A Distributed Learning Automata Scheme for Spectrum Management in Self-Organized Cognitive Radio Network

Mina Fahimi and Abdorasoul Ghasemi

Abstract—We propose a distributed Learning Automata (LA) for spectrum management problem in Cognitive Radio (CR) networks. The objective is to design intelligent Secondary Users (SUs) which can interact with the RF environment and learn from its different responses through the sensing. It is assumed there is no prior information about the Primary Users (PUs) and other SUs activities while there is no information exchange among SUs. Each SU is empowered with an LA which operates in the RF environment with different responses. That is, the SUs are considered as agents in a self-organized system which select one channel as an action and receive different responses from the environment based on how much their selected actions are favorable or unfavorable. Using these responses, SUs control their accesses to the channels for appropriate spectrum management with the objective to incur less communication delay, less interference with PUs, and less interference with other SUs. The proposed LA-based distributed algorithm is investigated in terms of asymptotic convergence and stability. Simulation results are provided to show the performance of the proposed scheme in terms of SUs' waiting times, interference with other SUs, the number of interruptions by PUs during their transmissions, and fairness.

Index Terms—Cognitive radio networks, multi-agent, learning automata, self-organization, non-stationary environment

1 Introduction

In order to promote the spectrum utilization, Cognitive Radio (CR) technology has been proposed for a more flexible and intelligent wireless communication [1]. In a class of CR networks the Secondary Users (SUs) explore the spectrum idle spaces and opportunistically exploit them tacking into account the interference with licensed or Primary Users (PUs) [2]. Therefore to be aware of RF environment, each SU should behave intelligently by spectrum sensing and appropriately interacting with this environment [3].

The cognitive cycle [1] provides SUs' reliable communications using four functionalities including spectrum sensing, spectrum management, spectrum mobility and handoff management, and spectrum sharing. First using the sensing, the CR agents explore the spectrum to find possible opportunities as an initial step for the environment perception. Then, these sensing observations are used by SUs, as intelligent agents, for spectrum management to choose the best available channel for possible exploitation. Upon a PU's packet arrival on this channel, the corresponding SU must vacate this channel and perform spectrum handoff in order to switch to another available channel or wait until the channel becomes idle again. Finally, efficient spectrum sharing mechanisms are required to avoid SUs' contention for available opportunities. Therefore, efficient design of the cognitive cycle involves with appropriate interaction with the environment to learn both the PUs' activities and the behavior of other SUs [4]. The former is required to avoid collisions with PUs and is attained by efficient spectrum sensing and handoff management. The latter should be used for distributed coordinating of SUs to share the opportunities efficiently and fairly and can be attained with or without message exchanging.

Due to the dynamic nature of the network environment, using artificial intelligence techniques in designing context aware CR networks are received more attention in recent researches. For example, there are several studies on using machine learning, game theory, artificial neural networks, and reinforcement learning in designing efficient protocols for these networks [5], [6], [7]. Specifically, a CR network can be modeled as a distributed self-organized multi-agent system in which each SU or agent perform spectrum management by efficient interacting with the environment through a learning policy [5]. In this approach the effects of other SUs' decisions can be considered as a part of the responses of the environment for each SU. Learning algorithms are appropriate techniques in designing and analyzing self-organized networks [8], [9].

The objective of this paper is on performing spectrum management by SUs as agents in a self-organized manner to exploit available spectrum opportunities on primary channels efficiently and fairly only with a learning module and without any information exchange with other SUs. The Learning Automata (LA) based module is used to make the appropriate strategy in spectrum decisions, spectrum resource allocation, and interference management autonomously and independently by each SU using environmental feedbacks or responses [10]. That is the PUs' packet arrival rates and service times as well as the number of other SUs and their arrival rates or channel selection strategies are unknown for each SU.

- *The authors are with the Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran.
E-mail: mfahimi@mail.kntu.ac.ir, arghasemi@kntu.ac.ir.*

*Manuscript received 30 Sept. 2015; revised 21 June 2016; accepted 11 Aug. 2016. Date of publication 24 Aug. 2016; date of current version 3 May 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2016.2601926*

From each SU point of view, this environment is a non-stationary environment [10], [11] since the SUs changing their decisions over the time while new SUs may join and current SUs may leave the network. The SU's learning module is designed to improve its channel selection policy by updating the current strategy over time from the received different rewards or penalties from the non-stationary environment. Selecting busy or idle channel, contending with other SUs in exploiting an idle channel, and possible required handoff during a packet transmission are the responses of the environment which are used during the on-line learning process. The system level objective is to ensure that the decision making by SUs converges to an equilibrium point in which the system stability is guaranteed and the QoS of SUs are satisfied.

The rest of this paper is organized as follows: In Section 2, some related work about using learning approach in CR networks are reviewed. System model and problem statement are presented in Section 3. In Section 4, we review the LA and Q-model environment and its properties. Then, in Section 5 CR network is modeled as a multi-agent system and the responses of the environment and learning policy for this environment are discussed. Some analytical results and properties of the proposed spectrum management scheme are investigated in Section 6. Simulation results and discussion are provided in Section 7 before concluding the paper in Section 8.

2 Related Work

Distributed spectrum management schemes can be classified according to how the SUs receive or infer the required information about the environment and how this information is used in making decisions or updating the channel selection strategies. Traditionally the required information for each SU about other SUs' strategies can be received by message passing among SUs by an in band information exchange or using a common control channel [12], [13], [14], [16], [17], [18]. This information, on the other hand, can be learned by observing the local experience of each SU in interacting with the environment [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. Incorporating the received information in the former scheme for decision making by each SU is simpler but incurs overhead to the network for information exchange and synchronization. On the other hand, the latter scheme does not incur overhead, however, typically designing and analyzing efficient self-organized mechanisms are more difficult.

