Anti-jamming Transmissions with Learning in Heterogenous Cognitive Radio Networks

Tianhua Chen, Jinliang Liu, Liang Xiao*, Lianfen Huang
Dept. Communication Engineering, Xiamen University, 361000, China
*Email: lxiao@xmu.edu.cn

***Abstract*—This paper investigates the interactions between a secondary user (SU) with frequency hopping and a jammer with spectrum sensing in heterogenous cognitive radio networks. The power control interactions are formulated as a multi-stage anti-jamming game, in which the SU and jammer repeatedly choose their power allocation strategies over multiple channels simultaneously without interfering with primary users. We propose a power allocation strategy for the SU to achieve the optimal transmission power and channel with unaware parameters such as the channel gain of the opponent based on reinforcement learning algorithms including Q-learning for and WoLF-Q. Simulation results show that the proposed power allocation strategy can efficiently improve the SU's performance against both sweeping jammers and smart jammers with learning in heterogenous cognitive radio networks.**

***Index Terms*—Heterogenous cognitive radio networks, anti-jamming games, power control, reinforcement learning.**

I. Introduction

In heterogenous cognitive radio networks (CRNs), secondary users (SUs) have to avoid interfering with the transmissions of primary users (PUs). However, jammers inject interfering signals to interrupt the ongoing transmissions of SUs and thus throw denial of service attackers in cognitive radio networks. Anti-jamming transmissions in CRNs have recently attracted extensive research attentions due to the emergence of smart jammers [1]–[5].

Game theory has shown its strength to address jamming attacks. For example, indirect reciprocity principle was applied to suppress attacks including jamming in wireless networks in [6]. In [7], a zero-sum game was formulated for an ad hoc CRN against cognitive jamming. Anti-jamming transmission between a jammer and a defense party was formulated as an asynchronous game [8]. In [9], the jamming game in military background was considered. In [10], fictitious play was investigated in a jamming game in CRNs. [11] investigated the channel access game between two opponent networks with Q-learning.

As the system parameters such as the channel states and the actions of the opponents are not always known in time, reinforcement learning has become an important method for secondary users to choose their transmission strategies with improved anti-jamming performance. In [12], the interaction between an SU and jammer in time variant radio channels was formulated as a stochastic game with minimax-Q learning. The transmission strategy of control channel with multi-agent reinforcement learning was developed to achieve the optimal channel allocation for control data in [13]. In [14], a Q-learning based channel hopping approach was proposed to address jamming. A channel selection strategy with Q-learning was proposed for CRNs in [15]. Jamming strategy can be learnt by leveraging the past history of attacks in CRN in [16]. Adaptive jamming strategy with delayed information regarding the transmitter and receiver was proposed based on reinforcement learning [17].

In this paper, we formulate a repeated anti-jamming game in heterogenous cognitive radio networks, in which an SU chooses its anti-jamming power allocation strategy over multiple channels including both the transmission power and the selected channel. We propose a power allocation strategy for SUs with reinforcement learning to address jamming attacks in unknown radio environments.

The rest of the paper is organized as follows: We present the anti-jamming game in section II and propose the power control strategies with learning in section III. Simulations are presented in section IV, and conclusions are drew in section V.

II. Anti-jamming Game in Heterogenous CRNs

A. System model

In heterogenous cognitive radio networks with multiple channels, an SU determines its transmission power and channel ID according to the RF environment and the other radios in the network. More specifically, an SU with limited spectrum resources aims at selecting proper actions by estimating the behaviors of other nodes. An anti-jamming game between an SU and a jammer in heterogenous CRNs is considered in this section. The SU aims to maximize its throughput at a low transmission cost. The jammer aims to interrupt the communication by creating interference considering energy consumption. Both the SU and jammer are assumed to access only one of M available channels in a time slot and their transmit power can be selected from K levels.

The actions of SU and jammer at time n are given by $\mathbf{x}^n = [P_s, I_s]$ and $\mathbf{y}^n = [P_j, I_j]$ where $P_{s/j} \in \{P_k\}_{1 \le k \le K}$

The work of L. Xiao was partly supported by NSFC (61271242, 61001072, 61301097), NCETFJ, and Fundamental Research Funds for the Central Universities (2012121028, 2013121023). L. Huang was partially supported by 2012 National Natural Science Foundation of China (Grant number 61172097), 2014 National Natural Science Foundation of China (Grant number 61371081).

