

Anti-jamming Communications Using Spectrum Waterfall: A Deep Reinforcement Learning Approach

Xin Liu, Yuhua Xu, *Member, IEEE*, Luliang Jia, *Student Member, IEEE*, Qihui Wu, *Senior Member, IEEE*, and Alagan Anpalagan, *Senior Member, IEEE*

Abstract—This letter investigates the problem of anti-jamming communications in dynamic and unknown environment through on-line learning. Different from existing studies which need to know (estimate) the jamming patterns and parameters, we use the spectrum waterfall, i.e., the raw spectrum environment, directly. Firstly, to cope with the challenge of infinite state of raw spectrum information, a deep anti-jamming Q-network is constructed. Then, a deep anti-jamming reinforcement learning algorithm is proposed to obtain the optimal anti-jamming strategies. Finally, simulation results validate the proposed approach. The proposed approach is relying only on the local observed information and does not need to estimate the jamming patterns and parameters, which implies that it can be widely used various anti-jamming scenarios.

Index Terms—Anti-jamming, Deep Q-Network, Deep Reinforcement Learning

I. INTRODUCTION

Anti-jamming is always an active research topic, as wireless transmissions are naturally vulnerable to jamming attacks. The mainstream anti-jamming techniques includes Frequency Hopping Spread Spectrum (FHSS) and Direct-Sequence Spread Spectrum (DSSS) [1]. Recently, to address the interactions between the legitimate users and the jammers, game theory has been widely applied in the literature [2]–[7]. In methodology, these approaches need to know the jamming strategies, which implies that the legitimate users are required to estimate the jamming patterns and parameters from the observed environment. However, with the rapid development of artificial intelligence and universal software radio peripheral (USRP) [8], the jammers can easily create dynamic and intelligent jamming attacks. As a consequence, there are two limitations with regard to estimation-based anti-jamming communications: i) there may be information loss for unknown jamming patterns, and ii) if the intelligent jammer switches its strategies dynamically and rapidly, it is not possible to track and react it in real time. Thus, it is challenging and interesting to investigate anti-

jamming communication approaches in dynamic and unknown environment.

To overcome the above limitations, a promising way is to design new anti-jamming approaches that utilize the raw environmental information, which is known as spectrum waterfall [9], without estimating jamming patterns and parameters. These kind of anti-jamming approaches would avoid information loss and adapt to the dynamic environment, as can be expected. In addition, online learning is an effective way to solve the decision problems in dynamic environment. The widely used technique is Q-learning [10], which has been used in anti-jamming problems [2], [3]. Unfortunately, Q-learning is not able to deal with the raw environmental information directly because of the infinite state of the environment.

Motivated by the deep reinforcement learning technique for learning successful control policies from raw video data in [11], we investigate the anti-jamming problem in unknown and dynamic environment. First, the raw spectrum information is defined as the state of the environment to avoid losing the jammer information as much as possible. Then, a deep anti-jamming Q-network (DAQN) is constructed to realize the direct processing of raw spectrum information. Finally, a deep anti-jamming reinforcement learning algorithm (DARLA) is proposed. Simulation results show that the proposed DARLA achieves the best anti-jamming strategies in various scenarios. The main contributions are summarized as follows.

- Based on the deep reinforcement learning technique, a smart anti-jamming communication scheme is proposed. In particular, the raw spectrum information is defined as a state, which describes the detail features of jammer more accurately.
- The proposed algorithm is relying only on the locally observed information and does not need to estimate the jamming patterns and parameters the jammer in advance, i.e., it is model-free, which can be widely used in various anti-jamming scenarios.

Note that the most related work is [12], which also adopted deep reinforcement learning to investigate the anti-jamming problems. The main differences in this work are as follows: i) the environment state is presented by extracting features of signal-to-interference-plus-noise ratio (SINR) and primary user occupancy in [12], while it is presented by the raw spectrum information in this work, and ii) it requires the jammer to have the same channel-slot transmission structure

X. Liu is with the College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China. (e-mail:leo_nanjing@126.com).

Y. Xu and L. Jia are with the College of Communication Engineering, PLA Army Engineering University, Nanjing 210007, China. (e-mail: yuhuaenator@gmail.com;jiallts@163.com).

Qihui Wu is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China (e-mail: wuqihui2014@sina.com).