In [12], [13] the spectrum access problem is modeled as a graphical congestion game with the aim to minimize the interference and congestion among competitive neighboring SUs. In [14], a graphical game model is proposed for opportunistic spectrum access (OSA) and two uncoupled learning algorithms are proposed. The first one needs a common control channel between SUs to achieve the optimal solution whereas the second one achieves suboptimal solution without using this channel. In [15], by designing a game theoretic model for dynamic spectrum access problem, a learning algorithm based on regret tracking procedure is introduced. In [16], the authors proposed spatial adaptive play (SAP) and concurrent-SAP which needs global coordination of the players, to achieve a near-optimal solution. In [17], a dynamic strategy learning (DSL) algorithm is proposed for channel selection problem. By introducing a priority virtual queue interface in order to exchange information among multiple SUs, each SU reaches a better response solution over the time. In [18], the distributed OSA is considered as a multi-armed bandit problem and a distributed matching-learning algorithm is proposed to achieve high throughput with an upper-bound for the system regret provided that the required information exchange between SUs is guaranteed.

In [19], [20] stochastic learning automata (SLA) is used in a game theoretic solution without information exchange among SUs for OSA. In SLA, the agents use mixed strategies to select actions according to a L_{RI} automaton to reach an equilibrium point in a distributed environment [21]. Ref. [19], [20] consider only the interference among SUs as the response of the environment without attending to the priority of PUs and their possible interruptions.

In [22] a reinforcement multi-agent Q-Learning is proposed for a simple network with two SUs without information exchange. In [23] a distributed Q-Learning is proposed in order to decrease the interference between SUs assuming complete and incomplete information of the environment. The optimal policy is learned by agents in the complete information case which needs to train a neural network to find the appropriate values for reinforcement learning. In the incomplete information case, a suboptimal policy is made by training a complex neural network which makes the decision making complicated. A distributed reinforcement learning algorithm with less information and complexity is proposed in [24] to trade-off between exploration and exploitation in which by controlling the exploration phase, SUs learn a preferred set of channels for exploitation. In [25] different schemes are proposed for efficient exploration phase and finding the preferred set of channels by using reinforcement learning. These studies are focused on exploration and finding the preferred channels which reduce the sensing cost in CR networks without considering the priority of PUs and competition among SUs. In [26] a distributed learning mechanism is proposed based on Multi Response Learning Automata (MRLA) in which the effect of PUs on SUs' packet transmission as well as the interference between SUs are considered as different responses of the environment. Each SU learns to make decisions such that incur less interference and achieve high throughput. Assuming a simple traffic model for PUs, the effects of PUs' priority and their interruptions do not take into account in [26]. In [27], considering OSA as a multi-armed bandit problem, two distributed learning schemes are proposed with minimal total regret to maximize the system throughput. The first scheme needs some prior information of the system. In the second one, this information could be estimated by SUs cooperation provided that the PUs' traffic pattern do not change over the time. A fully distributed CR network with no information exchange among SUs is formulated as a multi-armed bandit problem with an optimal system regret in [28] in which none or only one of SUs receives response from the environment when interference occurs.

In this paper, a learning automaton based scheme is designed for channel selection by SUs in a distributed and self-organized manner by receiving the responses of the

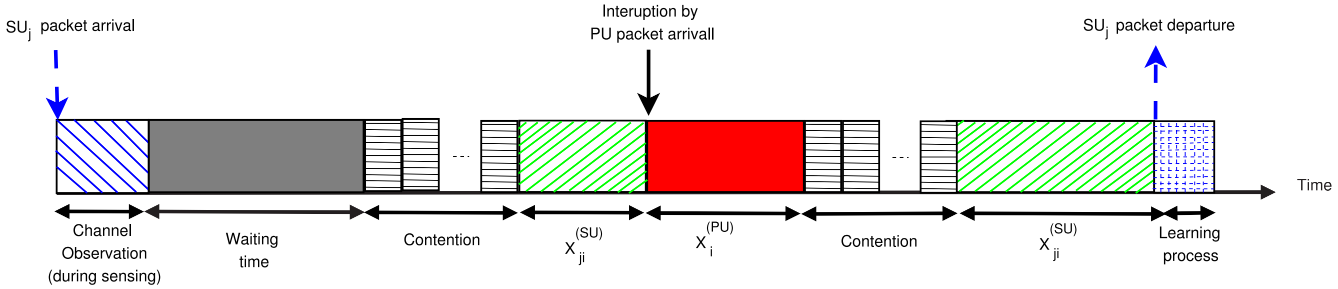

Fig. 1. SU_j's packet transmission on channel F_i and different possible raising events.

environment and without information exchange among SUs. It takes into account the priority of PUs and their interruptions as well as the collision between SUs in spectrum sharing problem. It is supposed that SUs do not have prior information about the RF environment. That is, the PUs' packet arrival rates and their service times as well as the number of other SUs and their arrival rates or channel selection strategies are unknown from each SU point of view. The environment is non-stationary in which SUs or PUs can leave or join the network when other SUs are taking service.

3 System Model and Problem Statement

We consider a CR network with M heterogeneous time slotted channels which are used by high-priority PUs. The set of channels and the corresponding set of dedicated PUs are denoted by $\mathcal{F} = \{F_1, F_2, \ldots, F_M\}$ and $\mathcal{PU} = \{PU_1, PU_2, \ldots, PU_M\}$, respectively. The average service rate of channel F_i is denoted by μ_i (arrivals/slot) where PU_i's packets arrive according to a Poisson process with average $\lambda_i^{(PU)}$ (arrivals/slot) and wait in a high-priority queue to get service. Also, there are N low-priority SUs which opportunistically exploit the available spectrum and their set is denoted by $\mathcal{SU} = \{SU_1, SU_2, \ldots, SU_N\}$. For SU_j there is an action set which is denoted by $\mathbf{a}_j = \{a_{j1}, a_{j2}, \ldots, a_{jM}\}$ where a_{ji} is the action of selecting channel F_i by SU_j.