TABLE I
SUMMARY OF SYMBOLS AND NOTATIONS

M	Number of channels
K	Number of transmit power levels
$P_{s/j} \in \{P_k\}_{1\le k\le K}$	Transmit/jamming power
$\mathbf{x}/\mathbf{y}$	Action of the SU/jammer
$E_{s/j}$	Transmission cost of the SU/jammer
$I_{s/j}$	Channel ID selected by the SU/jammer
$\mathbf{H}_{s/j} = [h_{s/j}^{I_{s/j}}]_{1\le I_{s/j}\le M}$	Channel power gain
α	Indicator of the PU
ρ	Occurrence probability of the PU
$C_{s/j}$	Hopping cost of the SU/jammer
$u_{s/j}$	Utility of the SU/jammer
ε	Channel background noise
$f(\cdot)$	Indicator function
$\mathbf{s} \in \mathcal{S}$	State in state set
$Q_{s/j}$	Quality function of the SU/jammer
$V_{s/j}$	Value function of the SU/jammer
δ	Discounting factor in learning
$\mu_{s/j}$	Learning rate of the SU/jammer
$N(\mathbf{s},\mathbf{x})$	Number of updates for Q(**s**,**x**)
$\pi_{s/j}$	Policy of the SU/jammer
$\bar{\pi}_{s/j}$	Average policy of the SU/jammer
$\delta_{win/lose}$	Learning rate in WoLF-Q
$C(\mathbf{s})$	Number of occurrence of **s**

denote the transmit and jamming power allocated on channel, respectively and $I_{s/j} \in \{1, \cdots, M\}$ denote the channel ID selected by SU and jammer, respectively. The transmission cost per unit power of the SU and jammer is E_s and E_j, respectively. Since each player is constrained to access one channel in a time slot, there is at most one non-zero element in action vector $\mathbf{x}^n$ or $\mathbf{y}^n$.

The channel power gains for SU and jammer are denoted as $\mathbf{H}_s = [h_s^{I_s}]_{1\le I_s\le M}$ and $\mathbf{H}_j = [h_j^{I_j}]_{1\le I_j\le M}$, respectively, where $h_{s/j}^{I_{s/j}}$ is the power gain of channel $I_{s/j}$, which is assumed to be a constant known to both players. Both the SU and jammer are not allowed to disrupt the PU's communication and thus must vacate the channel once a PU reclaims the channel usage right. The presence of the PU is indicated by α, where $\alpha = 0$ indicates that a PU is present on the chosen channel, otherwise, the PU is absent if $\alpha = 1$. The presence of a PU in each time slot is assumed to follow the distribution with probability ρ.

In order to increase the signal-to-interference-and-noise ratio (SINR) at the receiver and reduce transmission cost, the SU selects its transmit power and decides whether hops to another channel according to its strategy at the beginning of each time slot. The jammer also has the choice of hopping to another channel searching for the SU or adjusting its jamming power for better interference. The SINR at the SU's receiver is dragged down because of the existence of the jammer. In order to avoid being jammed, the SU chooses its hopping strategy with cost, denoted by C_s, and adjusts its transmit power. Similarly, the jammer decides whether to hop to another channel at cost, denoted by C_j, or stays in the current channel, and then adjusts its jamming power level to maximize the damage at lower cost. However, once the PU is present on the chosen channel, the SINR at the SU's receiver drops down to zero.

B. Game model

The interactive behaviours between an SU and a jammer can be modeled into a multi-stage game where the SU and jammer act simultaneously. The anti-jamming game can be described as an MDP, where state of the game at time n includes actions of the SU and jammer and activity of the PU in last time slot, i.e., $\mathbf{s}_s^n = [\alpha^{n-1}, \mathbf{y}^{n-1}]$ for the SU and $\mathbf{s}_j^n = [\alpha^{n-1}, \mathbf{x}^{n-1}]$ for jammer. For simplicity of denotation, we omitted the SU and jammer subscript in $\mathbf{s}^n$. Policies $\pi_s : \mathcal{S} \to p(\mathcal{X})$ and $\pi_j : \mathcal{S} \to p(\mathcal{Y})$ are defined for the SU and jammer mapping the state space $\mathcal{S}$ to the action spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively, that can maximize the expected sum of the discounted rewards. The immediate utility of the SU, denoted as u_s, at time n is given as

$$\begin{aligned} u_s(\mathbf{s}_s^n, \mathbf{x}^n) =& \alpha \frac{P_s h_s^{I_s}}{\varepsilon + P_j h_j^{I_s} f(I_s - I_j)} \\ & - C_s[1 - f(I_s^n - I_s^{n-1})] - E_s P_s, \end{aligned} \tag{1}$$