Alagan Anpalagan is with the Department of Electrical and Computer Engineering, Ryerson University, Toronto, Canada (alagan@ee.ryerson.ca).

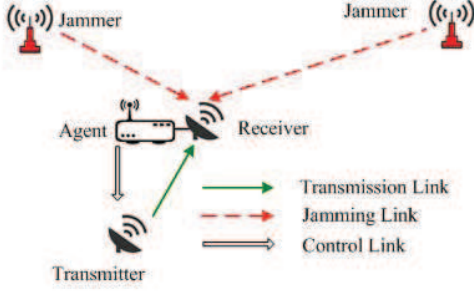


Fig. 1. System model.

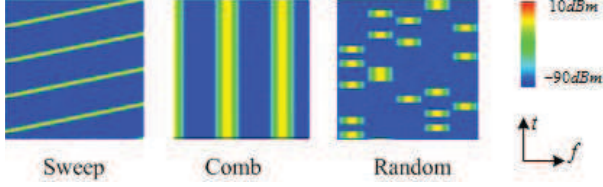


Fig. 2. Thermodynamic chart of various jamming pattern.

with the users in [12]. On the contrary, this requirement does not hold in our work, which makes the proposed approach more general.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider the transmission of one user (a transmitter-receiver pair) against one or several jammers, as shown in Fig. 1. The agent, which is disposed at the receiving end, sends anti-jamming strategies to the transmitter through a reliable control link. Jammers may adopt fixed, random, or possibly intelligent jamming patterns. However, we do not analyze the specific jamming models, but obtain the optimal anti-jamming strategies based on the raw spectrum information.

While the receiver receives the desired signal, the agent continuously senses the whole communication bands and stores the sensed values. Denote the spectrum vector as $\mathbf{P}_t = \{p_{t,1}, p_{t,2}, \dots, p_{t,N}\}$, where $p_{t,n}$ is the power of frequency n at time t and N is the number of sampling points in frequency space. In order to sufficiently use history spectrum information, a two-dimensional matrix, which describes time-frequency features of spectrum environment, is expressed as:

$$\mathbf{S}_t = \begin{bmatrix} \mathbf{P}_{t-1} \\ \mathbf{P}_{t-2} \\ \vdots \\ \mathbf{P}_{t-M} \end{bmatrix} = \begin{bmatrix} p_{t-1,1} & p_{t-1,2} & \cdots & p_{t-1,N} \\ p_{t-2,1} & p_{t-2,2} & \cdots & p_{t-2,N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{t-M,1} & p_{t-M,2} & \cdots & p_{t-M,N} \end{bmatrix}. \quad (1)$$

It is noted that \mathbf{S}_t contains all the spectrum information until time t , as M tends to infinity. However, the difficulty of the decision optimization problem is significantly increased with the increase of M . Therefore, M can take an appropriate value, which would be determined by the time-varying characteristics of the spectrum environment.

To illustrate the rationality of using \mathbf{S}_t as the basis of anti-jamming decision-making, we give the thermodynamic charts of the \mathbf{S}_t matrix of several common jamming patterns, also

known as spectrum waterfall [9], as shown in Fig. 2. Taking the swept jamming as example, we can accurately determine the frequency range and intensity (color) of jamming at the next moment by looking at the thermal chart, which also means we can determine the anti-jamming strategy accordingly.

In the unknown and dynamic environment, we do not consider estimation-based anti-jamming strategies. Instead, define \mathbf{S}_t as the environment state, and then consider a dynamic decision problem in which the agent (anti-jamming user) interacts with an environment through a sequence of observations of environment (\mathbf{S}_t), actions (a_t) and rewards (r_t). Specifically, an action a can be a combination decisions of frequency, power, coding schemes, spread spectrum, and other kinds of anti-jamming decisions, e.g., $a = (f, p)$ represents the combination actions of frequency (f) and power (p). The rewards associated with the actions and environment is defined as:

$$r(a, \mathbf{S}) = \begin{cases} R(a) - \lambda\delta & \beta(a, \mathbf{S}) \geq \beta_{th}(a) \\ 0 & \beta(a, \mathbf{S}) < \beta_{th}(a) \end{cases}, \quad (2)$$

where $R(a)$ is the bit rate when the action a is selected, λ is the cost when action changes, δ is an indication of action change ($\delta = 1$ if $a_t \neq a_{t-1}$; $\delta = 0$ if $a_t = a_{t-1}$), $\beta(a, \mathbf{S})$ is the received signal to interference plus noise ratio (SINR) in state \mathbf{S} with action a . $\beta_{th}(a)$ is the required SINR threshold for successful transmission. Note that $R(a)$ and $\beta_{th}(a)$ are modeled as a function of action a , the reason is as follows: the bit rate and SINR requirements change for different anti-jamming strategies, such as forward error correction and spread spectrum schemes.

