S4,A1)$	$Q(S4,A2)$	$Q(S4,A3)$	$Q(S4,A4)$	$Q(S4,A5)$	$Q(S4,A6)$	$Q(S4,A7)$	$Q(S4,A8)$	$Q(S4,A9)$
$S5:(200mW,0mW)$	$Q(S5,A1)$	$Q(S5,A2)$	$Q(S5,S3)$	$Q(S5,A4)$	$Q(S5,A5)$	$Q(S5,A6)$	$Q(S5,A7)$	$Q(S5,A8)$	$Q(S5,A9)$
$S6:(0mW,200mW)$	$Q(S6,A1)$	$Q(S6,A2)$	$Q(S6,A3)$	$Q(S6,A4)$	$Q(S6,A5)$	$Q(S6,A6)$	$Q(S6,A7)$	$Q(S6,A8)$	$Q(S5,A9)$
$S7:(200mW,100mW)$	$Q(S7,A1)$	$Q(S7,A2)$	$Q(S7,A3)$	$Q(S7,A4)$	$Q(S7,A5)$	$Q(S7,A6)$	$Q(S7,A7)$	$Q(S7,A8)$	$Q(S7,A9)$
$S8:(100mW,200mW)$	$Q(S8,A1)$	$Q(S8,A2)$	$Q(S8,A3)$	$Q(S8,A4)$	$Q(S8,A5)$	$Q(S8,A6)$	$Q(S8,A7)$	$Q(S8,A8)$	$Q(S8,A9)$
$S9:(300mW,0mW)$	$Q(S9,A1)$	$Q(S9,A2)$	$Q(S9,A3)$	$Q(S9,A4)$	$Q(S9,A5)$	$Q(S9,A6)$	$Q(S9,A7)$	$Q(S9,A8)$	$Q(S9,A9)$
$S10:(0mW,300mW)$	$Q(S10,A1)$	$Q(S10,A2)$	$Q(S10,A3)$	$Q(S10,A4)$	$Q(S10,A5)$	$Q(S10,A6)$	$Q(S10,A7)$	$Q(S10,A8)$	$Q(S10,A9)$

state S_{t+1} could be obtained. Then, the next state S_{t+1} will be input to another neural network, named Target NN (TNN), to output Q-values that correspond to S_{t+1}, $Q(S_{t+1},A1),Q(S_{t+1},A2),\cdots,Q(S_{t+1},AL)$. Based on the Q-values generated by TNN and the obtained reward, the $Q(S_t,A_t)$ is updated which will be used as the target value to train the ENN by the back propagation method. After multiple iterations, the ENN will be adopted as a new TNN to replace original one.

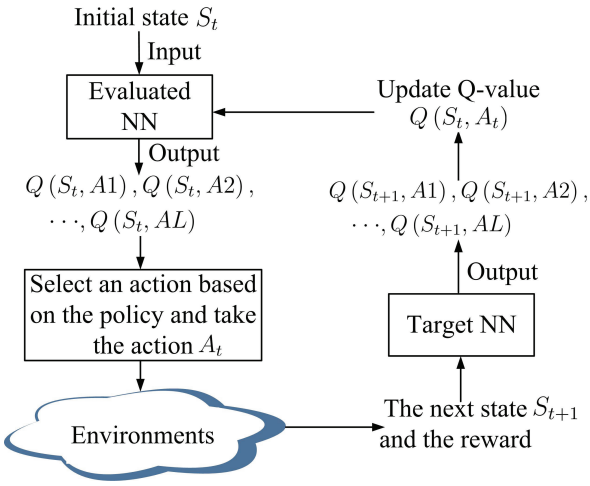

Fig. 3. An iteration of deep Q-network.

Under the condition of no centralized control, the reward of DSA user n is defined as

$$R_n=\sum_{m\in\mathbf{\Omega}_n}\log_2\left(1+\frac{|h_{nn}^m|^2\cdot p_n^m}{|h_{mn}^m|^2\cdot p_m^m+\sum\limits_{j\in\mathbf{\Phi}_m,j\neq n}|h_{jn}^m|^2\cdot p_j^m+B\cdot N_0}\right)-\kappa\cdot\sum_{m\in\mathbf{\Omega}_n}e^{\frac{I_n^m}{\Delta I}} \quad (2)$$

where I_n^m and ΔI stand for the interference strength received by PU m that comes from DSA user n, and a reference interference level, respectively. In (2), the first term is the achievable spectral efficiency of DSA user n on channel m, while the second term represents the penalty regarding interference caused to PUs. Apparently, ΔI could be viewed as a threshold. Once the interference suffered by PU m exceeds the threshold, the reward will exponentially decrease with the growth of I_n^m. κ is a weight to adjust the impact of penalty on the reward. From (2), it can be seen that the only feedback that DSA users need is I_n^m, since a DSA user can monitor its achievable spectral efficiency by itself without depending on any feedback from other users.