where the first term indicates the SINR of the SU, the second term represents the hopping cost of the SU, and the last term is the transmission energy consumption of the SU. In addition, ε is the constant background noise power, and the indicator function $f(\cdot)$ is another indicator function given by

$$f(x) = \begin{cases} 1, & x = 0 \\ 0, & \text{o.w.} \end{cases}. \tag{2}$$

The immediate utility of the jammer, denoted as u_j, at time n is given by

$$\begin{aligned} u_j(\mathbf{s}_j^n, \mathbf{y}^n) =& -\alpha \frac{P_s h_s^{I_s}}{\varepsilon + P_j h_j^{I_s} f(I_s - I_j)} \\ & - C_j[1 - f(I_j^n - I_j^{n-1})] - E_j P_j, \end{aligned} \tag{3}$$

where the first term is the reward of the jammer, the second term is hopping cost of the jammer, and the third term is transmission energy consumption of the jammer.

The SU decides its action $\mathbf{x}^n$ based on state $\mathbf{s}_s^n$ according to policy $\pi_s(\mathbf{s}_s^n)$ and receives a immediate utility $u_s(\mathbf{s}_s^n, \mathbf{x}^n)$. Similarly, the jammer chooses jamming policy $\mathbf{y}^n$ according to $\pi_j(\mathbf{s}_j^n)$, and receives a immediate utility $u_j(\mathbf{s}_j^n, \mathbf{y}^n)$. For ease of reference, the commonly used notations are summarized in TABLE 1.

III. Power Control With Reinforcement Learning

In practice, the information such as the state transition probability is not directly available for both players, Q-learning is an appropriate method which does not require any explicit information. Further, a modification of the Q-learning, WoLF-Q is introduced to update power allocation strategies.

A. Power allocation strategy with Q-learning

When information of MDP, such as the state transition probability, is not available, the optimal policy can not be obtained through MDP-based iteration. Therefore, an alternative method, Q-learning, is proposed to acquire the optimal policy without knowing the underlying Markov model [15].

At the beginning of each time slot, the SU and jammer decide their actions $\mathbf{x}^n$, $\mathbf{y}^n$ simultaneously based on their actions and behavior of the PU in last time slot. For the SU, the state $\mathbf{s}_s^n = [\alpha^{n-1}, \mathbf{y}^{n-1}]$, then get the next state $\mathbf{s}_s^{n+1} = [\alpha^n, \mathbf{y}^n]$. For simplicity of denotation, we omitted the subscript s in $\mathbf{s}^n$, if no confusion incurs.

Combining Q-learning algorithm with the SU's decision-making for channel hopping and power control, the learning rate of the SU, denoted by μ_s, is firstly updated by

$$\mu_s^n(\mathbf{s}^n, \mathbf{x}^n) = \frac{1}{1 + N(\mathbf{s}^n, \mathbf{x}^n)}, \tag{4}$$

where $N(\mathbf{s}^n, \mathbf{x}^n)$ indicates the number of occurrence for the state-action pair $(\mathbf{s}^n, \mathbf{x}^n)$. It is clear that $\mu(\mathbf{s}, \mathbf{x})$ decreases over time. After observing a sample $(\mathbf{s}^n, \mathbf{x}^n, u(\mathbf{s}^n, \mathbf{x}^n), \mathbf{s}^{n+1})$, the SU's quality function of state-action pair $(\mathbf{s}^n, \mathbf{x}^n)$, denoted as Q_s, is updated by

$$\begin{aligned} Q_s^{n+1}(\mathbf{s}^n,\mathbf{x}^n) &= (1 - \mu_s^n(\mathbf{s}^n, \mathbf{x}^n))Q_s^n(\mathbf{s}^n, \mathbf{x}^n) \\ &+ \mu_s^n(\mathbf{s}^n, \mathbf{x}^n)(u_s(\mathbf{s}^n, \mathbf{x}^n) + \delta V_s^n(\mathbf{s}^{n+1})), \end{aligned} \tag{5}$$

where the discounting factor, denoted by $\delta \in [0, 1]$ indicates the weight of a future payoff over the current payoff, and V_s is the value function updated by

$$V_s^{n+1}(\mathbf{s}) = \max_{\mathbf{x} \in \mathcal{X}} Q_s^{n+1}(\mathbf{s}, \mathbf{x}). \tag{6}$$

Then the transmit policy mapping from the state to an optimal action, denoted as π_s, is updated by

$$\pi_s^{n+1}(\mathbf{s}) = \arg\max_{\mathbf{x} \in \mathcal{X}} Q_s^{n+1}(\mathbf{s}, \mathbf{x}). \tag{7}$$

Details of SU's channel selection and power control with Q-learning are given in Algorithm 1.

