On the performance of overlaid wireless energy harvesting cognitive industrial sensor networks under jamming attacks

Hang Yang[1,2], and *Xunbo* Li[1,*]

[1]University of Electronic Science and Technology of China, Chengdu, China
[2]University of Alberta, Edmonton, Canada

Abstract. Two or more wireless sensor networks coexist in the same space while low energy consuming devices mobiles in a secondary network harvest ambient RF energy from transmissions by nearby active transmitters in the primary network. The channels are allocated to the primary network, while the overlaid secondary network can access the idle channel allocated to perform data transmission opportunistically and operate properly. In this paper, with the jammer implanted, we propose a novel solution in which we execute a deception strategy to exhaust the energy of the jammers. As a result, the energy constraint jammers will be challenging to achieve jamming attacks when the secondary transmitters (STs) transmit information. We formulate the problem first to tackle the issue; that is, we regard throughput optimization issues for ST under jamming attacks as a Markov decision process (MDP). Then, since the focus is mainly on the throughput of the secondary network, a learning algorithm is adopted to maximize it. Through the learning process, the STs can adapt to the dynamics of the primary network while executing proper actions to benefit the overall throughput online. Simulations validate the efficiency and the convergence of the algorithm we proposed.

Keywords: Cognitive industrial sensor networks, Energy harvest, Jamming attack, Deception strategy.

1 Introduction

Wireless cognitive sensor networks with wireless energy harvesting have drawn much attention recently. Powering low-power mobile devices by harvesting energy from ambient radio frequencies such as solar, wind, and kinetic activities makes wireless networks not only environmentally friendly but also self-sustaining. With the progress on designing efficient circuits and devices for radio-frequency (RF) energy harvesting suitable for low-power applications, the proposal of a novel network model becomes available. Two or more wireless sensor networks coexist in the same space while low-power mobiles in a secondary network, called secondary transmitters (STs), harvest ambient RF energy from

* Corresponding author: xbli@uestc.edu.cn

transmissions by nearby active transmitters in the primary network, called primary transmitters (PTs). The channels are allocated to the primary network while the STs are unlicensed users. The STs can be self-sustaining by harvesting energy from ambient RF signals of primary networks. By opportunistically accessing the idle channel allocated, the overlaid secondary network can perform data transmission and operate properly. While wireless power transfer techniques ease the energy limitation caused by constrained hardware of mobile devices on STs, mobile nodes can be self-maintained. Such networks have become more common since the wild spread of wireless charging techniques while focusing on the performance of the secondary users. And malicious transmitters which execute jamming attacks can easily harm the STs, while the energy for jamming attacks is also from harvested wireless RF energy.

In our network model, the PTs own the licensed spectrum; thus, the STs are not immune from intruders. With the wireless power supply, the intruders can also harvest wireless RF energy to execute attacks on the ST's information transmission process to harm the throughput of the overlaid secondary wireless energy harvesting cognitive sensor networks. In this paper, we assume the intruders can use smart sensing processes to distinguish between the signals from the PTs' transmission and that from the STs' transmission.

The jamming attacks issues in classical wireless systems are generally seen as there are such an amount of channels, and jammers can only manage to invalidate a few of them at a time. Therefore, the solution lies in choosing the unlikely attacked channels or switching to another channel when the one is under attack. However, the common situation is that the STs are designed to transmit on a fixed channel opportunistically. Thus, we need to come out with a totally different strategy to tackle this issue. Actually, since the jammers rely on the wireless power supply and their energy is limited, they cannot perform attacks on a single channel continuously. Therefore, we propose a novel solution in this work which we execute a deception strategy. That is, the STs can perform blank transmissions to crack the ability of the intruders while wasting the energy of the jammers. As a result, the jammers may find it challenging to execute jamming attacks when the STs transmit real information.

We formulate the problem first to tackle the issue; that is, we regard throughput optimization issues for ST under jamming attacks as an MDP. Then, since our objective is mainly on the throughput of the secondary network, a learning algorithm is adopted. The novel learning algorithm here is derived from simulation-based and policy gradient methods. Through the learning process, the STs can adapt to the dynamics of the primary network while executing proper actions to benefit the overall throughput online.