C. Selection of Neural Networks

A key factor to directly determine the performance of DQN is the selection of neural networks. The Feed-Forward Neural Network (FFNN) is widely used in diverse applications because of its characteristics of simple structure and being easy to train. However, in distributed DSA networks, the Recurrent Neural Network (RNN) may be a better choice to capture the dynamic of wireless environments. This is because the activation update in RNN needs to take into account not only current input data, but also the previous activations of recurrent neurons and output neurons. These feedback connections make RNN capable of learning temporal correlations in dynamic systems. For example, a typical application of RNN is the natural language processing, since understanding a sentence normally needs to consider previous sentences, namely temporal correlations. Similar to the natural language processing, temporal correlations also exist in the variations of wireless environments, since most of wireless devices adjust their transmission parameters following a fixed protocol, like the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) in Wi-Fi systems. Unfortunately, compared to FFNN, the training of the RNN has been proven to be very difficult [7]. This barrier makes the application of RNN

TABLE II
SIMULATION PARAMETERS.

Parameters	Values
Transmit power of PUs	400mW
Transmit power constraint of DSA users	300mW
Channel bandwidth B	2MHz
Noise spectral density N_0	-174dBm/Hz
Center frequency	5GHz
Path-loss model (WINNER II)	$41 + 22.7 \cdot \log_{10}(d[m])$ $+20 \cdot \log_{10}(f_c[GHz]/5)$
K-factor	8
Penalty weight κ	0.5
Reference interference level ΔI	10^{-4}mW

in distributed DSA networks more challenging. Thus, a special type of RNN, Reservoir Computing (RC), will be studied and utilized to construct DQN in this paper. RC can be deemed as a simplified RNN, in which only the weights of the output layer will be trained, while other weights in the input layer and the reservoir layers are generated randomly and fixed in the training process [9]. By this way, the difficulty of training RNN could be significantly alleviated.

IV. SIMULATION RESULTS AND ANALYSIS

By conducting simulation studies, the effectiveness of our proposed distributed power allocation strategy will be demonstrated. Furthermore, the convergence of RC based DQN used in distributed DSA network scenarios will be investigated. We consider a distributed DSA network with $M = 2$ wireless channels and $N = 4$ DSA users. On each channel, a PU keeps occupying it and transmitting data. Both PUs and DSA users are randomly distributed in an region of a 150m $\times$ 150m square. Moreover, the Rician channel model and the WINNER II channel model are adopted to calculate channel gains [10]. According to the aforementioned analysis, the tradeoff between exploration and exploitation is a critical factor for Q-learning which should be considered. In simulation studies, the total number of training is 14000. In first 4000 times training, the ε in the ε-greedy method is set to be a relatively large value, namely 0.5, allowing DSA users to fully explore and investigate all the possible power allocation strategies. After that, ε will be adjusted to be 0, under which DSA users attend to select the power allocation strategy able to bring in the optimal reward. The detailed simulation parameters are presented in Table II.

In simulations, Q-learning and FFNN based DQN will be employed as referred methods to testify the effectiveness and convergence of RC based DQN. Q-learning has been widely studied to improve the performance of DSA networks [11]. Unfortunately, Q-learning is not capable

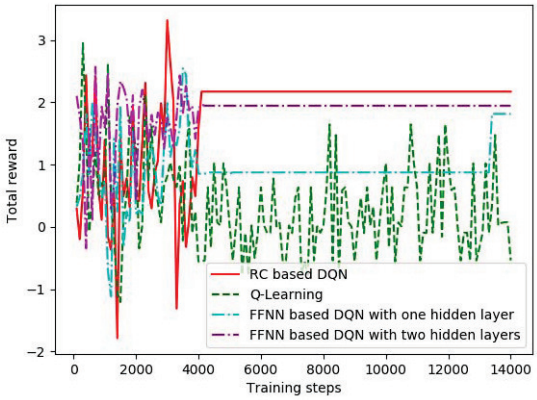

Fig. 4. Total reward versus training steps.

of handling the large size of states and actions. When the number of states and actions become large, it is very hard for Q-learning to converge. By comparing with Q-learning, we will investigate whether RC based DQN has preferable convergence performance. Because of training-friendly feature, FFNN is widely used in diverse applications. We will compare the performance of RC based DQN with that of FFNN based DQN to verify whether the temporal correlation nature of RC is able to improve performance in distributed DSA networks.

In Fig. 4, the total reward of all the DSA users is illustrated versus training steps. It is apparent that RC based DQN has excellent convergence behaviors, which can converge fast. The choice of the learning rate is crucial for convergence speed. Here, the learning rate is set to be 0.01, and all the methods use the same learning rate. Another obvious observation is that with RC based DQN our proposed distributed power allocation strategies could let DSA users obtain higher reward. According to the reward definition in (2), higher reward indicates that DSA users attain higher spectral efficiency and PUs suffer from lower interference.

The same phenomenon can be seen in Fig. 5, which presents the total data rate of all the DSA users with the unit of Mbits/s versus training steps. In distributed DSA networks, DSA users will compete with each other to gain more data rate. In such an environment, raising transmit power may be able to let a DSA user enhance data rate in a short term. However, after a while, the DSA user may suffer from more severe interference, as higher transmit power will also cause higher interference to other DSA users that will boost their transmit power to preserve communication quality as well. Hence, distributed power allocation strategies should be able to facilitate DSA users to reach a balance on transmit power to allow each DSA user to get a preferable performance.

