Performance Optimization for Cooperative Multiuser Cognitive Radio Networks with RF Energy Harvesting Capability

Hoang, Dinh Thai; Niyato, Dusit; Wang, Ping; Kim, Dong In

2015

Hoang, D. T., Niyato, D., Wang, P., & Kim, D. I. (2015). Performance Optimization for Cooperative Multiuser Cognitive Radio Networks with RF Energy Harvesting Capability. IEEE Transactions on Wireless Communications, 14(7), 3614-3629.

https://hdl.handle.net/10356/89362

https://doi.org/10.1109/TWC.2015.2408610

Performance Optimization for Cooperative Multiuser Cognitive Radio Networks with RF Energy Harvesting Capability

Dinh Thai Hoang[1], Dusit Niyato[1], Ping Wang[1], and Dong In Kim[2]

[1] School of Computer Engineering, Nanyang Technological University (NTU), Singapore

[2] School of Information and Communication Engineering, Sungkyunkwan University (SKKU), Korea

Abstract

We study the performance optimization problem for a cognitive radio network with radio frequency (RF) energy harvesting capability for secondary users. In such networks, the secondary users are able to not only transmit a packet on a channel licensed to a primary user when the channel is idle, but also harvest RF energy from the primary users' transmissions when the channel is busy. Specifically, we propose a system model where the secondary users are able to cooperate to maximize the overall network throughput through sensing a set of common channels. We first consider the case where the secondary users cooperate in a TDMA fashion and propose a novel solution based on a learning algorithm to find optimal channel access policies for the secondary users. Then, we examine the case where the secondary users cooperate in a decentralized manner and we formulate the cooperative decentralized optimization problem as a decentralized partially observable Markov decision process (DEC-POMDP). To solve the cooperative decentralized stochastic optimization problem, we apply a decentralized learning algorithm based on the policy gradient and the Lagrange multiplier method to obtain optimal channel access policies. Extensive performance evaluation is conducted and it shows the efficiency and the convergence of the learning algorithms.

Index Terms

RF energy harvesting, cognitive radio, Markov decision process, decentralized systems, learning algorithm.

I. Introduction

In conventional energy-constrained wireless communications systems (e.g., wireless sensor networks), energy supply for wireless nodes is always a crucial concern [1]. Traditionally, wireless nodes operate based on batteries that require physical charging or replacement to supply enough energy for their operations and data transmissions. This not only increases the maintenance cost, but also degrades the network performance if there is not sufficient energy supply to the wireless nodes. Recently, energy harvesting has emerged and become an effective way to enable long-term and maintenance-free operations of the wireless nodes. There are many forms of energy harvesting including solar, radio frequency (RF), wind, and vibration. Energy harvesting techniques can be adopted in various wireless networks, including cognitive radio networks [2] which can improve spectrum utilization and network performance, and also enhance energy efficiency of secondary users.

Among available energy harvesting techniques, RF energy harvesting is considered as a particularly appropriate solution in a cognitive radio environment because of the following advantages. Firstly, with the rapid development of communications systems worldwide, RF sources are available almost everywhere (e.g., base stations, access points, or even mobile phones), providing pervasive energy supply for secondary users. Secondly, RF energy can propagate and transfer over distance, allowing flexibility for mobile secondary users to gain energy supply. Thirdly, RF energy is broadcast in all directions, and hence multiple secondary users can benefit from the same RF source. Fourthly, the RF energy is controllable by adjusting transmit power at the RF sources. Despite many benefits, using RF energy harvesting in the cognitive radio networks is not straightforward. RF energy becomes a precious resource that needs to be jointly managed and optimized with radio spectrum. Although other forms of energy harvesting have been used in cognitive radio networks, the resource management schemes developed for them cannot be directly adopted for RF energy harvesting. Therefore, this paper aims to study the performance optimization problem for the radio and energy resource management in the RF energy harvested cognitive radio networks.

For the RF energy harvesting cognitive radio network under consideration, there are four different cases depending on the number of secondary users and available channels in the network as follows:

- *Single secondary user and single channel (SUSC) case:* This is the simplest case where a secondary user will always sense a channel. In this case, the problem is straightforward. If the sensed channel is busy due to the transmission of the primary user, the secondary user has an opportunity to harvest and store RF energy in its energy storage. By contrast, if the sensed channel is idle, then the secondary user has a chance to transmit its packet.

- *Single secondary user and multiple channels (SUMC) case:* There is only one secondary user, but there are multiple available channels. In practice, the secondary user cannot sense all channels at the same time (e.g., due to hardware limitation) and thus the secondary user faces the problem of selecting a channel to sense. Different channels may have different idle channel probability and primary user's signal strength. Therefore, the secondary user has to make a decision of channel selection to maximize its performance (e.g., maximum throughput or minimum average number of packets waiting in the data queue). This case was studied in [3] with both online and offline methods.
- *Multiple secondary users and single channel (MUSC) case:* In this case, we have multiple secondary users sharing one common channel. If the channel is busy, then all secondary users can harvest RF energy from the busy channel. However, if the secondary users sense the channel to be idle, then the secondary users face the multiple access channel (i.e., to transmit a packet or not). To avoid the undesirable transmission collision, caused when there are more than one secondary user transmitting data simultaneously, which degrades spectrum utilization and wastes energy, we can use multiple access schemes such as backoff algorithm [4] or time division multiple access (TDMA) technique [5]. More solutions for the channel multiple-access problem in wireless networks can be found in [2], [5].
- *Multiple secondary users and multiple channels (MUMC) case:* This is the most complex case and it is also the main focus of this paper. In this case, we face not only the channel selection problem (as in the SUMC case), but also the multiple access channel problem (as in the MUSC case) at the same time. Additionally, along with incomplete information from the environment and multiple decision makers from secondary users, finding the optimal control policy for secondary users is challenging.

In this paper, we consider the RF energy harvesting cognitive radio network with multiple secondary users and multiple available primary channels. In this network, secondary users are able to harvest RF energy from the busy channel and this harvested energy will be used to transmit data over an idle channel. Furthermore, we consider the case in which the secondary users cooperate in order to maximize the network performance in terms of the average throughput of the network. The secondary users are assumed to have no priori information of primary channels and they also cannot sense all channels simultaneously. Consequently, at decision epochs, the secondary users need to select one of channels to sense such that the average throughput for the system is maximized. To address the cooperative optimization problem among the secondary users, we propose two approaches as follows. In the first approach, we consider the case where the secondary users are cooperative in a round-robin fashion and each secondary user is equipped with an online learning algorithm in order to help secondary users explore the environment (i.e., primary channels) and then to make optimal decisions. In the second approach, secondary users are assumed to cooperate in a decentralized fashion without coordination. In this case, we formulate the cooperative optimal problem for secondary users as the decentralized partially observable Markov decision process model [6] with the aim of obtaining the decentralized optimal channel access policies for secondary users. Moreover, to deal with the curse of model and the curse of dimensionality (caused by multiple decision makers and incomplete information) of the decentralized systems, we study a decentralized learning algorithm developed based on the policy gradient and the Lagrange multiplier methods. These methods can work without prior environment parameters (e.g., the idle channel probability, the channel sensing error probability, and the successful packet transmission probability). Extensive performance evaluation and comparison are performed to show the efficiency as well as the convergence of the learning algorithms.

The rest of the paper is organized as follows. In Section II, we present an extensive review of the related work. Section III introduces the system model together with the assumptions considered in this paper. Section IV studies a learning solution based on the TDMA technique and Section V investigates a decentralized learning algorithm. Experiments are performed and results are analyzed in Section VI. We then conclude the paper in Section VII.

II. Related Work

A. Cognitive Radio Networking with Energy Harvesting

There were some research works studying different issues of cognitive radio networks with different forms of energy harvesting. For example, [7], [8] and [9] studied using energy harvested from general sources (e.g., ambient environment energy sources) for a specific secondary user in cognitive radio networks. Specifically, in [7], the authors considered an offline method based on a Markov decision process (MDP) to find optimal access policies for secondary users. Conversely, in [8], the authors used an online method to obtain a real-time adaptive energy management policy for a particular secondary user. Additionally, to avoid the high complexity of optimal solutions, the authors in [8] proposed a suboptimal solution that can reduce the complexity while helping secondary users to make decisions based on current and past information. In [9], the authors aimed to find the optimal detection threshold in order to maximize the secondary user's throughput under the energy causality constraint and the collision constraint. The analytical and numerical results show that, when the energy arrival rate is lower than expected energy consumption, the number of channel access attempts is limited by the amount of harvested energy and the initially available energy. Unlike [7], [8], and [9], the authors of [10] studied a method to exploit energy from polarization in cooperative cognitive radio networks. The authors first presented a framework for cooperative communications and introduced polarization signal processing technique to avoid interference between secondary and primary users. Then, the MDP framework and dynamic programming technique were used to maximize the throughput for secondary users.

B. RF Energy Harvesting in Cognitive Radio Networks

Recently, the RF energy harvesting technology has been improved with a significant efficiency for wireless nodes [11], [12], [13]. Cognitive radio is one of the target applications of such technology. For example, in [14], the authors considered using the cognitive radio network with RF energy harvesting capability for body area networks. The authors first presented the architecture of a BodyNet that allows secondary users to harvest energy from ambient radio signals. Then, routing protocols were proposed to support data collection in the BodyNet. In [15], the authors introduced a cognitive radio network architecture that allows secondary users to harvest RF energy when they move close to primary transmitters and opportunistically access channel when they are sufficiently far away from primary transmitters. Based on the proposed architecture, the authors then developed an analytical model to analyze the performance for secondary users in the network. In [16] and [17], the authors studied the throughput optimization problem for RF energy harvesting cognitive radio networks by using partially observable Markov decision process (POMDP) framework. While [16] uses POMDP to find an optimal policy (i.e., to access channel or to harvest energy) for a secondary user, the optimal policy in [17] is to determine whether a secondary user senses the target channel or remains idle. Moreover, in [17], energy causality and collision problems were also taken into account. To address the POMDP with constraints, the authors of [17] converted the constrained POMDP into unconstrained POMDP and used an approximation solution to find an optimal solution for the secondary user. However, in both [16] and [17], the authors considered the throughput optimization problem for an RF energy harvesting cognitive radio network with just one secondary user and one licensed channel. The channel selection problem has been considered in our previous work [18], [3]. In [18] and [3], we studied a model with a secondary user sensing a number of licensed channels and thus we need to determine which channel should be sensed by the secondary user at the decision time. In [18], we formulated an MDP model and used an offline method to find an optimal policy for the secondary user in the case when the secondary user has incomplete information about the channels. In [3], we developed an online method that allows to find an optimal solution for the secondary user in the case when the secondary user has no information about the channels.

Different from all existing works, to the best of our knowledge, the performance optimization problem for an RF energy harvesting cognitive radio network with multiple channels and multiple secondary users still has not been studied yet, and hence it is the main goal of this paper.