When SU_j has a packet arrival event in time slot t, it chooses action $a_j(t) = a_{ji}, i = 1, \ldots, M$ with probability p_{ji}. For efficient spectrum management, SU_j should adjust its action selection probability profile $\mathbf{p}_j = [p_{j1}, p_{j2}, \ldots, p_{jM}]$ taking into account the available service rates of the channels which may be exploited by other SUs. Therefore, the two dimensional matrix $\mathbf{P} = \left[\mathbf{p}_1^T, \mathbf{p}_2^T, \ldots, \mathbf{p}_N^T\right]^T$ represents the SUs' action selection probabilities in which its rows and columns corresponds to the SUs and channels, respectively. Let $\mathbf{p}_{-j}$ denotes the action selection probabilities of all SUs except SU_j. The packet arrival process of SU_j is a Poisson process with average $\lambda_j^{(SU)}$ (arrivals/slot). Hence the offered packets of SU_j for exploiting channel F_i is also a Poisson process with average $\lambda_{ji}^{(SU)} = p_{ji}\lambda_j^{(SU)}$ (arrivals/slot).

The packets service times of PU_i and SU_j on channel F_i are modeled by random variables $X_i^{(PU)}$ and $X_{ji}^{(SU)}$ respectively, which follow the exponential distribution with average $E[X_i^{(PU)}] = E[X_{ji}^{(SU)}] = \frac{1}{\mu_i}$. Also, it is assumed that the total load of PUs and SUs do not overload this channel, i.e., we have:

$$\lambda_i^{(PU)} + \sum_{j=1}^{N} \lambda_{ji}^{(SU)} < \mu_i. \tag{1}$$

At the beginning of each time slot, SU_j selects and perfectly senses a channel and will exploit this channel if it is idle [29]. If the selected channel is busy, the intended packet should wait in a low-priority queue which is assigned by SU_j to this channel. SU_j continues sensing this channel and contends with other SUs for exploitation using the CSMA method. If the packet transmission of SU_j is interrupted by the PU's packet arrival, the interrupted packet waits until the current channel becomes idle again. Fig. 1 shows the time between an arrival of a SU_j's packet until it departures.

The experienced possible contention with other SUs and interruptions by the PU is used in the learning procedure of SU_j at the end of each packet transmission. The objective of the learning procedure is adjusting the action selection probabilities such that SU_j encounters less interference with other SUs, less handoff occurrence during its transmission, and less communication delay. Note that the RF environment is completely heterogonous in which the PUs' service rates and their arrival rates are different. Therefore, SUs experience different feedbacks on different channels. Also, current SUs may leave or new SUs may join to the system which makes the environment more dynamic and change the responses of the environment non-stationary. Learning to select appropriate actions in this environment is a challenging issue in multi-agent systems.

4 Background on LA and Q-Model Environment

Learning Automata is an abstract model which selects an appropriate action out of a finite set of available actions during the interaction with a random environment. The random environment evaluates the selected input action and responses to the LA as an output, based on how much the selected input action is favorable or unfavorable. Then, the LA updates its action selection probabilities based on received response. A variable-structure LA [10], is defined mathematically by the quadruple $LA = \{\mathbf{a}, \beta, \mathbf{p}, T\}$ and a random environment is define by triple $E = \{\mathbf{a}, \beta, \mathbf{c}\}$. In these definitions, $\mathbf{a} = \{a_1, a_2, \ldots, a_r\}$ represents the finite action set for the LA and the input set for the environment.

In a simple P-Model environment, $\beta = \{0, 1\}$ represents the binary input set for the LA and the output set for the environment where $\beta = 0$ and $\beta = 1$ shows the favorable and unfavorable response respectively. In this environment, $\mathbf{c} = \{c_1, c_2, \ldots, c_r\}$ represents the penalty probabilities in which c_i corresponds to the penalty probability to action a_i and is defined by $c_i = Pr[\beta = 1|a = a_i]$. For variable-structure LA, $\mathbf{p} = \{p_1, p_2, \dots, p_r\}$ and $p(k+1) = T[a(k), \beta(k), p(k)]$ represent the action probability set and the learning algorithm respectively.

The environment can be modeled by Q-Model with more outputs in a finite discrete number of responses to each action. Let $\beta_{\mathbf{i}} = \{\beta_i^1, \beta_i^2, \dots, \beta_i^{m_i}\}$ denotes the finite set of the responses of the environment for action a_i, $i = 1, \dots, r$ where m_i is the number of these responses. Dedicated to action a_i and the corresponding environment output β_i^l there is a penalty probability which is defined by $c_i^l = Pr[\beta_i = \beta_i^l|a = a_i]$, $i = 1, \dots, r$, $l = 1, \dots, m_i$. Therefore, the expected penalty for action a_i is defined by [10]:

$$\begin{aligned} C_i =& E[\beta_i|a = a_i] \\ =& \beta_i^1 Pr\big[\beta_i = \beta_i^1|a = a_i\big] + \cdots \\ &+ \beta_i^{m_i} Pr\big[\beta_i = \beta_i^{m_i}|a = a_i\big] \\ =& \sum_{l=1}^{m_i} \beta_i^l c_i^l. \end{aligned} \tag{2}$$