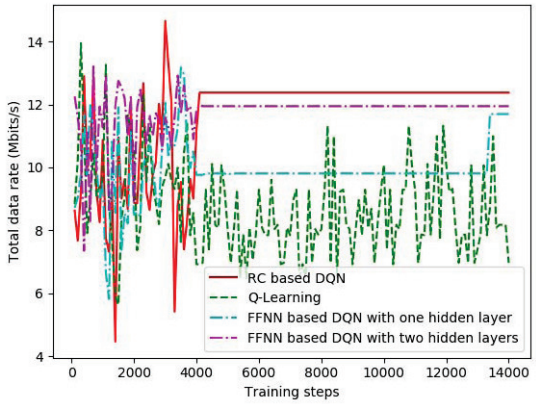

Fig. 5. Total data rate versus training steps.

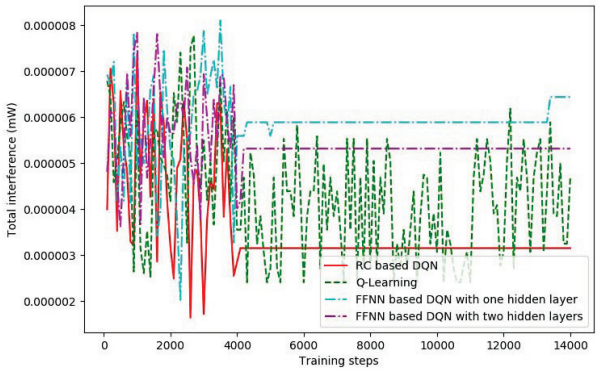

Fig. 6. Total interference versus training steps.

From both Fig. 5 and Fig. 6, it is obvious that our proposed power allocation strategy enable DSA users to reach a balance quickly. Besides, compared to other methods, our proposed method can make distributed DSA networks have relatively higher data rate, proving better data rate enhancement performance.

Another important performance indicator is the interference caused by DSA users, which is investigated in Fig. 6 by showing the total interference suffered by PUs versus training steps. Apparently, our proposed method is able to effectively and promptly restrain the interference caused to PUs in a relatively low level. Since the interference is treated as a penalty in the reward defined in (2), DSA users are encouraged to choose power allocation strategies that tend to lower the interference. Additionally, with the assistance of the powerful RC, DSA users are able to learn wireless environments better and make more appropriate decisions on power allocations to protect PUs from detrimental interference.

V. Conclusion

In this paper, we study resource allocation techniques that could be effectively utilized in distributed DSA networks. Firstly, technical challenges that may encounter in distributed DSA networks are analyzed. To tackle those challenges, a power allocation strategy based on reinforcement learning is proposed for intelligent distributed power allocations. However, typical reinforcement learning technologies, like Q-learning, cannot handle the large size of states and actions. In other words, if the number of channels and DSA users become large, reinforcement learning is very hard to be trained, causing instability. Thus, the reservoir computing, a type of recurrent neural network, is used to realize deep reinforcement learning for efficient operations. Moreover, the temporal correlation nature of reservoir computing could enable DSA users to accurately learn wireless environments variations and properly carry out distributed power allocations. The extensive simulation study indicates that our proposed power allocation strategy has excellent convergence behaviors. Moreover, the simulation results demonstrate that our proposed power allocation strategy could achieve better performance on both data rate enhancement and PUs protection.

References

[1] CISCO Whitepaper, "CISCO Visual Networks Index: Global Mobile Data Traffic Forecast Update, 2016-2021," Feb 2017.

[2] Federal Communications Commission, "Spectrum policy task force," Rep. ET Docket 02-135, Nov. 2002.

[3] H. Song, X. Fang, L. Yan, and Y. Fang, "Control/User Plane Decoupled Architecture Utilizing Unlicensed Bands in LTE Systems," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 132-142, October 2017.

[4] S. Bhattarai, J. J. Park, B. Gao, K. Bian, and W. Lehr, "An Overview of Dynamic Spectrum Sharing: Ongoing Initiatives, Challenges, and a Roadmap for Future Research," *IEEE Trans. Cognitive Commun. and Netw.*, vol. 2, no. 2, pp. 110-128, June 2016.

[5] 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 IoT J.*, (Early Access), Sept 2018.

[6] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *The MIT press*, Nov 2017.

[7] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *ICML*, pp. 1310-1318, Feb 2013

[8] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *ISCA*, pp. 338-342, Sep 2014.

[9] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127-149, 2009.

[10] P. Kyosti, "WINNER II channel models," D1.1.2, V1.2, Sep. 2007.

[11] N. Morozs, T. Clarke, and D. Grace, "Distributed Heuristically Accelerated Q-Learning for Robust Cognitive Spectrum Management in LTE Cellular Systems," *IEEE Trans. on Mobile Comput.*, vol. 15, no. 4, pp. 817-825, April 2016.