III. System Model and Assumptions

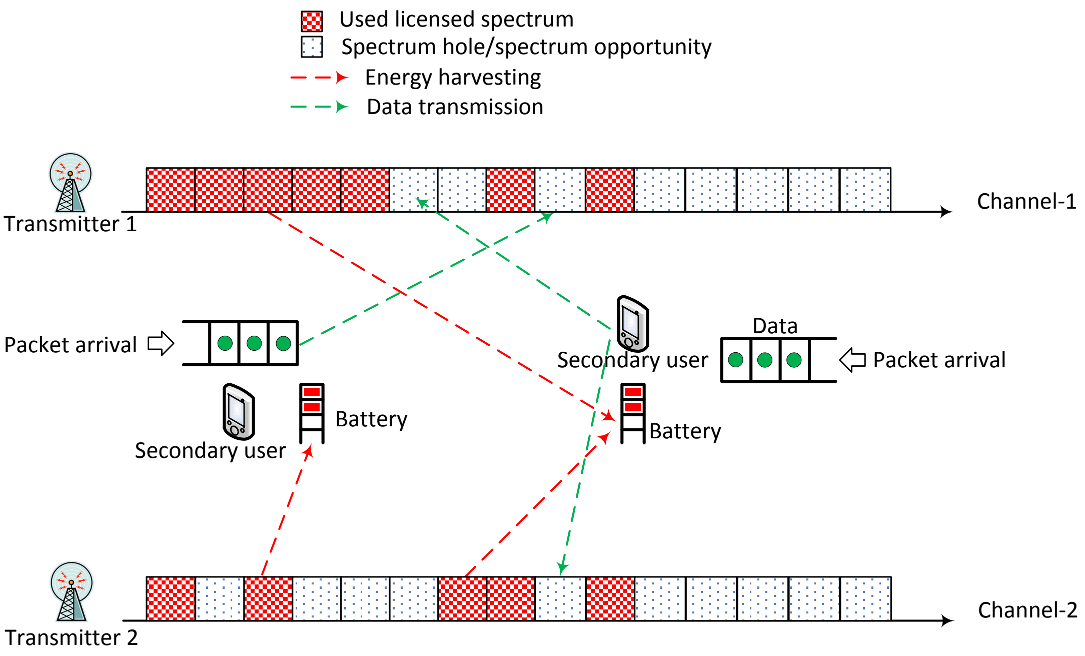


Fig. 1. System model.

We consider RF energy harvesting cognitive radio networks with multiple secondary users and multiple channels as shown in Fig. 1. In this system model, there are N secondary users sharing M channels that are allocated to primary users. The primary users use the channels to transmit data in a time slot basis. The secondary users are assumed to be equipped with a data queue and an energy storage. The data queue is used to store data generated or collected from other sensors and the maximum size of the data queue for secondary user n, for $n \in \{1, \dots, N\}$ is denoted by $\mathscr{D}_n$. The energy storage is a battery which can store RF energy harvested from radio signal[1] and the maximum capacity of the energy storage of secondary user n is denoted by $\mathscr{E}_n$.

[1] The design and implementation of circuits, antenna, and process of RF energy harvesting are beyond the scope of this paper. The readers may refer to [12], [19] for more details.

In each time slot, the probability of a packet arriving at the data queue of secondary user n is denoted by λ_n. The secondary user utilizes the energy in its energy storage for packet transmission. The amount of harvested RF energy by the secondary user depends on the channel status. Specifically, if the secondary user n senses the channel m and the channel m is busy, then secondary user n can harvest a unit of energy with a certain probability (i.e., a successful RF energy harvesting probability). However, if the channel m is idle, then secondary user n cannot harvest any energy. Conversely, the secondary user n can transmit a packet on the channel m or do nothing. If the receiver of the secondary user receives the transmitted packet successfully (i.e., no collision and no error), the receiver will send back an ACK to the secondary user. Upon receiving the ACK, the secondary user then removes this packet from its data queue. By contrast, if the packet transmission is unsuccessful, the packet is still in the data queue and the secondary user has to re-transmit the packet later. The unsuccessful packet transmission happens due to collision with primary user's transmission or other secondary users' transmissions, or channel error.

The channel sensing can be in error. The false alarm sensing error happens when a channel is idle, but the secondary user senses it to be busy. The miss detection happens when a channel is busy, but the secondary user senses it to be idle. The probabilities of such sensing error events are called false alarm probability and miss detection probability.

In the following, we consider two multiple access schemes for the secondary users. Firstly, the secondary users can access a channel based on a TDMA fashion in which a round-robin scheduling is applied. However, this scheme requires coordination among secondary users, e.g., to determine the sequence of packet transmissions. Alternatively, the secondary users can perform random access in a decentralized fashion. However, in this scheme, collision among secondary users can happen. We derive the optimal channel access policies and perform performance comparison for both schemes.

IV. A TDMA Learning Algorithm for RF Energy Harvesting Cognitive Radio Networks

We first consider TDMA for multiple accesses of secondary users. In particular, the secondary users are scheduled for channel access in a round-robin fashion. We propose a learning algorithm for the secondary users to make optimal decisions to adapt with environment conditions.

A. Problem Formulation

We define the state space of the secondary user n as follows[2]:

$$\mathcal{S} = \Big\{(e, d, \vartheta); e \in \{0, 1, \dots, \mathscr{E}\}, d \in \{0, 1, \dots, \mathscr{D}\}, \vartheta \in \{1, \dots, N\}\Big\}, \tag{1}$$

where e, d, and ϑ represent the energy level of the energy storage, the number of packets in the data queue, and the time schedule of the secondary user, respectively. $\mathscr{E}$ is the maximum capacity of the energy storage and $\mathscr{D}$ is the maximum data queue size. The time schedule variable ϑ is a sequence from 1 to N for channel access allocated to secondary users. The secondary user n will be allowed to access the channel when $\vartheta = n$. The state of secondary user n is then defined as a composite variable $s = (e, d, \vartheta) \in \mathcal{S}$.

$\mathcal{A}$ is the action space of secondary user n, which is a set of available channels for secondary users to select from. Then, at each time slot, each secondary user has to make a decision $a \in \mathcal{A} = \{0, 1, \dots, M\}$ to select one of the channels to sense. $a = 0$ means that the secondary user n does not select any channel.

We define the immediate reward function for each secondary user as follows:

$$\mathscr{T}(s, a) = \begin{cases} 1, & \text{if a packet is successfully transmitted} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We then consider a randomized parameterized policy [20], [21], [22]. Under the randomized parameterized policy, when secondary user n is at state s, the secondary user will select action a with the probability $\mu_\Theta(s, a)$ given as follows:

$$\mu_\Theta(s, a) = \frac{\exp\big(\theta_{s,a}\big)}{\sum_{a_i \in \mathcal{A}} \exp\big(\theta_{s,a_i}\big)}, \tag{3}$$

where $\Theta = \{\theta_{s,a} \in \mathbb{R}\}$ is the parameter vector of secondary user n at state s. Additionally, every $\mu_\Theta(s, a)$ must not be negative and $\sum_{a \in \mathcal{A}} \mu_\Theta(s, a) = 1$.

The objective of the policy is to maximize the average throughput of the secondary user under the randomized parameterized policy $\mu_\Theta(s, a)$ which is denoted by $\Psi(\Theta)$. By using the randomized parameterized policy $\mu_\Theta(s, a)$, the transition probability function will be parameterized as follows:

$$\mathbf{p}(s'|s, \Psi(\Theta)) = \sum_{a \in \mathcal{A}} \mu_\Theta(s, a)\mathbf{p}(s'|s, a), \tag{4}$$

[2] The TDMA technique is used in this case to avoid the collision among secondary users, and thus, the problem formulation for every secondary user is the same. Therefore, we omit the indicator of secondary users in this section for brevity of presentation.

for all $s, s' \in \mathcal{S}$, where $\mathbf{p}(s'|s, a)$ is the transition probability from state s to state s' when action a is taken. Similarly, we have the parameterized immediate throughput function defined as follows:

$$\mathscr{T}(s, \Theta) = \sum_{a \in \mathcal{A}} \mu_\Theta(s, a) \mathscr{T}(s, a). \tag{5}$$

The objective of the policy is to maximize the average throughput of the secondary user under the randomized parameterized policy $\mu_\Theta(s, a)$, which is denoted by $\Psi(\Theta)$.

Then we need to make some necessary assumptions as follows.

Assumption 1. *The Markov chain is aperiodic and there exists a state s^* which is recurrent for each of such Markov chain.*

Assumption 2. *For every state $s, s' \in \mathcal{S}$, the transition probability $\mathbf{p}(s'|s, \Psi(\Theta))$ and the immediate throughput function $\mathscr{T}(s, \Theta)$ are bounded, twice differentiable, and have bounded first and second derivatives.*

Assumption 1 implies that the system has a Markov property. Assumption 2 ensures that the transition probability function and the immediate reward function depend "smoothly" on Θ. This assumption is important when we apply gradient methods for adjusting Θ.

Then, we can define the parameterized average throughput (i.e., the throughput under the parameter vector Θ) by

$$\psi(\Theta) = \lim_{t \to \infty} \frac{1}{t} \mathbb{E}_\Theta \Big[\sum_{k=0}^{t} \mathscr{T}(s_k, \Theta_k) \Big], \tag{6}$$

where s_k is the state of the secondary user at time step k. $\mathbb{E}_\Theta[\cdot]$ is the expectation under parameter vector Θ. Under Assumption 1, the average throughput $\psi(\Theta)$ is well defined for every Θ, and does not depend on the initial state Θ_0. Moreover, we have the following balance equations

$$\begin{aligned} &\sum_{s \in \mathcal{S}} \pi_\Theta(s) \mathbf{p}(s'|s, \Psi(\Theta)) = \pi_\Theta(s'), \text{ for } s' \in \mathcal{S}, \\ &\sum_{s \in \mathcal{S}} \pi_\Theta(s) = 1, \end{aligned} \tag{7}$$

where $\pi_\Theta(s)$ is the steady-state probability of state s under the parameter vector Θ. These balance equations have a unique solution defined as a vector $\Pi_\Theta = \begin{bmatrix} \cdots & \pi_\Theta(s) & \cdots \end{bmatrix}^\top$. Then, the average throughput can be expressed as follows:

$$\psi(\Theta) = \sum_{s \in \mathcal{S}} \pi_\Theta(s) \mathscr{T}(s, \Theta). \tag{8}$$

B. Learning Algorithm Based on Policy Gradient Method

To update the parameter vector Θ, we will use the algorithm based on the gradient method as introduced in [25] as follows:

$$\Theta_{k+1} = \Theta_k + \rho_k \nabla \psi(\Theta_k), \tag{9}$$

where ρ_k is a step size and $\nabla \psi(\Theta_k)$ is the gradient of average throughput. Under a suitable step size satisfying Assumption 3 and Assumption 1, it is proved that $\lim_{k \to \infty} \nabla \psi(\Theta_k) = 0$ and thus $\psi(\Theta_k)$ converges [25].

Assumption 3. *The step size ρ_k is deterministic, nonnegative and satisfies the following conditions,*

$$\sum_{k=1}^{\infty} \rho_k = \infty, \ \textit{and} \ \sum_{k=1}^{\infty} (\rho_k)^2 < \infty. \tag{10}$$

We then propose Proposition 1 to calculate the gradient of the average throughput as follows:

Proposition 1. *Let Assumption 1 and Assumption 2 hold, then*

$$\nabla \psi(\Theta) = \sum_{s \in \mathcal{S}} \pi_\Theta(s) \Big(\nabla \mathscr{T}(s, \Theta) + \sum_{s' \in \mathcal{S}} \nabla \mathbf{p}(s'|s, \Psi(\Theta)) d(s', \Theta) \Big), \tag{11}$$

where $d(s', \Theta)$ is the differential throughput at state s'. In general, we can define the differential throughput at state s as follows:

$$d(s, \Theta) = \mathbb{E}_\Theta \left[\sum_{k=0}^{T-1} \left(\mathscr{T}(s_k, \Theta) - \psi(\Theta) \right) | s_0 = s \right], \tag{12}$$

where $T = \min\{k > 0 | s_k = s^*\}$ is the first future time that state s^* is visited. Here, we need to note that, the main aim of defining the differential throughput $d(s, \Theta)$ is to represent the relation between the average throughput and the immediate throughput at state s, instead of the recurrent state s^*. Additionally, under Assumption 1, the differential throughput $d(s, \Theta)$ is a unique solution of the following Bellman equation defined as follows:

$$d(s, \Theta) = \mathscr{T}(s, \Theta) - \psi(\Theta) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s'|s, \Psi(\Theta)) d(s', \Theta), \tag{13}$$

for all $s \in \mathcal{S}$.

Proposition 1 presents the gradient of the average throughput $\psi(\Theta)$ and the proof of Proposition 1 can be done in a similar way as Appendix A of [3].

We then propose a learning algorithm that allows secondary users to update the parameter vector Θ at each time step based on estimating the gradient of average throughput (Algorithm 1).