The LA updates the selection probability of action a_i taking into account the environment response as given by (3) which is known as SL_{RP} scheme where α is the learning parameter [10]. It is worth to note that when an agent selects action $a_{m\neq i}$, the probability of action a_i is updated based on the response of the environment to action $a_{m\neq i}$ which is $\beta_{m\neq i}$ as we can see in the first part of (3). If $\beta_{m\neq i}$ is high, the probability of selecting action a_i will increase and vice versa

$$p_i(t+1) = \begin{cases} p_i(t) + \beta_{m\neq i}^l(t)\Big[\frac{\alpha}{r-1} - \alpha p_i(t)\Big] \\ \quad -\Big[1 - \beta_{m\neq i}^l(t)\Big]\alpha p_i(t), \quad if\ a(t) \neq a_i. \\ p_i(t+1) = p_i(t) - \beta_i^l(t)\alpha p_i(t) \\ \quad +\Big[1 - \beta_i^l(t)\Big]\alpha(1 - p_i(t)), \quad if\ a(t) = a_i. \end{cases} \tag{3}$$

In many practical cases, the environment may change during the learning process where the penalty probabilities are varying. In these non-stationary environments, the actions of the LA affect the penalty probabilities of the environment. The value of expected penalty, C_i, is depended on the value of action selection probabilities for non-stationery environment.

According to how the penalty values are changed with the actions of the LA, three mathematical models are introduced in [10] which are referred to as Model A, Model B, and Model C. In Model A, the penalties changes are constant however in Model B penalty values are functions of action selection probabilities which are monotonically increasing functions respect to action selection probabilities. In Model C, penalty values are dependent on action selection probabilities and the penalty values of the previous step.

Adopting Model B, we consider a set of penalties for each action of SUs which their values are monotonically increasing functions respect to action selection probabilities. In this model, the action which is selected more often will have higher penalty probability [30]. This feature enables the agents to control their actions in the multi-agent environment specially in the case of resource allocation and sharing among multiple agents where none of agents can perform one action unilaterally to use the system resources.

5 CR Network as a Multi-Agent System

Consider each SU as an intelligent agent empowered with an LA in Q-Model environment. SU_j has no prior information about the other agents, SU_{-j}, and the environment except the channels service rates. The PUs' arrival rates on different channels, other SUs' arrival rates, and their action selection probabilities are unknown during the learning process. SU_j receives different responses depending on the different events in the environment and also the actions of SU_{-j}, and makes the decisions about channel selection probabilities only by observing the environment and its responses.

5.1 The Possible Events in the Environment

Each SU is equipped by a scheduler which makes the decision about channel selection probabilities and updates these probabilities according to the system feedback from the transmitted packet. A simple flow of different situations that an SU may encounter during a packet transmission is shown in Fig. 2. In this figure, all possible scenarios after sensing a given channel are shown.

Assume that based on current action probability profile, SU_j decides to send a new arriving packet on channel F_i. SU_j sends its packet immediately if channel F_i is idle. However, if it finds this channel busy, the packet should wait in the low-priority queue of SU_j. During the waiting time of this packet, other SUs may also have packet arrivals which are decided to send on channel F_i. When channel F_i becomes idle, the waiting SUs will collide and then, they will contend with each other to exploit this channel. SU_j's transmission is therefore deferred until it wins the contention. If SU_j wins, it can transmit its packet if no interruption by high-priority PUs is sensed. However, its transmission on channel F_i may be interrupted if there is a PU's packet arrival during the service time of SU_j's packet. In this case, SU_j should perform handoff procedure and the interrupted packet must return to the low-priority queue and will be retransmitted when the channel becomes idle. Finally, when SU_j's packet transmission on channel F_i is finished, it performs the learning procedure based on the different perceptions on channel F_i. In the following, different events in the environment which can be effective in the learning process are discussed. Based on these events and their sequence, the different responses of the environment are introduced which can help the SUs to perform better spectrum decisions in upcoming time slots. From SU_j point of view sensing channel busy or idle, collision with other SUs, and interruption by the PUs' arrivals can be considered as the most important events during its packet transmission and the respective responses of the environment can help SU_j to learn an appropriate spectrum decision. Therefore, in order to follow up the effect of these events in the learning process and the punishment of the environment respect to these events, the probability of these events are computed in the following. These probabilities help us to analyze the proposed scheme and follow up the behavior of the SUs during the learning process.