Later, we do simulations to validate the efficiency and the convergence of the algorithm we proposed. Through the simulations, we demonstrate that under such multiple jammers' conditions, our learning algorithms are capable of effectively balancing deception actions to data transmission ratio so as to achieve optimized solutions. The convergence proofs are given with simulations while we compare other algorithms. It has to be emphasized here that in our problem formulation, the smart jammers are only aimed at degrading the throughput of the secondary network and are not into the primary network. That is, the secondary users are their targets.

The rest of the paper is organized as follows. Related works are provided in section 2, and our system models are provided in section 3. In section 4, we focus on a single ST and provide the learning algorithm to optimize the performance of the ST. The simulations are present in section 5, while section 6 concludes the paper.

2 Related work

Many optimization problems for the secondary network with energy harvesting lie in the cognitive radio network and focus on throughput[1], energy consumption[2, 3], and channel

access[4, 5] in the literature. This paper is in the realm of cognitive industrial wireless sensor networks, and the network operation is more energy constraint and computation complexity limited. Here we are concerning the average packet update delay of the secondary network instead of the average packet delay of the secondary network. Our assumption is more on the industrial application facts than the communication side. Targeting jamming with one target channel in industrial is not found so now in the existing literature. These are the main contribution of our work.

3 System model

We formulate the primary cognitive wireless sensor network with energy harvesting circuits and coexisting with STs and jammers in the same spatial domain. While the STs harvest energy and use it to transmit data, the jammers conduct jamming attacks to destroy the communication. We assume the PTs utilize the channel on a time slot basis and p_{ch}^{idle} denotes the idle probability of the channel.

The STs are equipped with a battery to store the harvested energy together with a buffer to store unsent incoming data. The STs sense the status of the channel at each time slot as defined; it can be either busy or idle. If it is busy, the ST will stand by. Otherwise, if it is not, the ST will execute transmission while it has harvested or stored enough energy. Concerning spectrum sensing of ST, it may come across some errors, and one is that it misses detecting the channel idle state as busy, and the other is that it misses detecting the channel state as idle. Above two types of errors are straightforward, and we describe them as type I error and type II error and denote the probability of them by p_{ST}^{I} and p_{ST}^{II} respectively.

With the assumption that we utilize the channel on a time slot basis, we also assume that the ST can complete energy harvest and data transmission simultaneously on a time slot basis. We denote the battery capacity of the ST as E_{ST} units while the energy harvest circuits can bring e_{ST}^{h} units of energy in each time slot with probability λ_{ST}^{h}. For the sake of simplicity, here we ignore the energy consumption of other operations on ST. The working scheme of the powered ST is on the condition that it could just stand by, transmit data while its data storing buffer is not empty or send the trap packet to exhaust the jammers' energy(deception). On the contrary, when the ST has no energy, the ST will stand by. Also, we assume the jammer is capable of distinguishing the transmission between the PTs and STs. When the jammer detects the ST's signals, the jammer will perform the attack by jamming into the whole time slot on the channel. In fact, the assumption is realistic since the PT's signals can be recognized by techniques as in [6].

The paper aims to maximize the throughput of the STs under the attacks of the jammers; with our deception strategy, there's a trade-off between transmitting actual data or transmitting deception packets. If the ST transmits fake packets to deceive the jammer into attacking, the action may consume lots of energy of the jammer, which benefits the STs. That is, when the ST has actual data packets to transmit, the jammer may not have enough energy to execute an attack. In practice, the power consumption of transmitting actual and fake data is different; otherwise, the deception method would be in vain since the STs can choose the retransmit organism. Here, we denote the energy consumption of the actual packet transmission as e_{ST}^{tr} while the energy consumption of the fake packet transmission as e_{ST}^{de}, and $e_{ST}^{de} \ll e_{ST}^{tr}$. This could happen when the ST only transmits on a fraction of

time on the slot to deceive the jammer into executing the attack while on an energy-saving basis.

Similar to STs, we assume each of the jammers can harvest e_{ja}^{h} units of energy with probability λ_{ja}^{h} each time and is equipped with a battery with a capacity of E_{ja}. The working scheme of the powered jammers is on the condition that they will perform jamming attacks once they detect signals sent from the STs. On the contrary, they will stand by when the jammers have no energy. We assume the jammer needs e_{ja}^{at} units of energy to execute the attack. It is straightforward that $e_{ja}^{at} \gg e_{ST}^{tr}$, since the jammer needs to jam the channel.