The jammer takes action $\mathbf{y}^n$ at state $\mathbf{s}_j^n$, i.e., $\mathbf{s}_j^n = [\alpha^{n-1}, \mathbf{x}^{n-1}]$, to maximize the sum of immediate utility and expected utility conditioned on the current action. For simplicity of denotation, we omitted the subscript j in $\mathbf{s}^n$, if no confusion incurs. The updating process is similar to that of the SU, and the learning rate of the jammer, denoted as μ_j, is given by

$$\mu_j^n(\mathbf{s}^n, \mathbf{y}^n) = \frac{1}{1 + N(\mathbf{s}^n, \mathbf{y}^n)}. \tag{8}$$

The jammer's quality function of state-action pair $(\mathbf{s}^n, \mathbf{y}^n)$, denoted by Q_j, is updated as

$$\begin{aligned} Q_j^{n+1}(\mathbf{s}^n,\mathbf{y}^n) &= (1 - \mu_j^n(\mathbf{s}^n, \mathbf{y}^n))Q_j^n(\mathbf{s}^n, \mathbf{y}^n) \\ &+ \mu_j^n(\mathbf{s}^n, \mathbf{y}^n)(u_j(\mathbf{s}^n, \mathbf{y}^n) + \delta V_j^n(\mathbf{s}^{n+1})), \end{aligned} \tag{9}$$

where the value function V_j is updated by

$$V_j^{n+1}(\mathbf{s}) = \max_{\mathbf{y} \in \mathcal{Y}} Q_j^{n+1}(\mathbf{s}, \mathbf{y}). \tag{10}$$

Algorithm 1 Power control for SU with Q-learning.

Initialize $Q_s(\mathbf{s}, \mathbf{x}) = \mathbf{0}$, $V_s(\mathbf{s}) = \mathbf{0}$, $\pi_s(\mathbf{s}) = [P_1, 0, ..., 0]'$ and $\mu_s(\mathbf{s}, \mathbf{x}) = \mathbf{1}$, $\forall \mathbf{s}, \mathbf{x}$.
Repeat (for each episode)
  Observe the initial system state $\mathbf{s}^1$;
  For n=1, 2, 3,...
    Select an action $\mathbf{x}^n$ at current state $\mathbf{s}^n$;
      At random with a small probability η;
      Otherwise, $\mathbf{x}^n = \pi_s(\mathbf{s}^n)$ by (7).
    Observe the subsequent state $\mathbf{s}^{n+1}$ and immediate reward u_s;
    Update $\mu_s(\mathbf{s}^n, \mathbf{x}^n)$ by (4);
    Update $Q_s(\mathbf{s}^n, \mathbf{x}^n)$ by (5);
    Update $V_s(\mathbf{s}^n)$ by (6);
  End for
End for

Algorithm 2 Jamming power control with Q-learning.

Initialize $Q_j(\mathbf{s}, \mathbf{y}) = \mathbf{0}$, $V_j(\mathbf{s}) = \mathbf{0}$, $\pi_j(\mathbf{s}) = [P_1, 0, ..., 0]'$ and $\mu_j(\mathbf{s}, \mathbf{y}) = \mathbf{1}, \forall \mathbf{s}, \mathbf{y}$.
Repeat (for each episode)
  Observe the initial system state $\mathbf{s}^1$;
  For n=1, 2, 3,...
    Select an action $\mathbf{y}^n$ at current state $\mathbf{s}^n$;
      At random with a small probability η;
      Otherwise, $\mathbf{y}^n = \pi_j(\mathbf{s}^n)$ by (11).
    Observe the subsequent state $\mathbf{s}^{n+1}$ and immediate reward u_j;
    Update $\mu_j(\mathbf{s}^n, \mathbf{y}^n)$ by (8);
    Update $Q_j(\mathbf{s}^n, \mathbf{y}^n)$ by (9);
    Update $V_j(\mathbf{s}^n)$ by (10);
  End for
End for

The optimal jamming policy obtained by maximizing the jammer's quality function, denoted as π_j, is given by

$$\pi_j^{n+1}(\mathbf{s}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} Q_j^{n+1}(\mathbf{s}, \mathbf{y}). \tag{11}$$

Algorithm 2 shows the jamming strategy including the jamming power and channel ID.