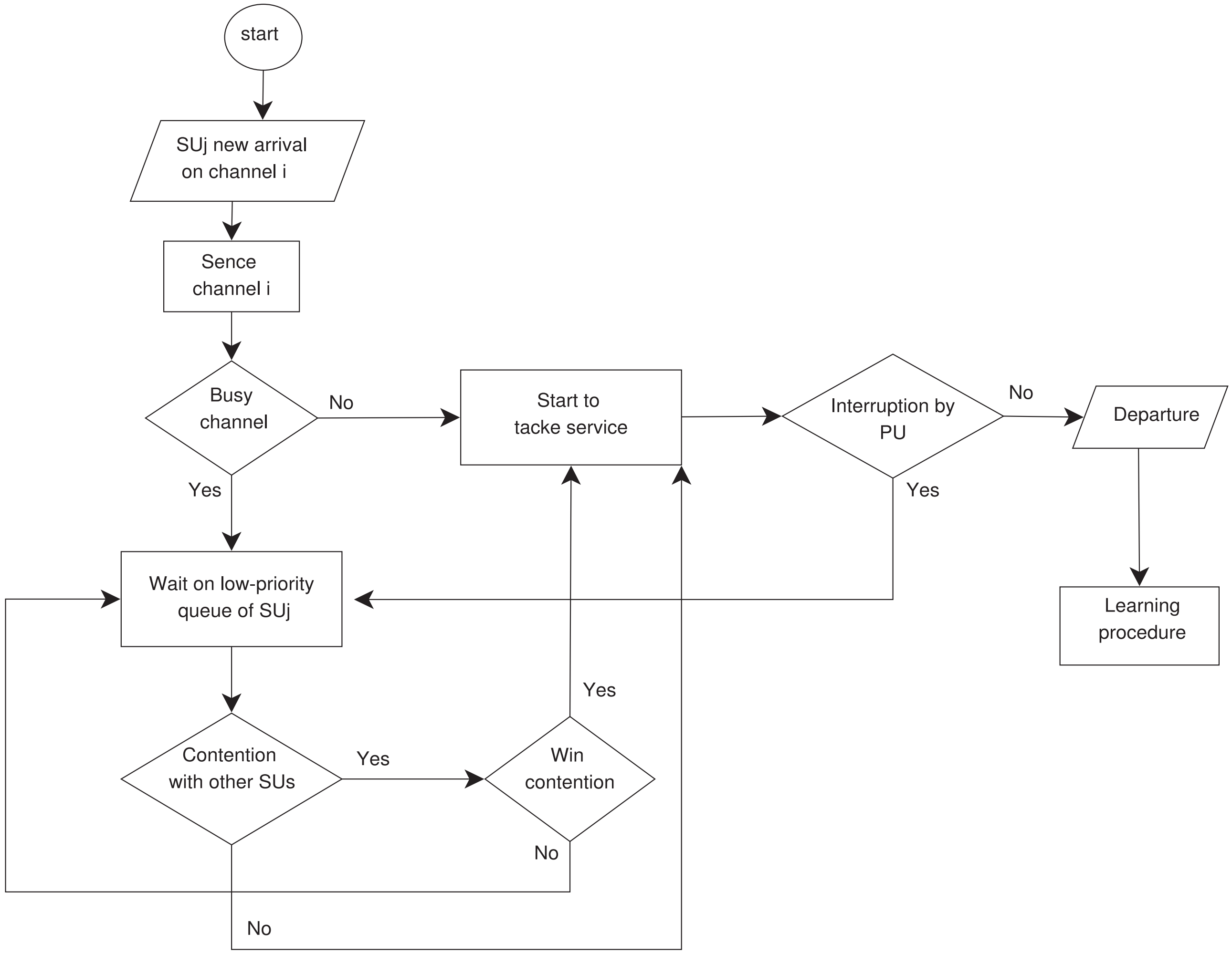

Fig. 2. Flow of different situations that SUs may encounter.

At the beginning of a packet transmission, SU_j senses channel F_i. Since SU_j's packets arrival is a Poisson process, using the *PASTA, i.e., Poisson Arrivals See Time Average,* property each packet sees the environment, channel F_i, in its average status.

Let $P_i^{(busy)}$ denote the probability that SU_j senses channel F_i busy. This probability can be computed by the average proportion of time that this channel is busy and is given by (4)

$$P_i^{(busy)} = \frac{\lambda_i^{(PU)} + \sum_{j=1}^{N} p_{ji}\lambda_j^{(SU)}}{\mu_i}. \qquad (4)$$

SU_j senses channel F_i idle with probability $1 - P_i^{(busy)}$ and in this case, it can send its packet immediately without any collision. However, if it senses this channel busy, it must wait and during this waiting time, if other SUs have packet arrival on channel F_i, there will be a collision between these SUs when this channel becomes idle. This can be inferred that the collision event is dependent on the busy status of channel F_i and the probability of the collision event is under the condition of the busy probability of channel F_i. In addition, in order to compute the probability of the collision between SUs, denoted by $P_{(k\neq j)i}^{(arrival)}$, it is needed to compute the probability of at least one arrival of SU_{-j} during waiting time of SU_j when channel F_i is busy. The expected waiting time of SU_j's packet on channel F_i is denoted by $E[W_{ji}]$. Since the probability of zero arrival for one of $SU_k, k = 1, \ldots, N, k \neq j$ on channel F_i during the $E[W_{ji}]$ is $\exp(-E[W_{ji}]p_{ki}\lambda_k^{(SU)})$, the probability that SU_{-j} have at least one packet arrival during the $E[W_{ji}]$ is

$$P_{(k\neq j)i}^{(arrival)} = 1 - \exp\left(- E[W_{ji}] \sum_{\substack{k=1 \\ k \neq j}}^{N} p_{ki}\lambda_k^{(SU)} \right). \qquad (5)$$

In order to compute $E[W_{ji}]$ using PRP M/G/1 queueing model for high-priority PUs and low-priority SUs we have (6) and (7) [31]

$$E[W_{ji}] = \frac{E[R_i]}{\left(1 - E[X_i^{(PU)}]\lambda_i^{(PU)}\right)\left(1 - E[X_i^{(PU)}]\lambda_i^{(PU)} - \sum_{j=1}^{N} p_{ji}\lambda_j^{(SU)} E[X_{ji}^{(SU)}]\right)}, \qquad (6)$$

where $E[R_i]$ is the average remaining service time on channel F_i

$$E[R_i] = \frac{1}{2}\left(E[(X_i^{(PU)})^2]\lambda_i^{(PU)} + \sum_{j=1}^{N} (E[(X_{ji}^{(SU)})^2] p_{ji}\lambda_j^{(SU)}) \right). \qquad (7)$$