In the following sections, we will mimic the actions of the STs to maximize the throughput while dealing with smart jamming attacks. We will only illustrate the case of one ST since the scenarios are likewise. A learning algorithm will be proposed to manage the transmission and deception strategy, together with a highly adaptive MAC access mechanism for an industrial wireless sensor network.

4 The optimal strategy for a single secondary transmitter

We study the case of an ST under the attack from jammers in this section. First, we formulate the optimization problem as an MDP. Then the aforementioned learning algorithm, which is based on simulation-based and policy gradient methods, is followed by later contents to help optimize the performance metrics of the ST.

4.1 Optimization problem formulation

The problem is formulated in four parts, that is, state space, action space, transition probability and reward function, respectively.

4.1.1 State space

$$\varsigma \triangleq \left\{ (d,e);\ \ d \in \{0,1\}, e \in \{0,1,2,\cdots,E_{ST}\} \right\} \tag{1}$$

where d is data packets exist in the data buffer, and without loss of generality, we only define two values, 0 and 1, for not having or having packets to transmit. e is the battery status of the ST, which stands out in discrete levels.

4.1.2 Action space

The ST will need to choose an action to make after each channel sensing, and the possible actions are stand by, transmit data and execute deception. So for the action space:

$$\mathfrak{A} \triangleq \left\{ a; a \in \{1,2,3\} \right\} \tag{2}$$

where

$$a=\begin{cases}1, & \text{stand by}\\ 2, & \text{transmit data}\\ 3, & \text{deception}\end{cases}$$

And the allowed transition for the state $i\in\varsigma$ are as follows:

$$\mathfrak{A}_i=\begin{cases}\{1\}, & e_i<e_{ST}^{de}\\ \{1,3\}, & e_{ST}^{de}\le e_i<e_{ST}^{tr}\\ & \text{or } e_{ST}^{tr}\le e_i \,\&\, d_i=0\\ \{1,2,3\}, & e_{ST}^{tr}\le e_i \,\&\, d_i>0\end{cases} \tag{3}$$

where the first denotes the battery level is too low to act, and the ST can only choose to stand by. The second denotes that the ST has enough energy to execute deception but not enough to transmit actual data; thus, the ST can choose to perform deception or stand by. The third denotes that the ST has enough energy to perform any one of the actions.

4.1.3 Transition probability

Typically, we need to come up with a transition probability matrix for the MDP. But under our assumption that the objective of the jammer is to attack the ST's transmissions, the jammer will not disclose its information to the ST. As a matter of fact, network information like the channel idle probability p_{ch}^{idle}, the type I error probability p_{ST}^{I}, and the type II error probability p_{ST}^{II} are all inaccessible. Thus, we cannot directly get the transition probability matrix.

The main idea of the learning method[7] we propose is to simulate the actions of ST, channel, and jammers through parameters generating; thus, the ST can update its parameters together with optimized decisions as directed. Next, we do formula computing.

For a control policy κ, the transition probability(probability of going from one state to another in the stochastic process in one step) function is:

$$\begin{aligned}&\mathbf{p}\left(i(t+1)\middle|i(t),\kappa\right)=\mathbf{p}\left[\left(d(t+1),e(t+1)\middle|d(t),e(t)\right),\kappa\right]\\&=\begin{cases}\mathbf{p}_{\text{simu}}\,p\left(\left(d(t),e(t)\right)\right)p\left(\kappa\left(a(t)\right)\right) & E^*=e(t+1)\,\&\,D^*=d(t+1)\\ 0 & \text{otherwise}\end{cases}\end{aligned} \tag{4}$$

And $E^*=\min\left(\left[e(t)-e'(t)\right]^+ +e''(t),E_{ST}\right)$,

$D^*=\min\left(\left(\left[d(t)-d'(t)\right]^+ +d''(t)\right),1\right)$. Where $\mathbf{p}_{\text{simu}}$ is the probability function for the simulator which generates the parameters. d and e stand for the packets number and energy level, respectively; thus, $p\left(\left(d(t),e(t)\right)\right)$ represents the probability of state i on

time t. $p\left(\kappa\left(a\left(t\right)\right)\right)$ represents the probability when taking action a according to control policy κ at time t of ST. $d^{'}\left(t\right)$ stands for the sent packets while $d^{''}\left(t\right)$ stands for the received. $e^{'}\left(t\right)$ stands for the energy consumed while $e^{''}\left(t\right)$ stands for the harvested.