B. Power control with WoLF-Q

A modification to the traditional Q-learning algorithm, WoLF-Q algorithm, was introduced to update the learning process with variant learning rates [18]. In WoLF-Q algorithm, two learning rates, denoted by δ_{win} and δ_{lose} ($\delta_{win}, \delta_{lose} \in [0, 1]$, and $\delta_{win} < \delta_{lose}$), are used according to the comparison of the current expected value and the current expected value of the estimated average policy. If the current expected value is lower (i.e., the player is losing), the larger learning rate δ_{lose} is used to learn faster, otherwise δ_{win} is used. Similar to part A, the subscript s and j are omitted in players' individual state $\mathbf{s}^n$. According to WoLF-Q in [18], after observing the state $\mathbf{s}^n = [\alpha^{n-1}, \mathbf{y}^{n-1}]$, the SU updates the number of occurrence of state, denoted as $C(\mathbf{s})$, by

$$C(\mathbf{s}^n) = C(\mathbf{s}^n) + 1. \tag{12}$$

According to WoLF-Q, the SU estimates the average defense policy for $\forall \mathbf{x} \in \mathcal{X}$, denoted as $\bar{\pi}_{s,w}$ which is given by

$$\bar{\pi}_{s,w}^{n+1}(\mathbf{s}^n, \mathbf{x}) = \bar{\pi}_{s,w}^n(\mathbf{s}^n, \mathbf{x}) + \frac{\pi_{s,w}^n(\mathbf{s}^n, \mathbf{x}) - \bar{\pi}_{s,w}^n(\mathbf{s}^n, \mathbf{x})}{C(\mathbf{s}^n)}. \tag{13}$$

Algorithm 3 Power control for SU with WoLF-Q.

Initialize $Q_s(\mathbf{s},\mathbf{x})=\mathbf{0}$, $V_s(\mathbf{s})=\mathbf{0}$, $\pi_{s,w}(\mathbf{s})=\frac{1}{|\mathcal{X}|}$, $\mu_s(\mathbf{s},\mathbf{x})=\mathbf{1}$ and $C(\mathbf{s})=0,\forall \mathbf{s},\mathbf{x}$.
Repeat (for each episode)
  Observe the initial system state $\mathbf{s}^1$;
  For n=1, 2, 3,...
    Select an action $\mathbf{x}^n$ at current state $\mathbf{s}^n$;
      Uniformly at random with a small probability η;
      Otherwise, With probability $\pi_{s,w}(\mathbf{s}^n)$.
    Observe the subsequent state $\mathbf{s}^{n+1}$ and immediate reward u_s;
    Update $\mu_s(\mathbf{s}^n,\mathbf{x}^n)$ by (4);
    Update $Q_s(\mathbf{s}^n,\mathbf{x}^n)$ by (5);
    Update $V_s(\mathbf{s}^n)$ by (6);
    Update $C(\mathbf{s}^n)$ by (12);
    Update estimated average policy $\bar{\pi}_{s,w}$ by (13);
    Update actual policy $\pi_{s,w}$ by (14);
  End for
End for

Algorithm 4 Jamming power control with WoLF-Q.

Initialize $Q_j(\mathbf{s},\mathbf{y})=\mathbf{0}$, $V_j(\mathbf{s})=\mathbf{0}$, $\pi_{j,w}(\mathbf{s})=\frac{1}{|\mathcal{Y}|}$, $\mu_j(\mathbf{s},\mathbf{y})=\mathbf{1}$ and $C(\mathbf{s})=0,\forall \mathbf{s},\mathbf{y}$.
Repeat (for each episode)
  Observe the initial system state $\mathbf{s}^1$;
  For n=1, 2, 3,...
    Select an action $\mathbf{y}^n$ at current state $\mathbf{s}^n$;
      Uniformly at random with a small probability η;
      Otherwise, With probability $\pi_{j,w}(\mathbf{s}^n)$.
    Observe the subsequent state $\mathbf{s}^{n+1}$ and immediate reward u_j;
    Update $\mu_j(\mathbf{s}^n,\mathbf{y}^n)$ by (8);
    Update $Q_j(\mathbf{s}^n,\mathbf{y}^n)$ by (9);
    Update $V_j(\mathbf{s}^n)$ by (10);
    Update $C(\mathbf{s}^n)$ by (12);
    Update estimated average policy $\bar{\pi}_{j,w}$ by (18);
    Update actual policy $\pi_{j,w}$ by (19);
  End for
End for