Since $X_{ji}^{(SU)}$ and $X_i^{(PU)}$ follow the exponential distribution with average $\frac{1}{\mu_i}$, $E[W_{ji}]$ is given by

$$E[W_{ji}] = \frac{p_{ji}\lambda_j^{(SU)} + R_{ji}}{\left(\mu_i - \lambda_i^{(PU)}\right)\left(\mu_{ji} - p_{ji}\lambda_j^{(SU)}\right)}, \tag{8}$$

where R_{ji} and μ_{ji} are

$$R_{ji} = \lambda_i^{(PU)} + \sum_{\substack{k=1 \\ k \neq j}}^{N} p_{ki}\lambda_k^{(SU)} \tag{9}$$

$$\mu_{ji} = \mu_i - R_{ji}. \tag{10}$$

By substituting (8) in (5) we can compute the probability of at least one arrival of SU_{-j} on channel F_i during the waiting time of SU_j when channel F_i is busy. We assume $P_{ji}^{(collision)}$ as the collision probability of SU_j on channel F_i and it can be computed as $P_{ji}^{(collision)} = \Pr(arrival_{(k\neq j)i}|F_i^{(busy)})P_i^{(busy)} + \Pr(arrival_{(k\neq j)i}|F_i^{(idle)})P_i^{(idle)}$ in which $\Pr(arrival_{(k\neq j)i}|F_i^{(busy)})$ means that at least one arrival of SU_{-j} when SU_j senses channel F_i busy and $\Pr(arrival_{(k\neq j)i}|F_i^{(idle)})$ means that at least one arrival of SU_{-j} when SU_j senses channel F_i idle. As we mentioned before, from SU_j point of view, when this SU has a packet arrival and senses channel F_i idle, it means there is not any arrivals of other SUs on that moment and therefore $\Pr(arrival_{(k\neq j)i}|F_i^{(idle)}) = 0$. Therefore, the probability of the collision between SU_j and SU_{-j} can be considered as $P_{ji}^{(collision)} = P_{(k\neq j)i}^{(arrival)}P_i^{(busy)}$.

After the collision, when SU_j wins the contention, it starts to take service on channel F_i. Since handoff event also affects the learning process of SU_j, we need to compute the probability of handoff occurrence on channel F_i which is denoted by $P_i^{(handoff)}$ during the packet transmission of this user. This probability is given by [32]

$$P_i^{(handoff)} = \frac{\lambda_i^{(PU)}}{\lambda_i^{(PU)} + \mu_i}. \tag{11}$$

In addition, the handoff event is not dependent on the collision event between SUs as well as the busy or idle status of the channel. Whether or not SU_j senses the corresponding channel busy, or collides with other SUs or not, it may confront a PU's packet arrival during its service time. Therefore, this event is not under the condition of other discussed events. In the next section, we introduced responses of the environment corresponding to these events.

5.2 Evaluating Environment Repossess

Assume SU_j selects action a_{ji}, i.e., to transmit its packet on channel F_i. According to different possible events which may happen during its packet transmission the following responses of the environment are possible. Note that SU_j makes decision about its actions only by using these responses without any information exchange with other SUs or any prior information about PUs or the actions of SU_{-j}. Also, the probability of each responses of the environment can be computed using $c_{ji}^l = Pr[\beta_{ji} = \beta_{ji}^l | a_j = a_{ji}]$, $l = 1, \ldots, 6$ according to value of $P_i^{(busy)}$, $P_{(k\neq j)i}^{(arrival)}$, and $P_i^{(handoff)}$.

- The arriving packet of SU_j finds the channel busy, it also contends with other SUs before exploiting the channel, and during its exploitation handoff occurs. The response of the environment is the greatest possible penalty with value β_{ji}^1 where the corresponding penalty probability is given by (12)

$$\begin{aligned} c_{ji}^1 &= P_i^{(handoff)}\, P_{ji}^{(collision)} \\ &= P_i^{(handoff)}\, P_i^{(busy)}\, P_{(k\neq j)i}^{(arrival)}. \end{aligned} \tag{12}$$

- The arriving packet of SU_j finds the channel busy, but it exploits the channel without contending with other SUs, and during its exploitation handoff occurs. The response of the environment is β_{ji}^2 with penalty probability in (13)

$$\begin{aligned} c_{ji}^2 &= P_i^{(handoff)}\, P_{ji}^{(Nocollision)} \\ &= P_i^{(handoff)}\, P_i^{(busy)}\, (1 - P_{(k\neq j)i}^{(arrival)}). \end{aligned} \tag{13}$$

- The arriving packet of SU_j finds the channel idle but during its exploitation handoff occurs. The response of the environment is β_{ji}^3 with penalty probability in (14)

$$c_{ji}^3 = P_i^{(handoff)}\, (1 - P_i^{(busy)}). \tag{14}$$

- The arriving packet of SU_j finds the channel busy, it also contends with other SUs before exploiting the channel without any handoff. The response of the environment is β_{ji}^4 with penalty probability in (15)

$$\begin{aligned} c_{ji}^4 &= (1 - P_i^{(handoff)})\, P_{ji}^{(collision)} \\ &= (1 - P_i^{(handoff)})\, P_i^{(busy)}\, P_{(k\neq j)i}^{(arrival)}. \end{aligned} \tag{15}$$

- The arriving packet of SU_j finds the channel busy, but it exploits the channel without contending with other SUs, and also without any handoff. The response of the environment is β_{ji}^5 with penalty probability in (16)

$$\begin{aligned} c_{ji}^5 &= (1 - P_i^{(handoff)})\, P_{ji}^{(Nocollision)} \\ &= (1 - P_i^{(handoff)})\, P_i^{(busy)}\, (1 - P_{(k\neq j)i}^{(arrival)}). \end{aligned} \tag{16}$$

- The arriving packet of SU_j finds the channel idle, without any handoff. The response of the environment is β_{ji}^6 with penalty probability in (17)

$$c_{ji}^6 = (1 - P_i^{(handoff)})\, (1 - P_i^{(busy)}). \tag{17}$$

Where in (12), (13), (14), (15), (16), and (17) $1 = \beta_{ji}^1 > \beta_{ji}^2 > \beta_{ji}^3 > \beta_{ji}^4 > \beta_{ji}^5 > \beta_{ji}^6 = 0$ are some selected con-stants.