4.1.1 Reward function

Here it refers to the throughput of the ST:

$$\wp\left(i,a\right)=\begin{cases}1, \text{packets received}\\ 0, \qquad \text{otherwise}\end{cases} \tag{5}$$

For now, we have formulated the optimization problem, and next we will try a new policy-based method to solve this problem as there is no direct traditional way to work it out.

4.2 Parameterization modelling

We utilize a randomized parameterized policy as in literature[8, 9] to make the decision for ST, and the parameter vector is Φ ($\Phi=\left\{\varphi_{i,a}\in\mathbb{R}\right\}$, at state i with action a). On the policy, the ST will execute an action a with probability $\mu_{\Phi}\left(i,a\right)$, when currently at state i:

$$\mu_{\Phi}\left(i,a\right)=\frac{\exp\left(\varphi_{i,a}\right)}{\sum_{a_j\in\mathfrak{A}_i}\exp\left(\varphi_{i,a_j}\right)} \tag{6}$$

with components $\mu_{\theta}\left(i,a\right)$ such that

$$\sum_{a\in\mathfrak{A}_i}\mu_{\Phi}\left(i,a\right)=1 \tag{7}$$

And the transition probability and the immediate reward after being parameterized for $i,i^{'}\in\varsigma$ is as:

$$\mathbf{p}\left(i^{'}\left|i,\kappa\left(\Phi\right)\right.\right)=\sum_{a\in\mathfrak{A}_i}\mu_{\Phi}\left(i,a\right)\mathbf{p}\left(i^{'}\left|i,a\right.\right) \tag{8}$$

$$\wp\left(i,\Phi\right)=\sum_{a\in\mathfrak{A}_i}\mu_{\Phi}\left(i,a\right)\wp\left(i,a\right) \tag{9}$$

Also, we have the average throughput function as:

$$\chi\left(\Phi\right)=\lim_{m\to\infty}\frac{1}{m}\mathrm{E}_{\Phi}\left[\sum_{k=0}^{m}\wp\left(i_k,\Phi_k\right)\right] \tag{10}$$

where $\mathrm{E}_{\Phi}\left[\cdot\right]$ represents the expectation, i_k represents the state of ST at the time k ..

As with the same assumption in [8], let $\overline{\varsigma}$ be the closure of ς , we have:

Assumption I The Markov chain corresponding to every $\varsigma \in \overline{\varsigma}$ is aperiodic. And, there's a state i^* recurrent for every such Markov chain.

Assumption II For state $i, i^{'} \in \varsigma$, $\varsigma \in \overline{\varsigma}$, the transmission probability $\mathbf{p}\left(i^{'} \middle| i, \kappa\left(\Phi\right)\right)$ and reward function $\wp\left(i, \Phi\right)$ are bounded, twice differentiable, and have both bounded first and second derivatives.

With assumption I, the following balance equations can be derived:

$$\sum\nolimits_{i\in\varsigma} \pi_{\Phi}\left(i\right)\mathbf{p}\left(i^{'} \middle| i, \kappa\left(\Gamma\right)\right) = \pi_{\Phi}\left(i^{'}\right), i^{'} \in \varsigma \tag{11}$$

$$\sum\nolimits_{i\in\varsigma} \pi_{\Phi}\left(i\right) = 1 \tag{12}$$

where exists a unique solution $\pi_{\Phi} = \left[\cdots \pi_{\Phi}\left(i\right)\cdots\right]^{\mathrm{T}}$, with $\pi_{\theta}\left(i\right)$ being the steady-state probability of state i under a particular vector Φ . As a performance metric we derive the average throughput here

$$\chi\left(\Phi\right) = \sum\nolimits_{i\in\varsigma} \pi_{\Phi}\left(i\right)\wp\left(i, \Phi\right) \tag{13}$$

Under Assumption I, it's evident that the average reward is wholly defined and doesn't rely on the initial state. We can define the differential reward throughput at state $i^{'}$ as:

$$d\left(i^{'}, \Phi\right) = \mathrm{E}_{\Phi}\left[\sum\nolimits_{k=0}^{T-1}\left(\wp\left(i_k, \Phi\right) - \chi\left(\Phi\right)\right)\middle| i_0 = i^{'}\right] \tag{14}$$

where T is the cyclicality, that is, the first future time state i^* is revisited and $T = \min\left\{k > 0 \middle| i_k = i^*\right\}$.