Algorithm 1 Algorithm to update Θ at every time step

At time step k, the state is s_k, and the values of Θ_k, z_k, and $\widetilde{\psi}(\Theta_k)$ are available from the previous iteration. We update z_k, Θ_k, and $\widetilde{\psi}$ according to:

$$z_{k+1} = \begin{cases} \frac{\nabla \mu_{\Theta_k}(s_k, a_k)}{\mu_{\Theta_k}(s_k, a_k)}, & \text{if } s_k = s^* \\ z_k + \frac{\nabla \mu_{\Theta_k}(s_k, a_k)}{\mu_{\Theta_k}(s_k, a_k)}, & \text{otherwise,} \end{cases} \tag{14}$$

$$\Theta_{k+1} = \Theta_k + \rho_k(\mathscr{T}(s_k, a_k) - \widetilde{\psi}_k) z_{k+1}, \tag{15}$$

$$\widetilde{\psi}_{k+1} = \widetilde{\psi}_k + \kappa \rho_k(\mathscr{T}(s_k, a_k) - \widetilde{\psi}_k). \tag{16}$$

In Algorithm 1, κ is a positive constant, ρ_k is the step size of the algorithm, z_k is an auxiliary variable to compute the updated value of $\frac{\nabla \mu_{\Theta_k}(s_k, a_k)}{\mu_{\Theta_k}(s_k, a_k)}$, and $\widetilde{\psi}(\Theta_k)$ is the estimated value of the average throughput. The convergence of Algorithm 1 can be proved in a similar way as Appendix B of [3].

V. A Decentralized Solution for RF Energy Harvesting Cognitive Radio Networks

In this section, we consider the case where secondary users cooperate in a decentralized manner. We first formulate the cooperative optimization problem as decentralized partially observable Markov decision process [6] (DEC-POMDP) and then examine a decentralized learning algorithm to obtain optimal policies for secondary users.

A. Optimization Formulation

We formulate the optimization problem for the RF energy harvesting cognitive radio network with multiple secondary users and multiple channels (i.e., the MUMC case) as a DEC-POMDP in a discrete time system. A general DEC-POMDP model can be defined as a tuple $< N, \mathcal{S}, \mathcal{A}, \mathbf{p}, g, \mathcal{O}, \mathbf{o} >$, where

- N is the total number of secondary users,
- $\mathcal{S}$ is a finite set of states and it is known as the global state space of the network,
- $\mathcal{A}$ is a finite set of joint actions,
- $\mathbf{p}$ is a joint transition probability function,
- g is the global immediate cost function,
- $\mathcal{O}$ is a finite set of joint observations, and
- $\mathbf{o}$ is a joint observation probability function.

1) State Space: We define $\mathcal{S} \triangleq (\mathcal{S}_1 \times \cdots \times \mathcal{S}_n \times \cdots \times \mathcal{S}_N)$, as the global system state space where $\mathcal{S}_n$ is the local state space of secondary user n.

We define the state space of the secondary user n as follows:

$$\mathcal{S}_n = \Big\{(e_n, d_n); e_n \in \{0, 1, \ldots, \mathscr{E}_n\}, d_n \in \{0, 1, \ldots, \mathscr{D}_n\}, \Big\} \tag{17}$$

where e_n, and d_n represent the energy level of the energy storage, and the number of packets in the data queue, respectively. Again, $\mathscr{E}_n$ is the maximum capacity of the energy storage and $\mathscr{D}_n$ is the maximum data queue size. The state of secondary user n is then defined as a composite variable $s_n = (e_n, d_n) \in \mathcal{S}_n$.

2) Action Space: The joint action space $\mathcal{A}$ is a composition of sets of local action spaces from secondary users. The joint action space can be defined as follows

$$\mathcal{A} \triangleq (\mathcal{A}_1 \times \cdots \times \mathcal{A}_n \times \cdots \times \mathcal{A}_N), \tag{18}$$

where $\mathcal{A}_n$ is the local state space of secondary user n that is a set of available channels for secondary users to select from. Then, at each time slot, each secondary user has to make a decision $a_n \in \mathcal{A}_n = \{0, 1, \ldots, M\}$ to select one of the channels to sense. $a_n = 0$ means that the secondary user n does not select any channel.

3) Transition Probability Function: The transition probability matrix for secondary users n can be expressed as follows:

$$\mathbf{P}_n(a_n) = \left[\begin{array}{ccccc} \mathbf{B}^{n}{}_{0,0}(a_n) & \mathbf{B}^{n}{}_{0,1}(a_n) & & & \\ \mathbf{B}^{n}{}_{1,0}(a_n) & \mathbf{B}^{n}{}_{1,1}(a_n) & \mathbf{B}^{n}{}_{1,2}(a_n) & & \\ & \ddots & \ddots & \ddots & \\ & & & \mathbf{B}^{n}{}_{\mathscr{D}_n,\mathscr{D}_n-1}(a_n) & \mathbf{B}^{n}{}_{\mathscr{D}_n,\mathscr{D}_n}(a_n) \end{array}\right] \begin{array}{l} \leftarrow d_n = 0 \\ \leftarrow d_n = 1 \\ \vdots \\ \leftarrow d_n = \mathscr{D}_n \end{array}, \tag{19}$$

where each row of matrix $\mathbf{P}_n(a_n)$ corresponds to the number of packets in the data queue. Each element of matrix $\mathbf{P}_n(a_n)$, i.e., matrix $\mathbf{B}^n_{d_n,d^{'}_n}(a_n)$, corresponds to the transition of the data queue state from d_n in the current time slot to $d^{'}_n$ in the next time slot. Then, similar to $\mathbf{P}_n(a_n)$, we will construct matrix $\mathbf{B}^n_{d_n,d^{'}_n}(a_n)$, where each element of matrix $\mathbf{B}^n_{d_n,d^{'}_n}(a_n)$ represents the transition of the energy level from state e_n to state $e^{'}_n$. There are two cases to derive matrix $\mathbf{B}^n_{d_n,d^{'}_n}(a_n)$, i.e., for $d_n = 0$ and $d_n > 0$. For $d_n = 0$, since the data queue is empty, the energy level will never decrease and it only increases when the sensed channel is busy, no miss detection sensing error, and the secondary user harvests energy successfully. In the case of $d_n > 0$, we have to consider three sub-cases, namely, when the number of packets decreases, remains the same, or increases. The number of packets decreases when the sensed channel is idle, there is no false alarm sensing error, there is no arrival packet, the energy storage is not empty, and the packet is transmitted successfully. Similarly, for the rest of cases, we will consider specific sub-cases and construct corresponding matrices for these cases. Then we can construct a joint transition probability matrix for the system from all transition probability matrices of secondary users.

However, to derive a joint transition probability matrix for the whole system, we need to know the transition probability matrix of each secondary user. Furthermore, to derive transition probability matrices for secondary users, we need to know environment parameters, e.g., idle channel probability, miss detection probability, false alarm sensing error probability, successful packet transmission probability, and successful RF energy harvesting probability. However, in practice, it is not easy and even impossible to obtain these probabilities for secondary users. Therefore, we propose a learning algorithm that is developed based on the simulation-based method [26]. The general idea of the simulation-based method is based on a "simulator" that can simulate an environment by generating parameters (e.g., idle channel probability, etc) given the system process. After the simulation, the simulator yields the results which will be used by the secondary users to obtain channel access policies. Based on received observations (e.g., a packet is successful transmitted or a unit of energy is successfully harvested), the secondary users can update their local information (i.e., the energy level and the number of packets in the energy queue and data queue, respectively) accordingly.

With the simulation-based learning algorithm, for a given control policy Ψ, the joint transition probability function $\mathbf{p}$ can be derived from the transition probability of the local state of the secondary users (i.e., a queue state, and energy state) as follows:

$$\mathbf{p}\big(s(t+1)|s(t),\Psi\big) = \mathbf{p}_{\text{env}}\mathbf{p}\Big(\big(d(t+1),e(t+1)\big)\big|\big(d(t),e(t)\big),\Psi\Big), \tag{20}$$

where $\mathbf{p}_{\text{env}}$ is the probability function of environment parameters that can be generated by the simulator. $s(t) \in \mathcal{S}$ denotes the joint state of the system at time slot t. $d(t)$ and $e(t)$ denote the joint queue state and the joint energy state at time slot t, respectively. $\mathbf{p}\Big(\big(d(t+1),e(t+1)\big)\big|\big(d(t),e(t)\big),\Psi\Big)$ is the joint transition probability of secondary users and this probability can be derived from the transition probability of the local states (i.e., queue state and energy state of secondary users) as follows:

$$\begin{aligned}&\mathbf{p}\Big(\big(d(t+1),e(t+1)\big)\big|\big(d(t),e(t)\big),\Psi\Big)\\ &=\begin{cases}\prod_{n=1}^{N}\mathbf{p}\big(d_n(t),e_n(t)\big)\mathbf{p}\big(a_n(t)\big), & \text{if } d_n(t+1)=\mathcal{D}_n^*,\;\; e_n(t+1)=\mathcal{E}_n^*\\ 0, & \text{otherwise,}\end{cases}\end{aligned} \tag{21}$$

where $\mathcal{D}_n^* = \min\Big(\big([d_n(t)-\phi_n(t)]^+ + \lambda_n(t)\big),\mathscr{D}_n\Big)$, and $\mathcal{E}_n^* = \min\Big(\big[e_n(t)-c_n(t)]^+ + y_n(t)\big],\mathscr{E}_n\Big)$.

Again, $\mathscr{D}_n$ is the maximum size of the data queue and $\mathscr{E}_n$ is the maximum size of the energy storage of secondary user n. Here, $\phi_n(t)$ is the number of packets transmitted by secondary user n at time step t, $\lambda_n(t)$ is the number of arriving packets, $y_n(t)$ is the amount of energy harvested, and $c_n(t)$ is the amount of energy used at time slot t. Furthermore, $[x]^+ = \max(x, 0)$.

4) Global Immediate Cost Function: In this model, we consider the case when all secondary users cooperate to maximize the network throughput. Thus, the global immediate reward function can be defined as follows:

$$g(s(t), a(t)) = \sum_{n=1}^{N} g(s_n(t), a_n(t)) = \sum_{n=1}^{N} r_n(t). \tag{22}$$

Here, the global immediate reward is the number of packets transmitted successfully at time slot t.

5) Observations and Observation Probability Function: In our system model, the observation of each secondary user is its local information from the data queue and the energy storage. Therefore, the observation of each secondary user is identical to its local state and thus the observation probability function is defined in the same way as in (21).

B. Parameterization for DEC-POMDP

Similar to Section IV, we use a parameterized randomized policy [20], [21], [22]. Under the parameterized randomized policy, when secondary user n is at state s_n, the secondary user will select action a_n with the probability $\mu_{\Theta_n}(s_n, a_n)$ given as follows:

$$\mu_{\Theta_n}(s_n, a_n) = \frac{\exp\left(\theta_{s_n,a_n}\right)}{\sum_{a_i \in \mathcal{A}_n} \exp\left(\theta_{s_n,a_i}\right)}, \tag{23}$$

where $\Theta_n = \{\theta_{s_n,a_n} \in \mathbb{R}\}$ is the parameter vector of secondary user n at state s_n. Moreover, every $\mu_{\Theta_n}(s_n, a_n)$ must not be negative and $\sum_{a_n \in \mathcal{A}_n} \mu_{\Theta_n}(s_n, a_n) = 1$, $\forall n$.

Under the parameterized randomized policies of the secondary users, the joint transition probability function and the average cost criterion can be parameterized as follows:

$$\mathbf{p}(s^{'}|s, \Psi(\Theta)) = \sum_{a \in \mathcal{A}} \mu_{\Theta}(s, a)\mathbf{p}(s'|s, a), \tag{24}$$

$$g(s, \Theta) = \sum_{a \in \mathcal{A}} \mu_{\Theta}(s, a) g(s, a), \tag{25}$$

where $\mu_{\Theta}(s, a) = \prod_{n=1}^{N} \mu_{\Theta_n}(s_n, a_n)$ and Θ is a joint parameter vector of the system.