The larger value of β_{ji}^l shows that the action of SU_j in selecting channel F_i is more unfavorable [10] and SU_j

precepts more unfavorable events during its transmission. We rewrite (3) for SU_j, based on its action selection probabilities and β^l_{ji} in (18) where $m_i = 6$ shows the number of the possible responses of the environment when SU_j selects channel F_i. SU_j updates its strategy profile only by using its previous strategy profile and received β^l_{ji} from the environment without any probability computations. We use (12), (13), (14), (15), (16), and (17) in order to analytically follow up the updating strategy and the behavior of SUs during the learning process.

$$p_{ji}(t+1) = \begin{cases} p_{ji}(t) + \beta^l_{jm\neq i}(t)\Big[\frac{\alpha}{M-1} - \alpha p_{ji}(t)\Big] - \Big[1 - \beta^l_{jm\neq i}(t)\Big]\alpha p_{ji}(t) \\ \quad if\ a_j(t) \neq a_{ji}\ and\ l = 1, 2, \ldots, m_i. \\ p_{ji}(t+1) = p_{ji}(t) - \beta^l_{ji}(t)\alpha p_{ji}(t) + \Big[1 - \beta^l_{ji}(t)\Big]\alpha(1 - p_{ji}(t)) \\ \quad if\ a_j(t) = a_{ji}\ and\ l = 1, 2, \ldots, m_i. \end{cases} \tag{18}$$

Using (12), (13), (14), (15), (16), and (17), the expected penalty for SU_j on selecting action a_{ji} which is denoted by C_{ji} can be computed by (19).

$$\begin{aligned} C_{ji}(\mathbf{P}) &= \sum_{l=1}^{6} \beta^l_{ji} c^l_{ji} \\ &= \frac{\lambda_i^{(PU)}}{\lambda_i^{(PU)} + \mu_i} \frac{p_{ji}\lambda_j^{(SU)} + \lambda_i^{(PU)} + \sum_{k\neq j}^N p_{ki}\lambda_k^{(SU)}}{\mu_i} \Bigg(\beta^1_{ji} - \beta^3_{ji} + (\beta^2_{ji} - \beta^1_{ji}) \\ &\quad \exp\Bigg(- \frac{p_{ji}\lambda_j^{(SU)} + R_{ji}}{(\mu_i - \lambda_i^{(PU)})(\mu_{ji} - p_{ji}\lambda_j^{(SU)})} \sum_{k\neq j}^{N} p_{ki}\lambda_k^{(SU)} \Bigg)\Bigg) \\ &\quad + \frac{\mu_i^{(PU)}}{\lambda_i^{(PU)} + \mu_i} \frac{p_{ji}\lambda_j^{(SU)} + \lambda_i^{(PU)} + \sum_{k\neq j}^N p_{ki}\lambda_k^{(SU)}}{\mu_i} \\ &\quad \Bigg(\beta^4_{ji} + (\beta^5_{ji} - \beta^4_{ji})\exp(- \frac{p_{ji}\lambda_j^{(SU)} + R_{ji}}{(\mu_i - \lambda_i^{(PU)})(\mu_{ji} - p_{ji}\lambda_j^{(SU)})} \sum_{k\neq j}^{N} p_{ki}\lambda_k^{(SU)})\Bigg) \\ &\quad + \beta^4_{ji}\ \frac{\lambda_i^{(PU)}}{\lambda_i^{(PU)} + \mu_i} \end{aligned} \tag{19}$$

It is worth to note that we do not consider the effect of multiple handoffs or multiple contentions with other SUs explicitly in our analysis separately because their probabilities and hence, their effects in the learning process are negligible. The effects of these events in the learning process are considered once at the end of the packet transmission. Specifically, in a scenario that an SU during its packet transmission, confronts with multiple collisions and contentions with the colliding SUs, the corresponding effect is considered once in the learning process at the end of the packet transmission. Also, if its packet transmission confronts multiple interruptions by PU, we consider only one interruption effect in learning process. Considering multiple penalties for multiple interruptions or multiple contentions per packet will increase the computational complexity of the proposed scheme without any significant results in practice due to their negligible probabilities.

In the next section, we show that the defined responses of the environment and therefore the computed expected penalty in (19) make the proposed algorithm analytically tractable. The pseudo code of the proposed multi-agent LA for multiple SUs is presented in Algorithm 1.

Algorithm 1. Event Based LA Scheme for SUs

```
eventList.createSortList()
loop
  if (there is an arrival event) then
    eventList.enqueue(arrivalEvent)
    E = eventList.dequeue()
  if (E is an arrival event) then
    call processArrival(E)
  if (E is a collision event) then
    call processCollision(E)
  if (E is a handoff event) then
    call processHandoff(E)
  if (E is a departure event) then
    call processDeparture(E)
  if (there is success departure) then
    eventList.enqueue(departureEvent)
end loop
processArrival(E)
    if (E is $SU_j$'s arrival)
          Sense $F_i$ where $F_i$=the selected channel
          based on $\mathbf{p}_j(t)$
          if ($F_i$ is busy) then
            wait and sense $F_i$ until it becomes idle
          if (there is a Collision with other SUs) then
            eventList.enqueue(collisionEvent)
      if (E is PU's arrival during $SU_j$'s service time)
        eventList.enqueue(handoffEvent)
processCollision(E)
    perform CSMA mechanism
processHandoff(E)
    append interrupted packet to $SU_j$'s queue
    eventList.enqueue(arrivalEvent)
    wait and sense $F_i$ until it becomes idle
processDeparture(E)
    choose appropriate value of $\beta^l_{ji}$ based on
    environment different events
    update $\mathbf{p}_j$ based on learning scheme (18)
```

6 Analysis of the Proposed Scheme

As we computed in (19), $C_{ji}(\mathbf{P})$ is a function of $\mathbf{p}_j$ and $\mathbf{p}_{-j}$ which shows the non-stationary property of the environment. We show the self-organization property for spectrum decision of SUs which behave based on Algorithm 1 in the following.