4.3 The idealized gradient algorithm

It is natural to consider gradient-type methods since our goal is to maximize the average reward. We use the following formula for the gradient $\nabla\chi\left(\Phi_k\right)$ of the average throughput with respect to the parameter vector Φ [10] with form:

$$\nabla\chi\left(\Phi\right) = \sum\nolimits_{i\in\varsigma} \pi_{\Phi}\left(i\right)\left(\nabla\wp\left(i, \Phi\right) + \sum\nolimits_{i^{'}\in\varsigma} \nabla\mathbf{p}\left(i^{'} \middle| i, \kappa\left(\Phi\right)\right)d\left(i^{'}, \Phi\right)\right) \tag{15}$$

As is seen, we do consider an algorithm with the form

$$\Phi_{k+1} = \Phi_k + \omega_k\nabla\chi\left(\Phi_k\right) \tag{16}$$

where $\nabla\chi\left(\Phi_k\right)$ denotes the gradient of the average throughput, and ω_k denotes the step size.

Similarly, if we could use a simulator $F_m\left(\Phi\right)$ of $\nabla\chi\left(\Phi_k\right)$ to execute the unbiased estimate of Φ and adopt the approximate gradient iteration to tune as:

$$\Phi_{k+1} = \Phi_k + \omega_k F_m\left(\Phi\right) \tag{17}$$

We then make the following assumptions[8] :

Assumption III The step size is nonnegative while satisfy

$$\sum_{k=1}^{\infty}\omega_k = \infty, \sum_{k=1}^{\infty}\omega_k^2 < \infty \tag{18}$$

Assumption II presents that both the transition probability and immediate reward are dependent on Φ. Assumption III indicates the convergence of the policy gradient method. Under Assumptions I & III, it is proved in[8] $\chi\left(\Phi\right)$ converges and $\lim_{k\to\infty}\nabla\chi\left(\Phi_k\right) = 0$ with probability 1.

Thus under Assumption I, the differential throughput $d\left(i,\Phi\right)$ matches the unique solution of the Bellman equation, which is defined as:

$$d\left(i,\Phi\right) = \wp\left(i,\Phi\right) - \chi\left(\Phi\right) + \sum_{i'\in\varsigma}\mathbf{p}\left(i'\middle|i,\kappa\left(\Phi\right)\right)d\left(i',\Phi\right) \tag{19}$$

4.4 Learning algorithm for estimation of $\nabla\chi\left(\Phi\right)$

Since the network may have an ample state space, as just mentioned earlier, We consider a new strategy which can perform estimation of the value of $\nabla\chi\left(\Phi\right)$ and update Φ; since it is impossible and inefficient to calculate it.

From (7) we have

$$\sum_{a\in\mathfrak{A}_i}\nabla\mu_\Phi\left(i,a\right) = 0 \tag{20}$$

From (9) we have

$$\begin{aligned}\nabla\wp\left(i,\Phi\right) &= \sum_{a\in\mathfrak{A}_i}\nabla\mu_\Phi\left(i,a\right)\wp\left(i,a\right)\\ &= \sum_{a\in\mathfrak{A}_i}\nabla\mu_\Phi\left(i,a\right)\left(\wp\left(i,a\right) - \chi\left(\Phi\right)\right)\end{aligned} \tag{21}$$

From (8) we obtain

$$\sum_{i'\in\varsigma}\mathbf{p}\left(i'\middle|i,\kappa\left(\Phi\right)\right)d\left(i',\Phi\right) = \sum_{i'\in\varsigma}\sum_{a\in\mathfrak{A}_i}\mu_\Phi\left(i,a\right)\mathbf{p}\left(i'\middle|i,a\right)d\left(i',\Phi\right) \tag{22}$$

Combining equations (21) , (22)and(15), and then we can rewrite (15) as:

$$\begin{aligned}\nabla\chi(\Phi) &= \sum_{i\in\varsigma}\sum_{a\in\mathfrak{A}_i}\pi_\Phi(i)\nabla\mu_\Phi(i,a)\left(\left(\wp(i,a)-\chi(\Phi)\right)+\sum_{i'\in\varsigma}\mathbf{p}\left(i'\middle|i,a\right)d\left(i',\Phi\right)\right) \\ &\triangleq \sum_{i\in\varsigma}\sum_{a\in\mathfrak{A}_i}\pi_\Phi(i)\nabla\mu_\Phi(i,a)\nu_\Phi(i,a)\end{aligned} \tag{23}$$