Then, we can define the parameterized average throughput by

$$\overline{\mathcal{C}}(\Theta) = \lim \sup_{T \to \infty} \frac{1}{T} \mathbb{E}_{\Psi(\Theta)}\left[\sum_{t=1}^{T} g\big(s(t), \Theta)\big)\right] \tag{26}$$

where T is the total time horizon.

In this paper, we assume that secondary users can sense one channel at a time, and thus the energy consumption for the sensing process is negligible. Furthermore, the spectrum sensing processes always consume much less energy than that of the data transmission processes. In the literature, this assumption is often made [23], [24]. However, in some systems which have very limited energy (e.g., sensor networks), the energy consumption of the spectrum sensing process can be significant compared with that used for data transmission, and we have to optimize it. Therefore, we impose the constraints that on average the energy consumption for spectrum sensing must not exceed a threshold $\mathscr{Q}_n^*$ per time slot, i.e.,

$$\overline{\mathscr{Q}}_n \leq \mathscr{Q}_n^*, \quad \forall n. \tag{27}$$

Furthermore, we consider the case when the secondary users cooperate to obtain a joint optimal stationary control policy $\Psi(\Theta)$ to maximize the average throughput for the system. However, the actions of secondary users are made based on their local information. As a result, the optimization problem is partially observable and the control policy $\Psi_n(\Theta_n)$ is a function of a local state only. Consequently, the optimization problem with constraints can be defined as follows:

$$\begin{aligned} \max_{\Theta} \;\; \overline{\mathcal{C}}(\Theta) = \sum_{n=1}^{N} \overline{\mathcal{C}}_n(\Theta) = \lim_{T\to\infty} \sup \frac{1}{T}\mathbb{E}_{\Psi(\Theta)}\Bigg[\sum_{t=1}^{T} g\big(s(t),\Theta)\big)\Bigg], \\ \text{s.t. } \overline{\mathscr{Q}}_n(\Psi(\Theta)) = \overline{\mathscr{Q}}_n(\Theta) \leq \mathscr{Q}_n^*, \;\; \forall n, \end{aligned} \tag{28}$$

where $g(s(t),\Theta)$ is the parameterized common immediate throughput generated at the joint state $s(t)$ after the joint action $a(t)$ is made, and $\overline{\mathcal{C}}(\Theta)$ is the parameterized average total throughput. $\overline{\mathscr{Q}}_n$ denotes the average energy consumption for sensing channels of secondary user n and $\mathscr{Q}_n^*$ denotes the target threshold that we want to control.

C. Lagrange Multiplier and Policy Gradient Method

To solve the optimization problem with constraints as defined in (28), we define a Lagrange function based on the *Lagrange multiplier* method as follows:

$$\mathscr{L}(\Theta,\gamma) = \sum_{n=1}^{N} \Big(\overline{\mathcal{C}}_n(\Theta) + \gamma_n(\overline{\mathscr{Q}}_n(\Theta) - \mathscr{Q}_n^*)\Big), \tag{29}$$

where γ_n is a Lagrange multiplier for the constraint of secondary user n. If we denote $\mathscr{G}\big(s,a\big) = \sum_{n=1}^{N}\Big(g\big(s_n,a_n\big) + \gamma_n\big(\mathscr{Q}_n - \mathscr{Q}_n^*\big)\Big)$ as the immediate value function, then the parameterized immediate value function will be

$$\mathscr{G}\big(s,\Theta\big) = \sum_{a\in\mathcal{A}} \mu_\Theta(s,a)\mathscr{G}\big(s,a\big). \tag{30}$$

We have the following balance equations:

$$\begin{aligned}&\sum_{s\in\mathcal{S}}\pi_s(\Theta)\mathbf{p}(s^{'}|s,\Psi(\Theta))=\pi_{s'}(\Theta),\ \text{for } s'\in\mathcal{S},\\&\sum_{s\in\mathcal{S}}\pi_s(\Theta)=1,\end{aligned}\tag{31}$$

where $\pi_s(\Theta)$ is the steady-state probability of the joint state s under the parameter vector Θ. These balance equations have a unique solution defined as a vector $\Pi_\Theta=\begin{bmatrix}\cdots & \pi_s(\Theta) & \cdots\end{bmatrix}^\top$. Then, the Lagrange function can be represented as follows:

$$\mathscr{L}(\Theta,\gamma)=\sum_{s\in\mathcal{S}}\pi_s(\Theta)\mathscr{G}\big(s,\Theta\big).\tag{32}$$

Then, to solve the Lagrange function, we use *Karush-Kuhn-Tucker* (KKT) conditions [27] to find a local optimal solution Θ^* that satisfies the following conditions:

$$\begin{aligned}&\nabla_\Theta\mathscr{L}(\Theta^*,\gamma^*)=\mathbf{0},\\&\overline{\mathscr{Q}}_n(\Theta^*)-\mathscr{Q}_n^*\leq 0,\\&\gamma_n^*(\overline{\mathscr{Q}}_n(\Theta^*)-\mathscr{Q}_n^*)=0,\forall n,\\&\gamma_n^*\geq 0.\end{aligned}\tag{33}$$

To obtain the gradient of the Lagrange function $\mathscr{L}(\Theta,\gamma)$, we first define the differential cost $q(s,a,\Theta)$ at state s under a control action a as follows

$$q(s,a,\Theta)=\mathbb{E}_{\Psi(\Theta)}\Bigg[\sum_{t=0}^{\mathbf{T}-1}\Big(\mathscr{G}\big(s(t),a(t)\big)-\mathscr{L}(\Theta,\gamma)\Big)\Big|s(0)=s,a(0)=a\Bigg],\tag{34}$$

where $\mathbf{T}=\min\{t>0|s(t)=s^\dagger\}$ is the first future time that state $s^\dagger=(s_1^\dagger,\ldots,s_N^\dagger)$ is visited and $\mathscr{G}\big(s(t),a(t)\big)=\sum_{n=1}^N\Big(g\big(s_n(t),a_n(t)\big)+\gamma_n\big(\mathscr{Q}_n(t)-\mathscr{Q}_n^*\big)\Big)$. $s^\dagger$ can be selected randomly from state space $\mathcal{S}$. Here, $g\big(s_n(t),a_n(t)\big)$ is the number of packets in the data queue and $\mathscr{Q}_n(t)$ is the used energy at time slot t of secondary user n. In (34), $q(s,a,\Theta)$ can be expressed as the differential cost if action a is made based on policy μ_Θ at state s. Then, we can obtain the gradient of the Lagrange function $\mathscr{L}(\Theta,\gamma)$ as in Proposition 2.

Proposition 2. *The gradient of the Lagrangian function is determined as follows:*

$$\nabla_{\Theta_n}\mathscr{L}(\Theta,\gamma)=\ \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\pi_s(\Theta)\mu_\Theta(s,a)\frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a_n)}{\mu_{\Theta_n}(s_n,a_n)}q(s,a,\Theta),\tag{35}$$

where $\pi_s(\Theta)$ is steady state probability of state $s\in\mathcal{S}$ and $\mu_\Theta(s,a)=\prod_{n=1}^N\mu_{\Theta_n}(s_n,a_n)$.

The proof of Proposition 2 is provided in Appendix A.

Then, based on the Proposition 2, secondary users will update their parameter vectors based on the idealized gradient algorithm at each time step t as follows [25]:

$$\Theta_n^{t+1} = \Theta_n^t + \rho_t \nabla_{\Theta_n} \mathscr{L}(\Theta, \gamma), \quad \forall n \tag{36}$$

where ρ_t is a suitable step size that satisfies Assumption 4.

Assumption 4. *The step size ρ_t is deterministic, nonnegative and satisfies the following conditions,*

$$\sum_{t=1}^{\infty} \rho_t = \infty, \text{ and } \sum_{t=1}^{\infty} (\rho_t)^2 < \infty. \tag{37}$$

In other words, the value of the step size to update the parameter vectors approaches zero when the time step goes to infinity.

D. Decentralized Online Learning Algorithm with Communications

With the idealized gradient method, secondary users can update their parameter vectors iteratively by using (36) to find a joint optimal solution. However, with the idealized gradient method, the secondary users need to know the Lagrange function $\mathscr{L}(\Theta, \gamma)$ to calculate their partial differential equation for updating their parameter vector Θ_n. Furthermore, we need to calculate the gradient of the Lagrange function $\mathscr{L}(\Theta, \gamma)$ with respect to Θ_n at every time step, which is impossible to compute if the system has a large state space. Therefore, in this paper, we propose a decentralized online learning algorithm with a small communication overhead that can estimate the gradient of the Lagrange function instead of computing its exact value. Then, the secondary users can update their parameter vectors independently and parallelly as shown in Algorithm 2.

In Step 4 of Algorithm 2, to reduce the communication overhead, we just need to send a state synchronization signal when the current state of the secondary user is the recurrent state of that user, thereby reducing the communication for the whole system.[3]

Assumption 5. *The step sizes $\alpha(t)$ for updating parameter vectors and $\beta(t)$ for updating Lagrangian multipliers are deterministic, nonnegative and satisfy the following conditions:*

$$\sum_{t=1}^{\infty} \alpha(t) = \sum_{t=1}^{\infty} \beta(t) = \infty, \ \sum_{t=1}^{\infty} \big(\alpha^2(t) + \beta^2(t)\big) < \infty, \text{and } \frac{\beta(t)}{\alpha(t)} \rightarrow 0.$$

The last condition, i.e., $\frac{\beta(t)}{\alpha(t)} \rightarrow 0$, implies that the sequence $\{\beta(t)\} \rightarrow 0$ faster than the sequence $\{\alpha(t)\}$. For example, we can choose $\alpha(t) = \frac{1}{t^{2/3}}$ and $\beta(t) = \frac{1}{t}$ or $\alpha(t) = \frac{1}{t}$ and $\beta(t) = \frac{1}{1+t\log t}$, and so on [28].

[3] In the case when state $s^{\dagger}$ is visited, each secondary user will receive all messages from all other users to notify that they are in recurrent state $s_n^{\dagger}$.

Algorithm 2 Decentralized Online Learning Algorithm with Communications

1) **Initialization:** Each secondary user determines the local parameter vector Θ_n^0.

2) **Sensing and Decision Epoch:** At the beginning of each time slot, the secondary user makes a decision to sense a target channel based on the information from its local state (i.e., the number of packets and the energy level). If the sensed channel is busy and the energy storage of the secondary user is not full, the secondary user will harvest RF energy. By contrast, if the sensed channel is idle, the energy queue and the data queue are not empty, then the secondary user will transmit a packet. Otherwise, the secondary user does nothing.

3) **Channel Processing:** After decisions are made, the secondary users perform RF energy harvesting or packet transmission according to the decisions made in the sensing and decision phases.

4) **Information Sharing:** At the end of each time slot, each secondary user determines its local current state and shares the following information, $\mathcal{I}_n = g\big(s_n(t), a_n(t)\big) + \gamma_n(\mathscr{Q}_n(t) - \mathscr{Q}_n^*)$. If the current state of secondary user n is $s_n^\dagger$, the user will also send a state synchronization signal υ_n to the other secondary users.

5) **Updating parameter Θ_n:** Each secondary user updates the local parameter Θ_n as follows:

$$\Theta_n^{t+1} = \Theta_n^t - \alpha(t)\Big(\mathcal{I}_G - \widetilde{\mathcal{L}}^t\Big) z_n^t, \tag{38}$$

where $\mathcal{I}_G = \sum_{n=1}^N \mathcal{I}_n$ is the current total value of the Lagrange functions of secondary users and $\widetilde{\mathcal{L}}^t$ is the estimated total value of Lagrange functions and it is updated as follows:

$$\widetilde{\mathcal{L}}^{t+1} = \widetilde{\mathcal{L}}^t - \alpha(t)\Big(\mathcal{I}_G - \widetilde{\mathcal{L}}^t\Big), \tag{39}$$

where $\alpha(t)$ is the step size satisfying Assumption 5, and

$$z_n^{t+1} = \begin{cases} \frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a_n)}{\mu_{\Theta_n}(s_n,a_n)}, & \text{if } s^\dagger \text{ is visited,} \\ z_n^t + \frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a_n)}{\mu_{\Theta_n}(s_n,a_n)}, & \text{otherwise.} \end{cases} \tag{40}$$

6) **Updating Lagrangian multiplier γ_n:** Each secondary user updates the local Lagrangian multiplier $\{\gamma_n\}$ as follows:

$$\gamma_n^{t+1} = \max\Big(\gamma_n^t + \beta(t)\big(\mathscr{Q}_n(t) - \mathscr{Q}_n^*\big), 0\Big), \tag{41}$$

where $\beta(t)$ is the step size satisfying Assumption 5.