The actual defense policy of the SU for $\forall \mathbf{x}\in\mathcal{X}$, denoted as $\pi_{s,w}$, is updated by

$$\pi_{s,w}^{n+1}(\mathbf{s}^n,\mathbf{x})=\pi_{s,w}^{n}(\mathbf{s}^n,\mathbf{x})+\triangle_{\mathbf{s},\mathbf{x}}, \tag{14}$$

where the weight is given by

$$\triangle_{\mathbf{s},\mathbf{x}}=\begin{cases}-\delta_{s,\mathbf{x}}, & \mathbf{x}\neq \arg\max_{\mathbf{x}'} Q_{s,w}(\mathbf{s},\mathbf{x}')\\ \sum_{\mathbf{x}\neq\mathbf{x}'}\delta_{s,\mathbf{x}'}, & \text{o.w.}\end{cases}, \tag{15}$$

where $\delta_{s,\mathbf{x}}=\min\{\pi_{s,w}(\mathbf{s},\mathbf{x}),\frac{\delta_s}{|\mathcal{X}|-1}\}$ and $|\mathcal{X}|$ is the size of the action set $\mathcal{X}$. The learning rate of the SU, denoted as δ_s, is given by

$$\delta_s=\begin{cases}\delta_{win}, & \Pi\\ \delta_{lose}, & \text{o.w.}\end{cases}, \tag{16}$$

where Π is true if the following holds:

$$\sum_{\mathbf{x}'}\pi_{s,w}^{n}(\mathbf{s}^n,\mathbf{x}')Q_{s,w}^{n+1}(\mathbf{s}^n,\mathbf{x}') > \sum_{\mathbf{x}'}\bar{\pi}_{s,w}^{n}(\mathbf{s}^n,\mathbf{x}')Q_{s,w}^{n+1}(\mathbf{s}^n,\mathbf{x}'). \tag{17}$$

Details of the SU's channel hopping and power control with WoLF-Q algorithm are presented in Algorithm 3.

Combining WoLF-Q algorithm with jamming policy updating process, the state occurrence count $C(s)$ is the same as Eq. (12). Similarly, the estimated average jamming policy for $\forall \mathbf{y}\in\mathcal{Y}$, denoted as $\bar{\pi}_{j,w}$, is given by

$$\bar{\pi}_{j,w}^{n+1}(\mathbf{s}^n,\mathbf{y})=\bar{\pi}_{j,w}^{n}(\mathbf{s}^n,\mathbf{y})+\frac{\pi_{j,w}^{n}(\mathbf{s}^n,\mathbf{y})-\bar{\pi}_{j,w}^{n}(\mathbf{s}^n,\mathbf{y})}{C(\mathbf{s}^n)}. \tag{18}$$

The actual jamming policy for $\forall \mathbf{y}\in\mathcal{Y}$, denoted as $\pi_{j,w}$, is given by

$$\pi_{j,w}^{n+1}(\mathbf{s}^n,\mathbf{y})=\pi_{j,w}^{n}(\mathbf{s}^n,\mathbf{y})+\triangle_{\mathbf{s},\mathbf{y}}, \tag{19}$$

where the weight is given by

$$\triangle_{\mathbf{s},\mathbf{y}}=\begin{cases}-\delta_{s,\mathbf{y}}, & \mathbf{y}\neq \arg\max_{\mathbf{y}'} Q_{j,w}(\mathbf{s},\mathbf{y}')\\ \sum_{\mathbf{y}\neq\mathbf{y}'}\delta_{s,\mathbf{y}'}, & \text{o.w.}\end{cases}, \tag{20}$$

where $\delta_{s,\mathbf{y}}=\min\{\pi_{j,w}(\mathbf{s},\mathbf{y}),\frac{\delta_j}{|\mathcal{Y}|-1}\}$ and $|\mathcal{Y}|$ is the size of the action set $\mathcal{Y}$. The learning rate of the jammer, denoted by δ_j, is given by

$$\delta_j=\begin{cases}\delta_{win}, & \Gamma\\ \delta_{lose}, & \text{o.w.}\end{cases}, \tag{21}$$

where Γ is true if the following holds:

$$\sum_{\mathbf{y}'}\pi_{j,w}^{n}(\mathbf{s}^n,\mathbf{y}')Q_{j,w}^{n+1}(\mathbf{s}^n,\mathbf{y}') > \sum_{\mathbf{y}'}\bar{\pi}_{j,w}^{n}(\mathbf{s}^n,\mathbf{y}')Q_{j,w}^{n+1}(\mathbf{s}^n,\mathbf{y}'). \tag{22}$$

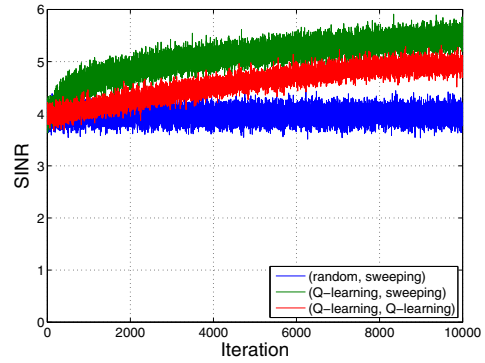
Details of the jammer's channel selection and power control with WoLF-Q algorithm are presented in Algorithm 4.