Then, the goal of the agent is to select anti-jamming actions in a fashion that maximizes cumulative future reward $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1}$, where γ is the discount factor. One way of achieving this goal is to compute the following optimal action-value (also known as Q) function [10]:

$$Q^*(\mathbf{S}, a) = \max_{\pi} E \{R_t \mid \mathbf{S}_t = \mathbf{S}, a_t = a, \pi\}, \quad (3)$$

where the anti-jamming policy $\pi = P(a \mid \mathbf{S})$ refers to a probability distribution over the actions. Based on the Bellman equation,

$$Q^*(\mathbf{S}, a) = E \left\{ r + \gamma \max_{a'} Q^*(\mathbf{S}', a') \mid \mathbf{S}, a \right\}. \quad (4)$$

III. ANTI-JAMMING COMMUNICATION SCHEME

The traditional Q-learning is unable to cope with the anti-jamming problem described in section II, as the state space size of \mathbf{S}_t is almost infinite. In order to solve this problem, a deep anti-jamming Q-network (DAQN) is constructed to address the interactive decision-making problem with raw spectrum information input, which contains decision network and update network, as shown in Fig. 3 and Fig. 4 respectively. We use a deep convolutional neural network (CNN) to approximate the optimal action-value as shown in decision network, where the input state \mathbf{S}_t is represented by a thermal chart of $M \times N$ pixels. After the processing of two convolutional layers and two fully connected layers, the output is the estimated Q function, where K is the size of action space. At last, the

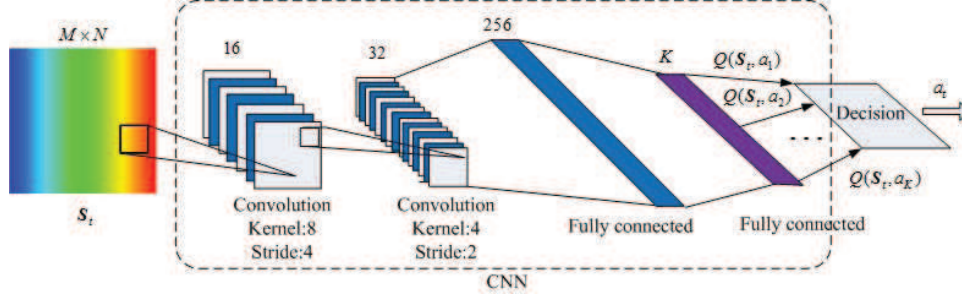


Fig. 3. Decision network of the DAQN.

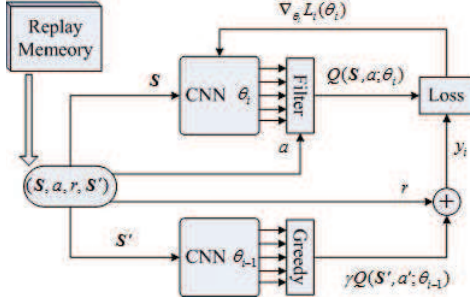


Fig. 4. Updating network of the DAQN.

decision layer outputs the corresponding action based on the estimated Q function.

However, reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator [11], such as the neural network is used to represent the Q function. The main reason is correlation during the learning process. The idea of experience replay is adopted to address these instabilities as shown in the update network. To perform experience replay, we store the agent's experiences $e_t = (S_t, a_t, r_t, S_{t+1})$ at each time-step t in data set $D_t = (e_1, \dots, e_t)$. When the experience pool is big enough, we construct target values $r + \gamma \max_{a'} Q(S', a'; \theta_{i-1})$ by randomly choosing elements in a uniform distribution $(S, a, r, S') \sim U(D)$, which reduces the correlation during sequential observation. The Q -learning update at iteration i uses the following loss function:

$$L_i(\theta_i) = E_{(S,a,r,S') \sim U(D)} \left[(y_i - Q(S, a; \theta_i))^2 \right], \quad (5)$$

where θ_i is the parameter of Q -network at iteration i and $y_i = r + \gamma \max_{a'} Q(S', a'; \theta_{i-1})$ is target value computed by Q -network parameter θ_{i-1} with greedy strategy. By assuming that y_i is the expected output of CNN with network weight θ_i when the input is S , we calculate the difference between real output $Q(S, a; \theta_i)$ and target value y_i to determine the update of network parameters. Differentiating the loss function with respect to the weights, we arrive at the following gradient:

$$\nabla_{\theta_i} L_i(\theta_i) = E_{(S,a,r,S')} [(y_i - Q(S, a; \theta_i)) \nabla_{\theta_i} Q(S, a; \theta_i)]. \quad (6)$$

According to the gradient descent algorithm, the network weight θ_i is updated according to (6). Although there are two

Algorithm 1: Deep Anti-jamming Reinforcement Learning Algorithm (DARLA)

Initialize : Set $D = \emptyset$, $\epsilon = 1$, Set θ with random weights, Sense initial environment S_1 .

For $t = 1, T$ **do**

With probability ϵ , select a random action a_t

otherwise, select $a_t = \arg \max Q(S_t, a; \theta)$

Execute action a_t and compute r_t and observe S_{t+1}

Store transitions (S_t, a, r, S_{t+1}) in D

If $Sizeof(D) > \mathcal{N}$ (Enough amount of transitions)

Sample random minibatch of transitions

(S, a, r, S') from D

Compute $y_i = r + \gamma \max_{a'} Q(S', a'; \theta)$

Compute gradient based on Eq.(6) and Update θ

End If

Calculate $\epsilon = \max(0.1, \epsilon - \Delta\epsilon)$

End For

CNN networks with different weights, as shown in Fig. 4, the actual implementation requires only one CNN network, as the computing of target values and the updating of network weights are in different stages. The algorithm for anti-jamming communication based on deep reinforcement learning is presented in Algorithm 1.

IV. NUMERICAL RESULTS AND DISCUSSIONS

In the simulation setting, the user and the jammer combat with each other in a frequency band of 20MHz, where the frequency resolution of spectrum sensing is 100kHz. The user performs a full band sensing every 1ms and retains the spectrum data within the 200ms. Hence, the size of matrix S_t is 200×200 . The bandwidth of user signal is 4MHz, and the center frequency is allowed to change in each 10ms with the step of 2MHz, which means $K = 9$. Both signal and jamming are raised cosine waveform with roll-off factor $\alpha = 0.3$, in which jamming power is 30dBm and signal power is 0dBm. The demodulation threshold β_{th} at all frequency is set to be 10dB, and the cost of action change λ is set to be $0.2R(a)$.

Four kinds of jamming patterns are given for simulation: i) Sweep jamming (sweep speed is 1GHz/s); ii) Comb jamming (three fixed frequency signals at 2MHz, 10MHz, and 18MHz);

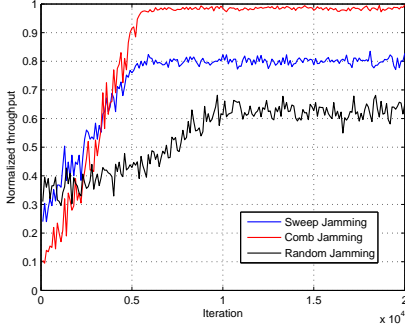


Fig. 5. Normalized throughput under different jamming patterns.

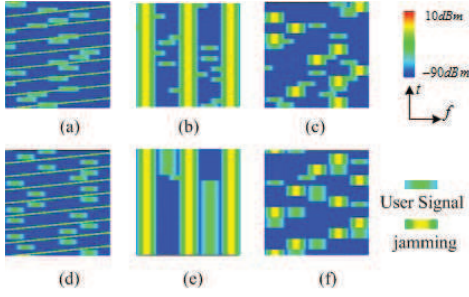


Fig. 6. Environmental states at initial and convergent stages under different jamming patterns.