With Algorithm 2, the secondary users can make decisions based on their local information and exchanged messages (i.e., $\mathcal{I}_n$ and υ_n defined in Algorithm 2). Appendix B provides the analysis and the proof of the convergence for Algorithm 2.

VI. Performance Evaluation

In this section, we perform simulations to evaluate the performance of the RF energy harvesting cognitive radio network.

A. Simulation Setup

We perform simulations through using MATLAB to evaluate the performance of the network under different parameters and scenarios. First, we consider the case with two primary channels and two secondary users. The secondary users cooperate to maximize the network throughput. In this case, we will show the convergence of the learning algorithms together with their optimal policies. We then increase the number of cooperative secondary users and compare the network throughput of the proposed algorithms with two other schemes, namely, greedy policy and threshold policy. For the greedy policy, the secondary users will transmit a packet when they meet certain conditions (i.e., the energy queue and the data queue is not empty). Otherwise the secondary users will harvest energy. For the threshold policy, the secondary users will transmit data when the data queue is not empty and the energy level in the energy queue is higher than a safety level. We set the safety level for the threshold policy for secondary user n as $\lfloor \mathscr{E}_n/2 \rfloor$ where $\lfloor \cdot \rfloor$ is the floor function. With the threshold policy, secondary users can reserve a certain energy in order to serve for data transmission when the sensed channel is idle, and also avoid collisions when the number of available channels is few. For both the greedy policy and threshold policy, we assume that secondary users know the channel idle probabilities of all the channels in advance. Thus if the secondary user wants to transmit a packet, it will sense the channel that has higher idle probability, whereas when the secondary user wants to harvest energy, it will choose the channel that has lower idle probability.

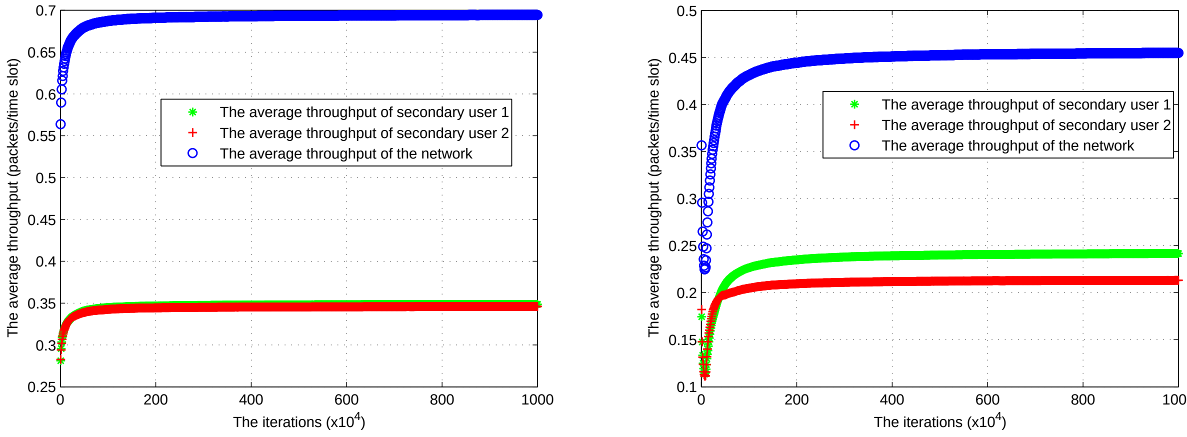
Finally, we increase the number of channels and consider two scenarios. In the case with two secondary users and two channels, we assume that the channel idle probabilities of channel 1 and channel 2 are 0.2 and 0.8, respectively. When we increase the number of channels to 3, the channel idle probability of channel 3 is 0.2 in the first scenario and 0.8 in the second scenario. The successful packet transmission and the successful energy harvesting probabilities for all cases are set to be 0.95. We set the false alarm probability and the miss detection probability to be 0.01. The maximum number of packets in the data queues of secondary users is 5 packets and the maximum levels of the energy queues are 5 units of energy. The packet arrival probability at data queues is 0.5.

For the parameter vector Θ, at the beginning of the learning algorithms, secondary users will choose to sense channels with the same probabilities. For example, if we have two channels to sense, the probabilities to sense channel 1 or channel 2 are equal, i.e., 0.5.

B. Simulation Results

In the first case, we show the convergence of the learning algorithms and the optimal policies obtained by using the proposed algorithms with two secondary users and two primary channels. Figures 2 (a) and (b) show the convergence of the TDMA learning algorithm (i.e., Algorithm 1) and the decentralized learning algorithm (i.e., Algorithm 2) respectively, through the average throughput of the network. As shown in Fig. 2, the average throughput of the Algorithm 1 converges to around 0.7 after 2×10^6 iterations, while the average throughput of the Algorithm 2 converges to approximately 0.46 after 4×10^6 iterations. For the TDMA learning algorithm, the secondary users have to experience two learning processes. In the first learning process, a secondary user needs to determine whether it should transmit a packet or harvest energy given the current state. In the second learning process, the secondary user needs to choose a channel to sense to maximize the throughput. For example, when the secondary user wants to transmit a packet, it will choose the channel with the highest channel idle probability. However, for the decentralized leaning algorithm, the secondary user has three learning processes. The secondary user has to learn not only when to transmit a packet and which channel to sense, but also the behaviors of other secondary users due to random access process. As a result, the convergence rate of the decentralized learning algorithm will be slower than that of the TDMA learning algorithm as shown in Fig. 2.

In Fig. 2, the average throughput for both secondary users obtained by the TDMA learning algorithm is nearly equal, while the average throughput of secondary users obtained by the decentralized learning algorithm has noticeable difference. The reason can be explained from the policies obtained from the learning algorithm. Fig. 3 and Fig. 4 present the policies obtained by the TDMA learning algorithm and Fig. 5 presents the policies obtained by the decentralized learning algorithm for both secondary users. In these figures, the x-axis, y-axis, and z-axis represent the number of packets, the energy levels, and the probability of sensing channel 1 of the secondary user. In Figs. 3 and 4, the policies obtained by both secondary users are almost the same, while in Fig. 5, the policies obtained by the decentralized leaning algorithm for both secondary users are different. For the TDMA learning



(a) (b)

Fig. 2. The convergence of (a) the TDMA learning algorithm and (b) the decentralized learning algorithm.

algorithm, the secondary users are cooperative in a round-robin fashion, and thus, at each time slot there is just only one secondary user interacting with the environment (i.e., primary channels). As a result, with the same environment, the secondary users will have the same policies to optimize their throughput. However, for the decentralized learning algorithm, the secondary users interact not only with the channels to explore the environment, but also with other secondary users to learn their behaviors. The policies applied for the secondary users are randomized parameterized policies, and thus, actions are selected randomly. Consequently, in the network, the policies for the secondary users can be different as long as their objective (i.e., the average throughput) is maximized. We need to note that, although the policies for the secondary users could be different, the average network throughput obtained by the decentralized learning algorithm still converges to the same point. The policies obtained by the TDMA learning algorithm can be explained as follows. The secondary users will sense channel 1 (higher channel busy probability) to harvest energy when they are not scheduled to access a time slot (i.e., not allowed to transmit a packet) and the secondary users will sense channel 2 which has higher idle probability to transmit a packet when they are scheduled (allowed to transmit a packet).

We then vary the number of secondary users to evaluate the network performance. We also compare the performance with other schemes to demonstrate the efficiency of the proposed algorithms. As shown in Fig. 6, when the number of secondary users increases from 2 to 5, the average total throughput obtained by the TDMA learning algorithm is the highest and it increases when the number of secondary users increases and becomes saturated. In contrast, the average throughput obtained by the decentralized learning algorithm, the greedy policy and the threshold policy decrease as the number of secondary users increases. Here, although the average throughput of the decentralized algorithm is greater than

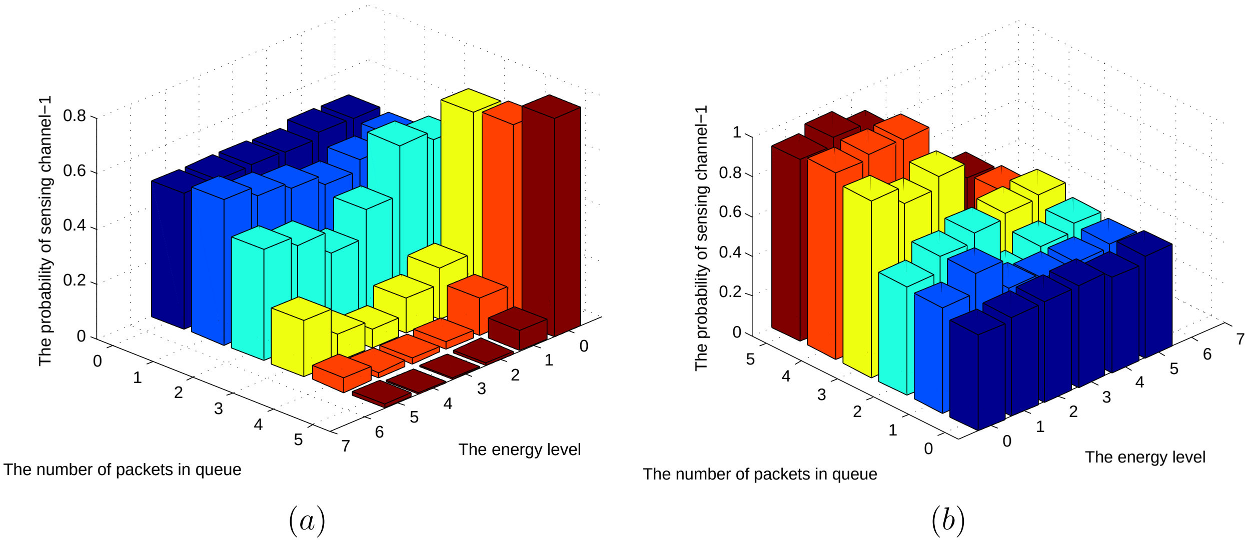


Fig. 3. The policy of secondary user 1 obtained by the TDMA learning algorithm for the cases that (a) it is not scheduled and (b) it is scheduled to access a time slot.