Note that the policy $\pi(\mathbf{s})$ during updating process may not be the true optimal one. Always choosing action $\mathbf{x}\in\mathcal{X}$ and $\mathbf{y}\in\mathcal{Y}$ according to $\pi_s(\mathbf{s})$ and $\pi_j(\mathbf{s})$ may improve the probability of the unexpected action and prevent the optimal one to be discovered. In order to investigate the rarely accessed state-action pairs, it assumed that the SU and jammer can visit other actions with a small probability η at each updating round.

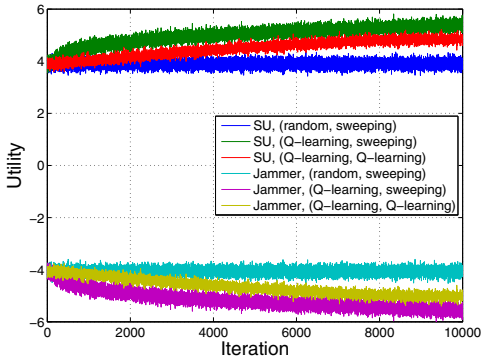
IV. Simulation Results

Simulations have been performed to evaluate performance of multi-channel power control anti-jamming game. In the simulations, we set $M=16$, $\varepsilon=1$, $C_s=C_j=0.01$, $E_s=E_j=0.01$, $\rho=0$ and $P_i\in\{0,5,10\}$, if not specified otherwise. In order to give facilities to analysis of interactions between SU and jammer , it is assumed that the PU is absent in the CRN, i.e., $\alpha=1$.

Power allocation strategies based on Q-learning in anti-jamming game in heterogenous CRNs were evaluated in Fig. 1. The SU using Q-learning determines its policy consisting of selecting a channel and transmitting power. The jammer selects a channel to jam with an appropriate power through Q-learning or sweeps over all channels with the maximum power. As shown in Fig. 1, in the case in which SU uses Q-learning against a sweeping jammer, the SINR increases with time and eventually converges to the optimal value (up to about 5.7 at 10000 time slots), which leads to the gradual rise in the utility of the SU and the loss of the benefit of the jammer. This can be explained by the SU's capability of faster

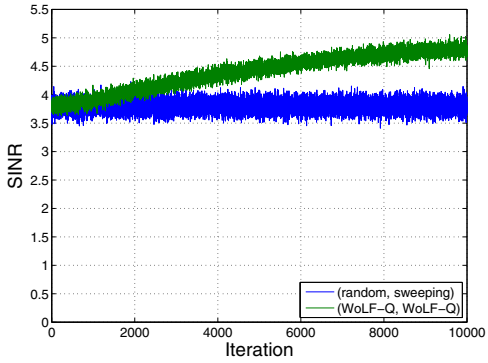


(a) SINR

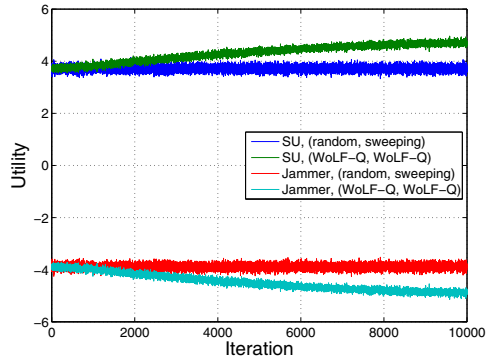


(a) Utility

Fig. 1. Performance of the SU against a jammer with $\delta = 0.7$.



(a) SINR



(a) Utility

Fig. 2. Performance of the SU against a jammer with $\delta_{win} = 0.05$ and $\delta_{lose} = 0.1$.

adaption to the jammer. Although the jammer uses Q-learning to select jamming channel and power, both the utility of the SU and the SINR increase and converge to maximized values because the SU with learning is more likely to avoiding to be jammed in multi-channel. In addition, when the SU with random strategy defends against a sweeping jammer, the SINR is around 4 on average.