where

$$\begin{aligned}\nu_\Phi(i,a) &= \left(\wp(i,a)-\chi(\Phi)\right)+\sum_{i'\in\varsigma}\mathbf{p}\left(i'\middle|i,a\right)d\left(i',\Phi\right) \\ &= \mathrm{E}_\Phi\left[\sum_{k=0}^{T-1}\left(\wp\left(i_k,a_k\right)-\chi(\Phi)\right)\middle|i_0=i,a_0=a\right]\end{aligned} \tag{24}$$

where T is the cyclicality, that is, the first time state i^* is revisited and $T=\min\left\{k>0\middle|i_k=i^*\right\}$. However, the exact value of the diferential reward is replaced by the approximation $\nu_\Phi(i,a)$ while executing the action a is on μ_Φ at state i.

Next, we will propose an algorithm to update the parameter vector Φ at the visit to the state i^*.

When reaching the $(m+1)$th visit to the recurrent state i^* on time k_{m+1}, derived from (17) we have:

$$\Phi_{m+1}=\Phi_m+\omega_m F_m\left(\Phi_m,\widetilde{\chi}_m\right) \tag{25}$$

$$\widetilde{\chi}_{m+1}=\widetilde{\chi}_m+C\omega_m\sum_{k'=k_m}^{k_{m+1}-1}\left(\wp\left(i_{k'},a_{k'}\right)-\widetilde{\chi}_m\right) \tag{26}$$

where

$$F_m\left(\Phi_m,\widetilde{\chi}_m\right)=\sum_{k'=k_m}^{k_{m+1}-1}\tilde{v}_{\Phi_m}\left(i_{k'},a_{k'}\right)\frac{\nabla\mu_{\Phi_m}\left(i_{k'},a_{k'}\right)}{\mu_{\Phi_m}\left(i_{k'},a_{k'}\right)} \tag{27}$$

$$\tilde{v}_{\Phi_m}\left(i_{k'},a_{k'}\right)=\sum_{k=k'}^{k_{m+1}-1}\left(\wp\left(i_k,a_k\right)-\widetilde{\chi}_m\right) \tag{28}$$

In the algorithm, C stands for a positive constant. $\nabla\mu_{\Phi_m}\left(i_{k'},a_{k'}\right)$ is the gradient of the randomized parameterized policy function defined in (6). And the algorithm helps to update Φ and estimated version average throughput $\widetilde{\chi}$. Although the algorithm above updates the value of the parameter vector Φ at the next visit to the state i^*, we still store all the values of $\tilde{v}_{\Phi_m}\left(i_{k'},a_{k'}\right)$ and $\dfrac{\nabla\mu_{\Phi_m}\left(i_k,a_k\right)}{\mu_{\Phi_m}\left(i_k,a_k\right)}$. While this method would be too complicated, we need to modify it to optimize the algorithm.

By combining (27) and (28) we derive out

$$F_m\left(\Phi_m,\widetilde{\chi}_m\right)=\sum_{k'=k_m}^{k_{m+1}-1}\frac{\nabla\mu_{\Phi_m}\left(i_{k'},a_{k'}\right)}{\mu_{\Phi_m}\left(i_{k'},a_{k'}\right)}\sum_{k=k'}^{k_{m+1}-1}\left(\wp\left(i_k,a_k\right)-\widetilde{\chi}_m\right)$$
$$=\sum_{k'=k_m}^{k_{m+1}-1}\left(\wp\left(i_k,a_k\right)-\widetilde{\chi}_m\right)h_{k+1} \tag{29}$$

where

$$h_{k+1}=\begin{cases}\dfrac{\nabla\mu_{\Phi_m}\left(i_k,a_k\right)}{\mu_{\Phi_m}\left(i_k,a_k\right)},\text{if } k=k_m\\ h_k+\dfrac{\nabla\mu_{\Phi_m}\left(i_k,a_k\right)}{\mu_{\Phi_m}\left(i_k,a_k\right)},\text{if } k=k_m+1,\ldots,k_{m+1}-1\end{cases} \tag{30}$$