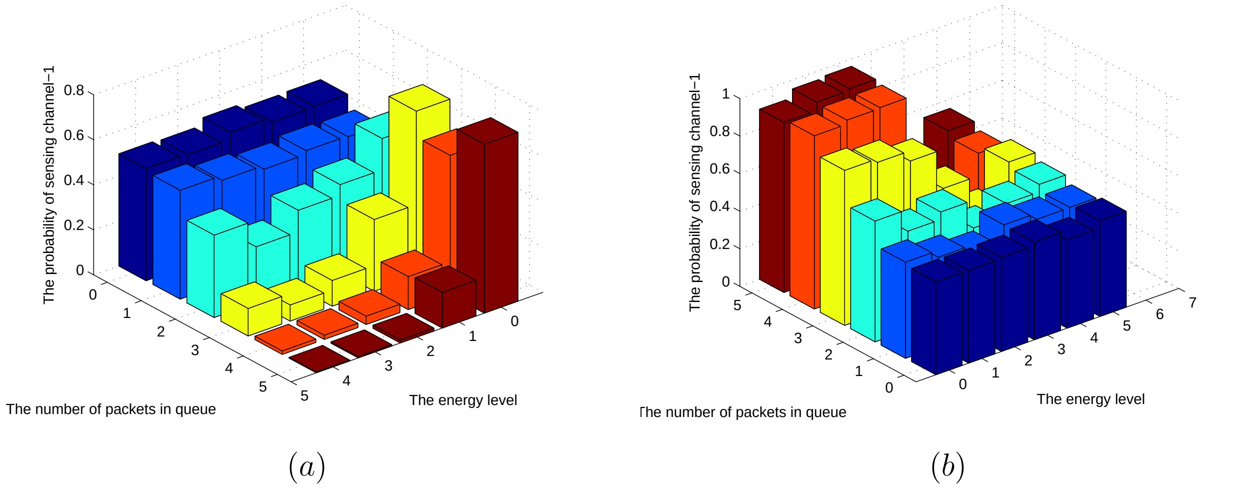


Fig. 4. The policy of secondary user 2 obtained by the TDMA learning algorithm for the cases that (a) it is not scheduled and (b) it is scheduled to access a time slot.

those of the greedy policy and the threshold policy, it is still much lower than that of the TDMA learning algorithm. The reason is due to the energy harvesting process of secondary users. In the TDMA learning algorithm, when any secondary users are not scheduled to access a time slot, they can harvest energy. This is not the case for the decentralized learning algorithm since the secondary users have to contend for transmission according to their states, losing opportunity to harvest energy. Additionally, the transmissions in the decentralized learning algorithm can result in collision, even lower the network throughput. Finally, for the greedy policy and the threshold policy, they are not optimized, and hence, their performance is inferior to those of the learning algorithms.

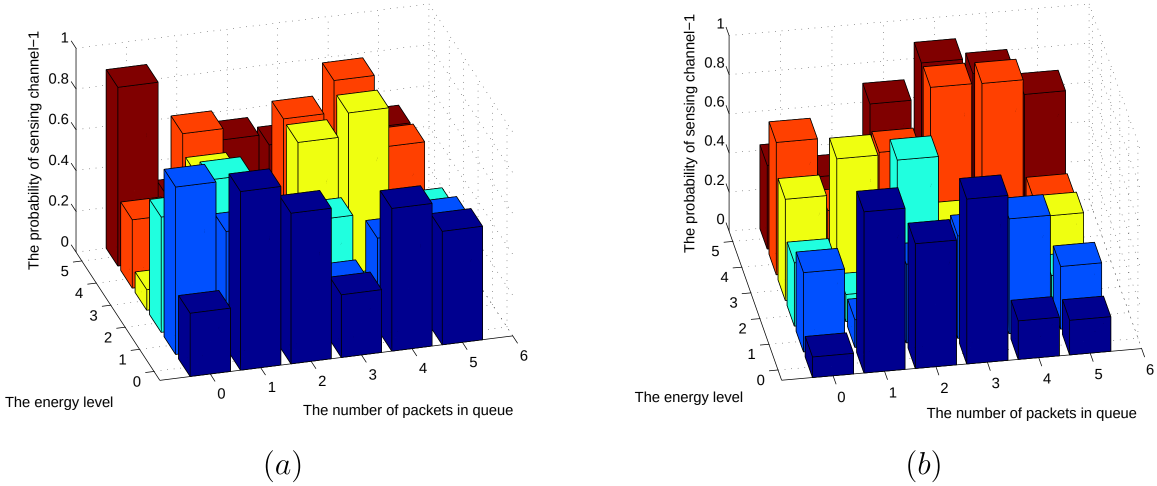


Fig. 5. The policy of secondary users (a) 1 and (b) 2 obtained by the decentralized learning algorithm.

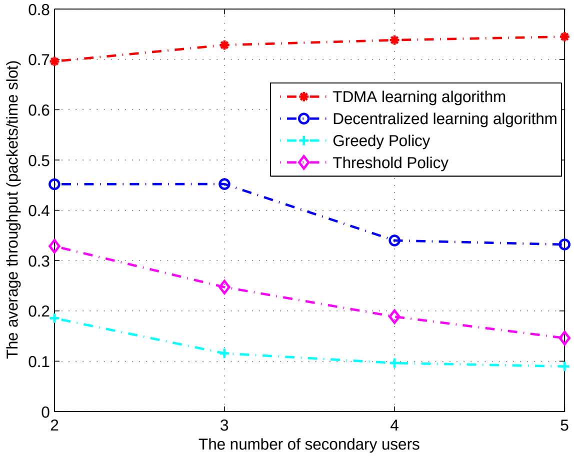
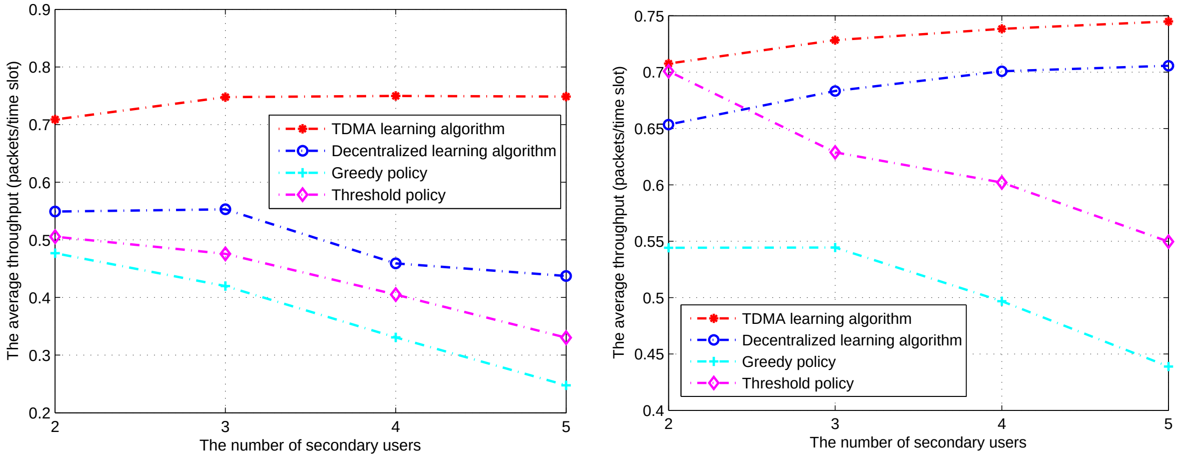


Fig. 6. The average throughput of the system under different algorithms with 2 channels.

We now investigate the case when there are three available primary channels. Figure 7 (a) shows the results for the case of two channels with high idle probability (i.e., 0.8) and one with low idle probability (i.e., 0.2). In Fig. 7 (a), the performance obtained by the algorithms is similar to the case with one channel having high idle probability and one channel having low idle probability as shown in Fig. 6. However, the average throughput obtained in the former case is greater than that in the latter case. The reason is that the secondary users now have more chances to harvest energy and transmit packets. Figure 7 (b) presents the results for the case of two channels with low idle probability and one channel with high idle probability. In Fig. 7 (b), the average throughput obtained by the algorithms increases significantly except the TDMA learning algorithm. The reason is because the TDMA learning algorithm works in a round-robin fashion. When the number of channels increases, the network performance will not be impacted. However, for other algorithms, when the number of



(a) (b)

Fig. 7. The average throughput with (a) two channels with low idle probability and one channel with high idle probability and (b) two channels with high idle probability and one channel with low idle probability.

idle channels increases, there are more chances for packets to be transmitted successfully in a time slot.

In practice, the number of users could be many, but the number of primary channels is limited. Normally, there are two or three channels, e.g., WLANs, 3G networks, or TV bands. Thus, in Fig. 6 and Fig. 7, we compare the performance of the algorithms with 2 or 3 channels. In all the cases, the performance of the TDMA learning algorithm is always better than that of other algorithms even when we continue increasing the number of users. However, in Fig. 8, when we have more channels, i.e., 2 channels with high idle probability and 2 channels with low idle probability, the performance of the TDMA learning algorithm is not always the best. Specifically, in Fig. 8, as the number of secondary users increases from 2 to 5, the average throughput of the decentralized learning algorithm increases and higher than that of the TDMA learning algorithm. However, if the number of secondary users keeps increasing, the average throughput of the decentralized learning algorithm will decrease, and it will be lower than that of the TDMA learning algorithm when the number of SUs is higher than 7. This is because when the number of SUs increases, while the number of channels is unchanged, the collision will be increased and thus throughput will be reduced for the decentralized learning algorithm. This leads to the following conclusions. In general scenarios of cognitive radio networks with RF energy harvesting technique, the TDMA learning algorithm is often a stable and efficient solution for secondary users. However, in some special cases when the number of available primary channels is large with enough idle channels and the number of users is not so many, the decentralized learning algorithm can yield better performance.

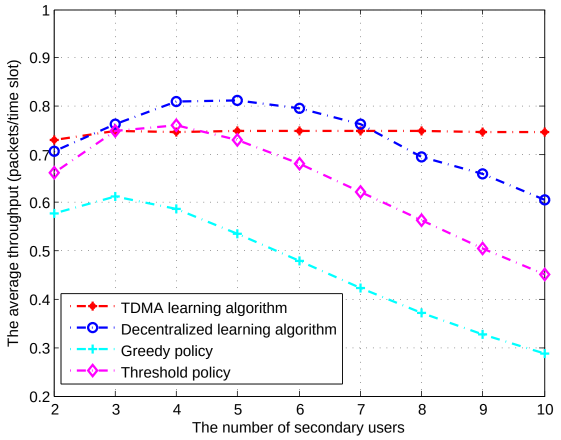


Fig. 8. The average throughput of the system under two channels with low idle probability and two channels with high idle probability.

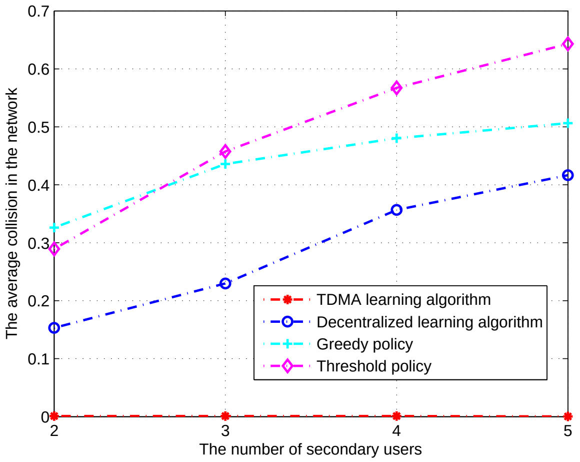


Fig. 9. Collision probability with two channels.

Additionally, we also show the average collision probability of the algorithms in Fig. 9. Note that the collision probability of the TDMA learning algorithm is very small. The collision in TDMA learning algorithm only occurs when a secondary user has a miss detection sensing error event. However, the average collision probabilities obtained by other algorithms are relatively high. This is because the collision can happen due to simultaneous transmissions by multiple secondary users.

VII. Conclusion

This paper has introduced the RF energy harvesting cognitive radio network. The secondary users in the network can harvest RF energy from a busy channel occupied by a primary user, and transmit their packets on an idle channel. The main focus is to solve the performance optimization problem for the RF energy harvesting cognitive radio network with multiple secondary users and multiple channels. We have studied two solutions, namely, a TDMA learning algorithm and decentralized learning algorithm to obtain optimal channel access policies for secondary users in an environment with incomplete information. The simulation results have clearly shown the convergence of the learning algorithms as well as their efficiency in terms of network throughput. For the future work, we will consider the case when the secondary users do not want to cooperate and they are selfishly interested in maximizing their own performance. In that case, a noncooperative game with equilibrium solution can be explored. Moreover, we may consider using realistic simulators such as NS-3, Castalia, and OMNET to verify results obtained from MATLAB in this paper. The implementation and experiment will provide more insight when the algorithms are used in the real system.