WoLF-Q was employed by the SU to update the defence policy and the jammer to determine jamming strategy. The effectiveness of WoLF-Q is evaluated in terms of the SINR and utility in Fig. 2. If both SU and jammer use WoLF-Q, the SINR for the SU begins to increase not long after the start of learning and then rises slightly and eventually converges to about 5.

V. Conclusions

In this paper, we have formulated a transmission game between an SU and a jammer in heterogenous cognitive radio networks with multiple channels. Power control strategies with reinforcement learning including Q-learning and WoLF-Q were proposed for SUs to achieve their optimal transmission strategies with unknown parameters such as the channel gains of the opponent. Simulation results indicate that the proposed power allocation strategies can significantly improve the performance against both sweeping and smart jammers with learning. For example, the SINR of an SU increases from 4 to around 5.7 by applying Q-learning in the power allocation over 16 channels against a sweeping jammer. The utility of the SU increases from 4 to 5 even in presence of a smart jammer with Q-learning.

References

[1] L. Xiao, H. Dai, and P. Ning, "Mac design of uncoordinated fh-based collaborative broadcast," *IEEE Wireless Communications Letters*, vol. 1, no. 3, pp. 261–264, 2012.

[2] Y. Li, L. Xiao, J. Liu, and Y. Tang, "Power control stackelberg game in cooperative anti-jamming communications," in *Proc. Int'l Conf. Game Theory for Networks*, pp. 93–98, 2014.

[3] J. Liu, L. Xiao, Y. Li, and L. Huang, "User-centric analysis on jamming game with action detection error," in *Proc. Int'l Conf. Game Theory for Networks*, pp. 120–125, 2014.

[4] X. He, H. Dai, and P. Ning, "A byzantine attack defender in cognitive radio networks: the conditional frequency check," *IEEE Tran. Wireless Commun.*, vol. 12, no. 5, pp. 2512–2523, 2013.

[5] W. Shen, P. Ning, X. He, H. Dai, and Y. Liu, "Mcr decoding: A mimo approach for defending against wireless jamming attacks," in *IEEE Conf. Communications and Network Security (CNS)*, pp. 133–138, 2014.

[6] L. Xiao, Y. Chen, W. S. Lin, and K. J. R. Liu, "Indirect reciprocity security game for large-scale wireless networks," *IEEE Trans. Information Forensics and Security*, vol. 7, no. 4, pp. 1368–1380, 2012.

[7] W. G. Conley and A. J. Miller, "Cognitive jamming game for dynamically countering ad hoc cognitive radio networks," in *Proc. IEEE Military Commun. Conference (MILCOM)*, pp. 1176–1182, 2013.

[8] B. DeBruhl and P. Tague, "Living with boisterous neighbors: Studying the interaction of adaptive jamming and anti-jamming," in *Proc. IEEE Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, 2012.

[9] S. Dastangoo, C. E. Fossa, and Y. L. Gwon, "Competing cognitive tactical networks," *Lincoln Laboratory Journal*, vol. 20, no. 2, 2014.
[10] K. Dabcevic, A. Betancourt, L. Marcenaro, and C. S. Regazzoni, "A fictitious play-based game-theoretical approach to alleviating jamming attacks for cognitive radios," *IEEE Int'l Conf. Acoustich, Speech and Signal Processing (ICASSP)*, pp. 8208–8212, 2014.
[11] Y. Gwon, S. Dastangoo, C. Fossa, and H. Kung, "Competing mobile network game: Embracing antijamming and jamming strategies with reinforcement learning," in *Proc. IEEE Conf. Commun. and Network Security (CNS)*, pp. 28–36, 2013.
[12] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, 2011.
[13] B. F. Lo and I. F. Akyildiz, "Multiagent jamming-resilient control channel game for cognitive radio ad hoc networks," in *Proc. IEEE Int'l Conf. Commun. (ICC)*, pp. 1821–1826, 2012.
[14] C. Chen, M. Song, C. Xin, and J. Backens, "A game-theoretical anti-jamming scheme for cognitive radio networks," *IEEE Network*, vol. 27, no. 3, 2013.
[15] Y. Wu, B. Wang, K. J. R. Liu, and T. C. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 4 – 15, 2012.
[16] S. Bhunia, S. Sengupta, and F. Vazquez-Abad, "Cr-honeynet: A learning & decoy based sustenance mechanism against jamming attack in crn," in *IEEE Military Commun. Conference (MILCOM)*, pp. 1173–1180, 2014.
[17] S. Amuru and R. M. Buehrer, "Optimal jamming using delayed learning," in *IEEE Military Commun. Conference (MILCOM)*, pp. 1528–1533, 2014.
[18] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, no. 2, pp. 215 – 250, 2002.