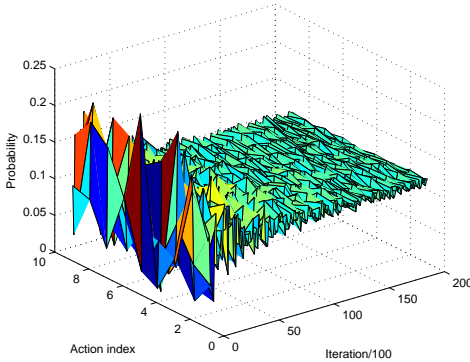


Fig. 7. Probability of user actions during learning.

iii) Random jamming (frequency is randomly changed every 20ms with the step of 4MHz); iv) Intelligent jamming (the jammer continuously observes the probability that the user signal appears at each frequency point, and chooses the largest one as jamming channel). For all Jamming patterns, the instantaneous bandwidth of the jamming is set to be 4MHz.

The normalized average throughput of legitimate user under different jamming patterns is given in Fig. 5. It is shown that the anti-jamming ability of user has been improved significantly with the proposed DARLA learning. Especially in the case of comb jamming, the normalized throughput is close to one after convergence, which indicates that the jamming is almost completely avoided.

Environmental states at initial stages under sweep, comb and random jamming patterns are given in Fig. 6(a), (b),

and (c) respectively, and the converging states are given in Fig. 6(d), (e), and (f) respectively. These states that contain time-frequency information can clearly reflect the past actions of user and jamming. Taking sweep jamming as an example, at the beginning of the learning procedure, the user adopts randomized action as it is unfamiliar with environment (the locations of rectangular blocks are randomly distributed), and after convergence, the frequency is properly changed before the jamming arrives (the rectangular blocks are distributed according to the slashes).

With regard to the intelligent jamming, since the probability distribution of user actions is the basis for jammer to release jamming, the best strategy for user is that the probability of each action is almost identical. The simulation results in Fig. 7 show the probabilities of each action being selected during the learning procedure, which is consistent with our analysis.

V. CONCLUSION

In this letter, we investigated the anti-jamming problem in unknown and dynamic environment. Aiming at employing the waterfall spectrum information directly, we constructed a deep anti-jamming Q-network to handle the complex interactive decision-making problem with infinite number of states. Then, a deep anti-jamming reinforcement learning algorithm was proposed. Using the proposed learning algorithm, the user is able to learn the best anti-jamming strategy by constantly trying various actions and sensing the spectrum environment. Simulation results in various scenarios are presented to validate the proposed anti-jamming communication approach. Future work on designing multi-user deep anti-jamming reinforcement learning algorithms is ongoing.

REFERENCES

- [1] L. Zhang, *et al.*, "United against the enemy: Anti-jamming based on cross-layer cooperation in wireless networks" *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5733-5747, Aug. 2016.
- [2] B. Wang, *et al.*, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877-889, Apr. 2011.
- [3] Y. Wu, *et al.*, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 112-118, Aug. 2011.
- [4] M.K. Hanawal, *et al.*, "Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems" *IEEE Trans. Mobile Computing.*, vol. 15, no. 9, pp. 2247-2259, Sep. 2016.
- [5] L. Xiao, *et al.*, "Anti-jamming transmission stackelberg game with observation errors," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 949-952, Jun. 2015.
- [6] L. Jia, *et al.*, "Bayesian stackelberg game for anti-jamming with incomplete information," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 1991-1994, Oct. 2016.
- [7] L. Jia, *et al.*, "A hierarchical learning solution for anti-jamming stackelberg game with discrete power strategies," *IEEE Wireless Commun. Lett.*, doi:10.1109/LWC.2017.2747543.
- [8] H. Zhu, *et al.*, "You can jam but you cannot hide: Defending against jamming attacks for geo-location database driven spectrum sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2723-2737, Oct. 2016.
- [9] W. Chen, *et al.*, "Perceptual spectrum waterfall of pattern shape recognition algorithm," in *Proc. IEEE Conf. ICACT 2016*, pp. 382 - 389.
- [10] C. J. C. H. Watkins, *et al.*, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279-292, 1992.
- [11] V. Mnih, *et al.*, "Human-level control through deep reinforcement learning," *Nature.*, vol. 518, no. 7540, pp. 529-533, Jan. 2015.
- [12] G. Han, *et al.*, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE Conf. ICASSP 2017*, pp. 2087-2091.