All in all, proposed algorithm can be expressed as follows. That is, at the time k, the state is i_k, and Φ_k, h_k, $\tilde{\chi}_k$ are accessible during the iteration and update as:

$$h_{k+1}=\begin{cases}\dfrac{\nabla\mu_{\Phi_k}\left(i_k,a_k\right)}{\mu_{\Phi_k}\left(i_k,a_k\right)},\text{if } i_k=i^*\\ h_k+\dfrac{\nabla\mu_{\Phi_k}\left(i_k,a_k\right)}{\mu_{\Phi_k}\left(i_k,a_k\right)},\text{if else}\end{cases} \tag{31}$$

$$\Phi_{k+1}=\Phi_k+\omega_k\left(\wp\left(i_k,a_k\right)-\widetilde{\chi}_k\right)h_{k+1} \tag{32}$$

$$\widetilde{\chi}_{k+1}=\widetilde{\chi}_k+C\omega_k\left(\wp\left(i_k,a_k\right)-\widetilde{\chi}_k\right) \tag{33}$$

where C is a positive constant. And only Φ and $\tilde{\chi}$ are required for the ST to update for each time.

For now, we have derived the theoretic model for the wireless energy harvesting cognitive industrial sensor networks with jamming attacks, which is by means countable. And we have proved that the algorithm is convergent while it operates efficiently.

5 Numerical results

5.1 Parameters

We perform simulations using Python to evaluate network configurations under different parameter settings. Without loss of generality, we mimic the situation with only one ST under attacks from jammers. As mentioned earlier, we set the data queue capacity as 1 and the energy storage capacity as nine units. The packet arrival probability is set to 0.4, and the data buffer will always be updated with the arriving packet. Also, the queued data in the buffer follows the first in first out policy, and once detect a sent packet is not received correctly, it may be discarded or kept. Either action depends on whether there's an arriving

packet; if there's a newly arriving packet, it will be updated, or it will stay for the next transmission. The reason for the strategy is that in industrial scenarios, we always have sufficient or repetitive data to transmit; the reason we transmit is mostly for updating the immediate status, the newer packet is always better or otherwise, the arriving packet will be deleted before entering the data queue. The energy consumption rates are 1 unit for deception, 3 units to transmit data, and 6 units to perform an attack. The energy harvest probability is set to 0.5, the sensing error probabilities of the ST are both set to 0.01, and the successful transmit probability of the ST (without attack) is set to 0.9. Also, we define there are 3 jammers, and they operate independently.

We compare different performance metrics between our proposed learning algorithm and the standard 1-persistent policy. The ST will transmit data when the channel is idle, and the ST has both data and energy. If a collision happens, the ST will retransmit the packet until it has been successfully received.

5.2 Simulation results

Convergence. We first compare the convergence of the learning algorithm and the 1-persistent algorithm. We evaluate the scenario with four jammers, and the simulation results are shown in Fig.1. We can derive from the graph that the learning algorithm convergent around 9×10^4 iterations; in other words, the throughput converges to 0.09 (average). While compared to the ST adopted 1-persistent algorithm, the average throughput is nearly tripled.

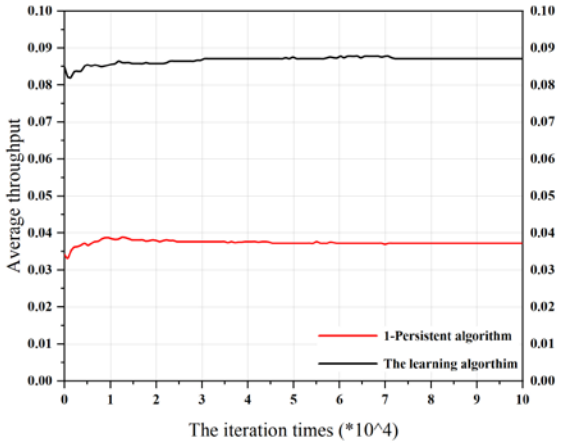

Fig. 1. The average throughput of the learning algorithm and the 1-Persistent algorithm under attacks with three jammers.

Optimality. Since the primary network is time-variant, the effectiveness of energy harvest may change. To mimic the impact, here we change the settings on the energy harvest probabilities of the ST and jammers. To make it fair and comparable, we set the energy harvest probability for the ST and jammer as the same. Then the performance metrics we evaluate are on domains of the average throughput and average update delay of the system. As is shown in the graph Fig.2, the average throughput of the ST increases as energy harvesting probability rises. Also, we can draw from graph Fig.3 that the average update delay of the learning algorithm decreases as the energy harvesting probability rises, while the average update delay of the 1-persistent algorithm decreases significantly but increase a little bit in the end; this is owed to the power of deception mechanism that the

system that adopted the learning algorithm benefits remarkably. Also, the performance of the learning algorithm stands out; as we can see from the graph, the average throughput gap between the learning algorithm and the 1-persistent algorithm increases while the energy harvesting probability rises. When almost every device in the system can harvest wireless RF energy every time, the average update delay of the system increases a bit since, with no deception strategy, the jammers will jam more time slots, and the ST will have less opportunity to transmit, thus increasing the time delay.