Appendix A
The proof of Proposition 2

From (32), we have

$$\mathscr{L}(\Theta,\gamma) = \sum_{s\in\mathcal{S}} \pi_s(\Theta)\mathscr{G}(s,\Theta), \tag{42}$$

where $\pi_s(\Theta)$ is the steady state probability, $\mathscr{L}(\Theta,\gamma)$ is the Lagrange function and $\mathscr{G}(s,\Theta)$ is the value of Lagrange function at state s. All are parameterized by parameter vector Θ.

Then, we take gradient for the Lagrange function $\mathscr{L}(\Theta,\gamma)$ with respect to the vector Θ as follows

$$\begin{aligned}\nabla_\Theta\mathscr{L}(\Theta,\gamma) &= \sum_{s\in\mathcal{S}} \pi_s(\Theta)\nabla_\Theta\mathscr{G}(s,\Theta) + \sum_{s\in\mathcal{S}} \nabla_\Theta\pi_s(\Theta)\mathscr{G}(s,\Theta),\\ &= \sum_{s\in\mathcal{S}} \pi_s(\Theta)\nabla_\Theta\mathscr{G}(s,\Theta) + \sum_{s\in\mathcal{S}} \nabla_\Theta\pi_s(\Theta)\mathscr{G}(s,\Theta) - \mathscr{L}(\Theta,\gamma)\sum_{s\in\mathcal{S}}\nabla_\Theta\pi_s(\Theta),\end{aligned}$$

(since $\sum_{s\in\mathcal{S}} \pi_s(\Theta) = 1$)

$$= \sum_{s\in\mathcal{S}} \pi_s(\Theta)\nabla_\Theta\mathscr{G}(s,\Theta) + \sum_{s\in\mathcal{S}} \nabla_\Theta\pi_s(\Theta)\Big(\mathscr{G}(s,\Theta) - \mathscr{L}(\Theta,\gamma)\Big), \tag{43}$$

We define the differential cost at state s by

$$d(s,\Theta) = \mathbb{E}_{\Psi(\Theta)}\Big[\sum_{t=0}^{\mathbf{T}-1}\big(\mathscr{G}(s,\Theta) - \mathscr{L}(\Theta,\gamma)\big)\big|s(0) = s\Big], \tag{44}$$

where $\mathbf{T} = \min\{t > 0|s(t) = s^\dagger\}$ is the first future time that state $s^\dagger$ is visited. The main objective of defining $d(s,\Theta)$ is to express the relation between the average value of the Lagrange function and the immediate value of the Lagrange function at state s. Furthermore, $d(s,\Theta)$ is a unique solution of the following Bellman equation,

$$d(s,\Theta) = \mathscr{G}(s,\Theta) - \mathscr{L}(\Theta,\gamma) + \sum_{s\in\mathcal{S}} \mathbf{p}(s'|s,\Psi(\Theta))d(s',\Theta). \tag{45}$$

Then, we replace $\mathscr{G}(s,\Theta)-\mathscr{L}(\Theta,\gamma)$ in (43) by $d(s,\Theta)-\sum_{s\in\mathcal{S}}\mathbf{p}(s'|s,\Psi(\Theta))d(s',\Theta)$ in (45) and have the following results:

$$\begin{aligned}
\nabla_\Theta\mathscr{L}(\Theta,\gamma) &= \sum_{s\in\mathcal{S}}\pi_s(\Theta)\nabla_\Theta\mathscr{G}(s,\Theta)+\sum_{s\in\mathcal{S}}\nabla_\Theta\pi_s(\Theta)\Big(d(s,\Theta)-\sum_{s\in\mathcal{S}}\mathbf{p}(s'|s,\Psi(\Theta))d(s',\Theta)\Big),\\
&= \sum_{s\in\mathcal{S}}\pi_s(\Theta)\nabla_\Theta\mathscr{G}(s,\Theta)+\sum_{s\in\mathcal{S}}\nabla_\Theta\pi_s(\Theta)d(s,\Theta)\\
&\quad+\sum_{s,s'\in\mathcal{S}}\bigg(\pi_s(\Theta)\nabla_\Theta\mathbf{p}(s'|s,\Psi(\Theta))-\nabla_\Theta\Big(\pi_s(\Theta)\mathbf{p}(s'|s,\Psi(\Theta))\Big)\bigg)d(s',\Theta),\\
&\text{(since } \nabla_\Theta\Big(\pi_s(\Theta)\mathbf{p}(s'|s,\Psi(\Theta))\Big)=\nabla_\Theta\pi_s(\Theta)\mathbf{p}(s'|s,\Psi(\Theta))+\pi_s(\Theta)\nabla_\Theta\mathbf{p}(s'|s,\Psi(\Theta)))\\
&= \sum_{s\in\mathcal{S}}\pi_s(\Theta)\nabla_\Theta\mathscr{G}(s,\Theta)+\sum_{s\in\mathcal{S}}\nabla_\Theta\pi_s(\Theta)d(s,\Theta)\\
&\quad+\sum_{s,s'\in\mathcal{S}}\pi_s(\Theta)\nabla_\Theta\mathbf{p}(s'|s,\Psi(\Theta))d(s',\Theta)-\sum_{s'\in\mathcal{S}}\nabla_\Theta\Big(\sum_{s\in\mathcal{S}}\pi_s(\Theta)\mathbf{p}(s'|s,\Psi(\Theta))\Big)d(s',\Theta),\\
&= \sum_{s\in\mathcal{S}}\pi_s(\Theta)\nabla_\Theta\mathscr{G}(s,\Theta)+\sum_{s\in\mathcal{S}}\nabla_\Theta\pi_s(\Theta)d(s,\Theta)\\
&\quad+\sum_{s,s'\in\mathcal{S}}\pi_s(\Theta)\nabla_\Theta\mathbf{p}(s'|s,\Psi(\Theta))d(s',\Theta)-\sum_{s'\in\mathcal{S}}\nabla_\Theta\pi_{s'}(\Theta)d(s',\Theta)\\
&\text{(since } \pi_{s'}(\Theta)=\sum_{s\in\mathcal{S}}\pi_s(\Theta)\mathbf{p}(s'|s,\Psi(\Theta))),\\
&= \sum_{s\in\mathcal{S}}\pi_s(\Theta)\Big(\nabla_\Theta\mathscr{G}(s,\Theta)+\sum_{s'\in\mathcal{S}}\nabla_\Theta\mathbf{p}(s'|s,\Psi(\Theta))d(s',\Theta)\Big).
\end{aligned}\tag{46}$$

From (30), we have $\mathscr{G}(s,\Theta)=\sum_{a\in\mathcal{A}}\mu_\Theta(s,a)\mathscr{G}(s,a)$ and we take gradient for function $\mathscr{G}(s,\Theta)$ under the parameter vector Θ_n as follows:

$$\begin{aligned}
\nabla_{\Theta_n}\mathscr{G}(s,\Theta) &= \sum_{a\in\mathcal{A}}\nabla_{\Theta_n}\mu_\Theta(s,a)\mathscr{G}(s,a),\\
&= \sum_{a\in\mathcal{A}}\mu_\Theta(s,a)\frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a)}{\mu_{\Theta_n}(s_n,a)}\mathscr{G}(s,a)\quad\text{since } \mu_\Theta(s,a)=\Pi_n\mu_{\Theta_n}(s_n,a),\\
&= \sum_{a\in\mathcal{A}}\mu_\Theta(s,a)\frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a)}{\mu_{\Theta_n}(s_n,a)}\Big(\mathscr{G}(s,a)-\mathscr{L}(\Theta,\gamma)\Big)\quad\text{since } \sum_{a\in\mathcal{A}}\mu_\Theta(s,a)=1.
\end{aligned}\tag{47}$$

From (24), we have $\mathbf{p}(s'|s,\Psi(\Theta))=\sum_{a\in\mathcal{A}}\mu_\Theta(s,a)\mathbf{p}(s'|s,a)$, and we also take gradient for function $\mathbf{p}(s'|s,\Psi(\Theta))$ under the parameter vector Θ_n as follows:

$$\nabla_{\Theta_n}\mathbf{p}(s'|s,\Psi(\Theta))=\sum_{a\in\mathcal{A}}\mu_\Theta(s,a)\frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a)}{\mu_{\Theta_n}(s_n,a)}\mathbf{p}(s'|s,a).\tag{48}$$

Then, we replace (47) and (48), into (46), and we derive the following results:

$$\begin{aligned}\nabla_{\Theta_n}\mathscr{L}(\Theta,\gamma) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\pi_s(\Theta)\mu_\Theta(s,a)\frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a_n)}{\mu_{\Theta_n}(s_n,a_n)}\Big(\mathscr{G}(s,a)-\mathscr{L}(\Theta,\gamma)+\sum_{s'\in\mathcal{S}}\mathbf{p}(s'|s,a)d(s',\Theta)\Big),\\ &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\pi_s(\Theta)\mu_\Theta(s,a)\frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_n,a_n)}{\mu_{\Theta_n}(s_n,a_n)}q(s,a,\Theta),\end{aligned} \tag{49}$$

where

$$\begin{aligned} q(s,a,\Theta) &= \mathscr{G}(s,a)-\mathscr{L}(\Theta,\gamma)+\sum_{s'\in\mathcal{S}}\mathbf{p}(s'|s,a)d(s',\Theta),\\ &= \mathbb{E}_{\Psi(\Theta)}\Big[\sum_{t=0}^{\mathbf{T}-1}\Big(\mathscr{G}\big(s(t),a(t)\big)-\mathscr{L}(\Theta,\gamma)\Big)\Big|s(0)=s,a(0)=a\Big].\end{aligned} \tag{50}$$

The proof now is completed.

Appendix B
The convergence proof of Algorithm 2

We analyze and then derive the proof for the convergence of the Algorithm 2 presented in Section V. In Algorithm 2, the parameter vectors and the Lagrange multipliers are updated at two different time scales, namely, $\alpha(t)$ and $\beta(t)$, respectively. At the time scale $\beta(t)$, the value of the Lagrange multipliers are updated by a function of $\beta(t)$ as follows:

$$\gamma_n^{t+1} = \gamma_n^t + \mathcal{F}(\beta(t)) \quad \forall n. \tag{51}$$

Furthermore, under the Assumption 5, time scale $\beta(t)$ can be represented as a function of $\alpha(t)$ as follows:

$$\gamma_n^{t+1} = \gamma_n^t + \mathcal{F}(\beta(t)) = \gamma_n^t + f(\alpha(t)) \quad \forall n. \tag{52}$$

Consequently, we can conclude that (38) views (41) as quasi-static (i.e., almost a constant), while (41) views (38) as almost equilibrated (as shown in Chapter 6 [28]). Therefore, we have to establish the proof of convergence for Algorithm 2 at different time scales separately.

We will first prove the convergence of the algorithm in the first time scale (i.e., $\alpha(t)$). In other words, we will prove that $\nabla_\Theta\mathscr{L}\big(\Theta^\infty(\gamma),\gamma\big)=0$.