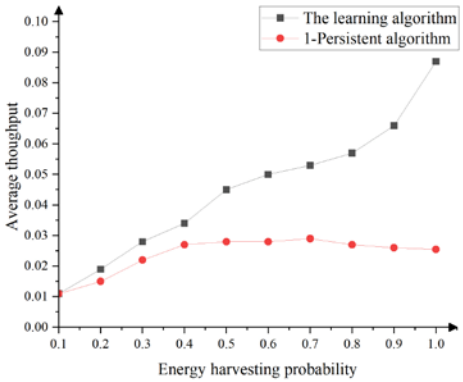

Fig. 2. The average throughput of the ST under various energy harvesting probability settings.

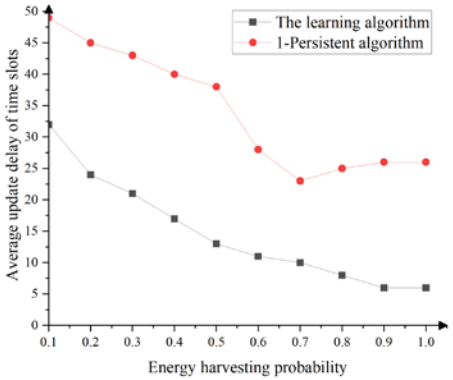

Fig. 3. The average update delay of time slots of the ST under various energy harvesting probability settings.

6 Conclusion

In the paper, we have investigated a secondary wireless sensor network operating on a channel allocated to the primary network while harvesting energy from the ambient wireless RF signals. The STs can recharge wirelessly and use the energy adopted to transmit. A learning algorithm is proposed in this work to ease the tension caused by jamming attacks and upgrade the performance metrics. The learning method not only helps the ST to defend against jammer attacks but also helps to execute proper actions to benefit the overall throughput online. Simulation reveals that the proposed learning algorithm is convergent and helps the ST maximize the overall throughput compared to the traditional

algorithm. In future research work, we will extend the scenario with multiple STs and multiple channels while comparing the system performance with other algorithms.

This research was financially supported by China Scholarship.

References

1. A. Sultan, "Sensing and Transmit Energy Optimization for an Energy Harvesting Cognitive Radio," *IEEE Wireless Communication Letters,* vol. 1, no. 5, pp. 500-503, 2012.
2. X. Gao, W. Xu, S. Li, and J. Lin, "An online energy allocation strategy for energy harvesting cognitive radio systems," in *International Conference on Wireless Communications & Signal Processing*, 2013.
3. S. Park, H. Kim, and D. Hong, "Cognitive Radio Networks with Energy Harvesting," *IEEE Transactions on Wireless Communications,* vol. 12, no. 3, pp. 1386-1397, 2013.
4. S. Park and D. Hong, "Optimal Spectrum Access for Energy Harvesting Cognitive Radio Networks," *IEEE Transactions on Wireless Communications,* vol. 12, no. 12, pp. 6166-6179, 2013.
5. D. T. Hoang, D. Niyato, P. Wang, and D. I. Kim, "Opportunistic Channel Access and RF Energy Harvesting in Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications,* vol. 32, no. 11, pp. 2039-2052, 2014.
6. Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *2010 IEEE symposium on security and privacy*, 2010, pp. 286-301: IEEE.
7. A. Zilinskas, "Simulation-based optimization: Parametric optimization techniques and reinforcement learning," *Interfaces,* vol. 35, no. 6, p. 535, 2005.
8. P. Marbach and J. N. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Transactions on Automatic Control,* vol. 46, no. 2, pp. 191-209, 2001.
9. O. Buffet, A. Dutech, and F. Charpillet, "Shaping multi-agent systems with gradient reinforcement learning," *Autonomous Agents and Multi-Agent Systems,* vol. 15, no. 2, pp. 197-220, 2007.
10. D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society,* vol. 48, no. 3, pp. 334-334, 1997.