Proof. We define vector $\mathbf{r}^t = \begin{bmatrix}\Theta^t & \widetilde{\mathcal{L}}^t\end{bmatrix}^\top$. Then, the update equations as given in (38) and (39) of the Algorithm 2 can be rewritten in the following form,

$$\mathbf{r}^{t+1} = \mathbf{r}^t + \alpha(t)\mathbf{H}^t, \tag{53}$$

where

$$\mathbf{H}^t = \begin{bmatrix} -\Big(\mathcal{I}_G - \widetilde{\mathcal{L}}^t\Big) z_n^t \\ -\Big(\mathcal{I}_G - \widetilde{\mathcal{L}}^t\Big) \end{bmatrix}. \tag{54}$$

Let t_m be the time when the recurrent state $s^\dagger$ is revisited at m-th time and thus (53) can be presented as follows:

$$\mathbf{r}^{t_{m+1}} = \mathbf{r}^{t_m} + \sum_{t=t_m}^{t_{m+1}-1} \alpha(t)\mathbf{H}^t. \tag{55}$$

If we denote $\alpha(m) = \sum_{t=t_m}^{t_{m+1}-1} \alpha(t)$, then we have $\epsilon_m = \sum_{t=t_m}^{t_{m+1}-1} \alpha(t)\big(\mathbf{H}^t - \mathbf{h}(\mathbf{r}^{t_m})\big)$ where

$$\mathbf{h}(\mathbf{r}^{t_m}) = \begin{bmatrix} -\mathbb{E}_\Theta[T]\nabla_\Theta \mathscr{L}(\Theta) - \mathscr{V}(\Theta)\big(\mathscr{L}(\Theta) - \widetilde{\mathcal{L}}(\Theta)\big) \\ -\mathbb{E}_\Theta[T]\big(\mathscr{L}(\Theta) - \widetilde{\mathcal{L}}(\Theta)\big) \end{bmatrix}, \tag{56}$$

for $\mathscr{V}(\Theta) = [\mathscr{V}_1(\Theta), \ldots, \mathscr{V}_N(\Theta)]$ and

$$\mathscr{V}_n(\Theta) = \mathbb{E}_\Theta\left[\sum_{t'=t_m+1}^{t_{m+1}-1} \big(t_{m+1} - t'\big)\frac{\nabla_{\Theta_n}\mu_{\Theta_n}(s_{t'}, a_{t'})}{\mu_{\Theta_n}(s_{t'}, a_{t'})}\right].$$

Then, (55) becomes

$$\mathbf{r}^{t_{m+1}} = \mathbf{r}^{t_m} + \alpha(m)\mathbf{h}(\mathbf{r}^{t_m}) + \epsilon_m. \tag{57}$$

Next, we will show that

$$\sum_{m=1}^{\infty} \alpha(m) = \infty, \text{ and } \sum_{m=1}^{\infty} \alpha^2(m) < \infty \quad \text{with probability one.} \tag{58}$$

From Assumption 5, we have $\sum_{m=1}^{\infty} \alpha(m) = \sum_{t=1}^{\infty} \alpha(t) = \infty$. Additionally, under Assumption 5, the sequence $\alpha(t)$ is non-increasing, and thus, $\alpha(m) = \sum_{t_m}^{t_{m+1}-1} \alpha(t) \leq \alpha(t_m)(t_{m+1} - t_m)$. Then, we derive $\sum_{m=1}^{\infty} \alpha^2(m) \leq \sum_{m=1}^{\infty} \alpha^2(t_m)(t_{m+1} - t_m)^2 < \sum_{t=1}^{\infty} \alpha^2(t) < \infty$.

Now, we can transform the update equations into the same form as presented in (25) in [20]. By using Lemma 2 and Lemma 3 in [20], we have the sequence $\mathbf{h}(\mathbf{r}^{t_m})$ bounded and the sequence $\sum_{m=1}^{\infty} \epsilon_m$ converges almost surely. As a result, we can conclude that

$$\lim_{m\to\infty} (\mathbf{r}^{t_{m+1}} - \mathbf{r}^{t_m}) = 0 \quad \text{with probability one.} \tag{59}$$

After that, based on Lemma 11 and Section C of Appendix 1 in [20], we can prove that the sequence $\mathscr{L}(\Theta^{t_m})$ and $\widetilde{\mathcal{L}}(\Theta^{t_m})$ converge to a common limit, and thus, $\nabla_\Theta \mathscr{L}(\Theta^{t_m})$ converges to zero, i.e., $\nabla_\Theta \mathscr{L}(\Theta^\infty) = 0$. The proof for the convergence in the first time scale now is completed. □

Then, we will prove the convergence of the algorithm in the second time scale (i.e., $\beta(t)$). It means, we will prove that $\lim_{t\to\infty} \gamma^t = \gamma^\infty$ with probability 1 and γ^∞ satisfies the constraints in (27).

Proof. From Assumption 5, since the time scale $\beta(t)$ is very small compared with the time scale $\alpha(t)$, this makes γ^t quasi-static compared with Θ^t and this also has an effect similar to fixing γ^t and running (41) to infinity. In turn, γ^t views Θ^t as a converged approximation to $\Theta^*(\gamma^t)$ [29], and thus, we can rewrite the equation in (41) as follows:

$$\gamma_n^{t+1} = \max\bigg(\gamma_n^t + \beta(t)\Big(\overline{\mathscr{Q}}_n(\Theta^*(\gamma)) - \mathscr{Q}_n^* + \mathscr{Q}_n(t) - \overline{\mathscr{Q}}_n(\Theta^*(\gamma))\Big), 0\bigg). \tag{60}$$

We denote $w_n^{t+1} = \mathscr{Q}_n(t) - \overline{\mathscr{Q}}_n(\Theta^*(\gamma))$ and let $\mathscr{F} \triangleq \sigma(\gamma^l, w_n^l, l \leq t)$ be the σ-algebra generated by $\{\gamma_n^l, w_n^l, l \leq t\}$. We can see that the sequence $\{w_n^t\}$ is a martingale difference sequence since its expectation with regard to the past is zero, i.e., $\mathbb{E}[w_k^{t+1}|\mathscr{F}^t] = 0$. Moreover, we can always find an appropriate constant K such that $\mathbb{E}[||w_n^{t+1}||^2|\mathscr{F}^t] \leq K(1 + ||\gamma_n^t||^2)$. Hence, by using the standard stochastic approximation in [28], the Lagrange update equation for secondary users n can be represented by the ordinary differential equation (ODE) as follows:

$$\dot{\gamma}_n^t = \overline{\mathscr{Q}}_n(\Theta^*(\gamma^t)) - \mathscr{Q}_n^*. \tag{61}$$

We define

$$\mathscr{I}(\gamma) = \sum_{n=1}^{N} \Big(\overline{\mathcal{C}}_n(\Theta^*(\gamma)) + \gamma_n(\overline{\mathscr{Q}}_n(\Theta^*(\gamma)) - \mathscr{Q}_n^*)\Big). \tag{62}$$

Then, based on the chain rule, we have

$$\frac{\partial \mathscr{I}(\gamma)}{\partial \gamma_n} = \overline{\mathscr{Q}}_n(\Theta^*(\gamma)) - \mathscr{Q}_n^*, \tag{63}$$

and thus

$$\dot{\gamma}_n^t = \frac{\partial \mathscr{I}(\gamma)}{\partial \gamma_n}. \tag{64}$$

Consequently, from Proposition 3 in [30], we have $\lim_{t\to\infty} \frac{\partial \mathscr{I}(\gamma)}{\partial \gamma_n} = 0$. In other words, $\overline{\mathscr{Q}}_n(\Theta^*(\gamma^\infty)) - \mathscr{Q}_n^* = 0$ which satisfies the constraints in (27). □

Now, the proof of the convergence of Algorithm 2 is completed.

REFERENCES

[1] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, “Energy-aware wireless microsensor networks,” *IEEE Signal Processing Magazine*, vol. 19, issue 2, pp. 40-50, 2002.

[2] E. Hossain, D. Niyato, and Z. Han, “Dynamic Spectrum Access and Management in Cognitive Radio Networks,” Cambridge University Press, 2009.

[3] D. T. Hoang, D. Niyato, P. Wang, D. I. Kim, “Opportunistic channel access and RF energy harvesting in cognitive radio networks,” *IEEE Journal of Selected Areas in Communications*, vol. 32, no. 11, November 2014.

[4] D. T. Hoang, D. Niyato, “Performance analysis for cognitive machine-to-machine communications”, in *IEEE International Conference on Communication Systems (ICCS)*, pp. 245-249, Singapore, November 2012.

[5] Mischa Schwartz, “Mobile wireless communications,” Cambridge University Press, 2005.

[6] D. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, “The complexity of decentralized control of Markov decision processes”, *Journal of Mathematics of Operations Research*, vol. 27, pp. 819-840, November 2002.

[7] A. Sultan, “Sensing and transmit energy optimization for an energy harvesting cognitive radio,” *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 500-503, October 2012.

[8] 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 (WCSP)*, pp. 1-5, October, 2013.

[9] 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, March 2013.

[10] Q. Zhang, B. Cao, Y. Wang, N. Zhang, X. Lin, and L. Sun, “On exploiting polarization for energy-harvesting enabled cooperative cognitive radio networking,” *IEEE Wireless Communications*, vol. 20, no. 4, pp. 116-124, August 2013.

[11] P. Nintanavongsa, U. Muncuk, D. Richard Lewis, and K. R. Chowdhury, “Design optimization and implementation for RF energy harvesting circuits,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 1, March, 2012.

[12] H. Ostaffe, “Power out of thin air: Ambient RF energy harvesting for wireless sensors,” 2010 [Online] powercastco.com/PDF/Power-Out-of-Thin-Air.pdf.

[13] A. M. Zungeru, L. M. Ang, S. Prabaharan, and K. P. Seng, “Radio frequency energy harvesting and management for wireless sensor networks,” in *Green Mobile Devices and Networks Energy Optimization and Scavenging Techniques*, pp. 341-368, 2012.

[14] N. Barroca, J. M. Ferro, L. M. Borges, J. Tavares, and F. J. Velez, “Electromagnetic energy harvesting for wireless body area networks with cognitive radio capabilities,” in *Proceedings of URSI Seminar of the Portuguese Communications*, Lisbon, Portugal, November 2012.

[15] S. Lee, R. Zhang, and K. Huang, “Opportunistic wireless energy harvesting in cognitive radio networks,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4788-4799, September 2013.

[16] S. Park, J. Heo, B. Kim, and W. Chung, “Optimal mode selection for cognitive radio sensor networks with RF energy harvesting,” in *Proceedings of IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, September 2012.

[17] 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, December 2013.

[18] D. Niyato, P. Wang, and D. I. Kim, “Channel selection in cognitive radio networks with opportunistic RF energy harvesting,” to be presented in IEEE ICC, Sydney, Australia, 10-14 June 2014.

[19] C. Mikeka, H. Arai, “Design issues in radio frequency energy harvesting system,” *Sustainable Energy Harvesting Technologies - Past, Present, and Future*, December 2011.

[20] P. Marbach, and J. N. Tsitsiklis, “Simulation-based optimization of Markov reward processes,” in *IEEE Transactions on Automatic Control*, vol. 46, pp. 191-209, Feb. 2001.

[21] O. Buffet, A. Dutech, and F. Charpillet, “Shaping multi-agent systems with gradient reinforcement learning,” *Journal of Autonomous Agents and Multi-Agent System*, vol. 15, pp. 197-220, Jan. 2007.

[22] J. Baxter, P. L. Barlett, L. Weaver, “Experiments with infinite-horizon, policy-gradient estimation,” *Journal of Artificial Intelligence Research*, vol. 15, pp. 351-381, Nov. 2001.

[23] Ahmed El Shafie, Ahmed Sultan, “Optimal random access and random spectrum sensing for an energy harvesting cognitive radio,” in *IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications*, pp.403-410, Oct. 2012.

[24] Gozde Ozcan and M. Cenk Gursoy, “Cognitive radio transmissions exploiting multi-user diversity under channel and sensing uncertainty,” *IEEE Communications Letters*, vol. 17, issue 9, pp.1714-1717, August 2013.

[25] Dimitri P. Bertsekas, “Nonlinear Programming,” Athena Scientific, Belmont, MA, 1995.

[26] Abhijit Gosavi, “Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning,” Springer Press, 2003.

[27] H. W. Kuhn, and A. W. Tucker, “Nonlinear Programming,” in *Proceedings of 2nd Berkeley Symposium on Mathematical Statics and Probability*, pp. 481-492, 1951.

[28] V. S. Borkar, “Stochastic Approximation: A Dynamic Viewpoint,” Cambridge University Press, 2008.

[29] V. R. Konda, and J. N. Tsitsiklis, “Convergence rate of linear two-time-scale stochastic approximation,” *The Annals of Applied Probability*, vol. 14, no. 2, pp. 796-819, 2004.

[30] D. P. Bertsekas, and J. N. Tsitsiklis, “Gradient convergence in gradient methods with errors,” *Society for Industry and Applied Mathematics Journal on Optimization*, vol. 10, no. 3, pp. 627-642, 1999.