6.1 Self-Organization Property

We first discuss how the selected actions by the SUs affect the responses of the environment and vice versa.

Proposition 1. *The expected penalty $C_{ji}(\mathbf{P})$ in (19) meets the properties of the non-stationary environment Model B and also is monotonically increasing function respect to $\mathbf{p}_{-j}$.*

Proof. Please see Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2016.2601926. □

Proposition 1 implies that if SU_j selects channel F_i more often, it receives more penalty that causes SU_j controls its access to this channel in a self-organizing manner which avoids selfish exploitation of this channel by SU_j. This behavior leads to fairness among multiple SUs without any information exchange. Also, if other SUs, SU_{-j}, exploit channel F_i more often, the environment response makes this channel unfavorable selection for SU_j. Therefore, only by interaction with the environment, the SUs perform an appropriate spectrum management which leads to less interference with PUs, less contention between themselves, and also provides fairness among them.

6.2 Convergence Behavior

The asymptotic behavior of the proposed scheme in terms of convergence and stability is discussed in this section. Let define

$$\triangle \mathbf{P}(t) = E\big[\mathbf{P}(t+1)|\mathbf{P}(t)\big] - \mathbf{P}(t), \tag{20}$$

in which the value of $p_{ji}(t+1)$ follows (18).

Proposition 2. *The value of components of $\triangle \mathbf{P}(t)$ for SU_j on selecting channel F_i is given by (21)*

$$\triangle p_{ji}(t) = \alpha\left(\frac{1}{M-1}\sum_{m\neq i}^{M} p_{jm}(t)C_{jm}(\mathbf{P}(t)) - p_{ji}(t)C_{ji}(\mathbf{P}(t))\right). \tag{21}$$

Proof. Please see Appendix B, available in the online supplemental material. □

As we can see in the components of $\triangle\mathbf{P}(t)$ in (21), $\triangle\mathbf{P}(t)$ is a function of $\mathbf{P}(t)$. Therefore, we can rewrite (20) as follow:

$$\triangle \mathbf{P}(t) = \alpha \mathbf{f}(\mathbf{P}(t)), \tag{22}$$

where $f_{ji}(\mathbf{P}(t)) = \frac{1}{M-1}\sum_{m\neq i}^{M} p_{jm}(t)C_{jm}(\mathbf{P}(t)) - p_{ji}(t)C_{ji}(\mathbf{P}(t))$ and in consequence rewrite (18) in (23)

$$\mathbf{P}(t+1) = \alpha \mathbf{f}(\mathbf{P}(t)) + \mathbf{P}(t). \tag{23}$$

Therefore, the updating rule of action selection probabilities for SU_j follows the components of Equation (23). Now, we show that the proposed distributed mechanism in which the SUs follow Algorithm 1, converges to an equilibrium point that is unique and asymptotically stable.

Proposition 3. *Irrespective of $\mathbf{P}(0)$ and for sufficiently small values of α, Algorithm 1 converges to equilibrium point $\mathbf{P}^*$ which is the solution of the system of Equation in (24)*

$$\begin{cases} p^*_{1i}C_{1i}(\mathbf{P}^*) = p^*_{1m}C_{1m}(\mathbf{P}^*) \quad i,m = 1,\ldots,M \\ \sum_{i=1}^{M} p_{1i} = 1 \\ p^*_{2i}C_{2i}(\mathbf{P}^*) = p^*_{2m}C_{2m}(\mathbf{P}^*) \quad i,m = 1,\ldots,M \\ \sum_{i=1}^{M} p_{2i} = 1 \\ \vdots \\ p^*_{Ni}C_{Ni}(\mathbf{P}^*) = p^*_{Nm}C_{Nm}(\mathbf{P}^*) \quad i,m = 1,\ldots,M \\ \sum_{i=1}^{M} p_{Ni} = 1. \end{cases} \tag{24}$$

Proof. Please see Appendix C, available in the online supplemental material. □

This proposition shows that Algorithm 1 has at least one equilibrium point and there is at least one solution for the system of Equation (24). In order to investigate the stability property of the equilibrium point of proposed scheme, we use the Lyapunov theorem for discrete-time autonomous systems.

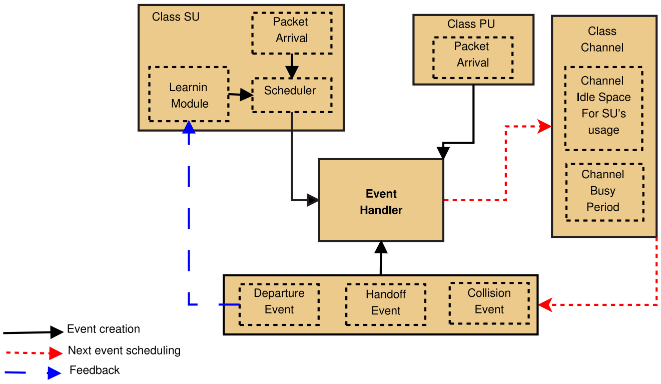

Fig. 3. Simplified simulator structure.

Proposition 4. *The equilibrium point $\mathbf{P}^*$, when Algorithm 1 converges is asymptotically Lyapunov stable.*

Proof. Please see Appendix D, available in the online supplemental material. □

Proposition 5. *The equilibrium point $\mathbf{P}^*$ which is computed by system of equations (24) is unique over the domain of $\mathbf{P}$.*

Proof. Please see Appendix D, available in the online supplemental material. □

It is worth to note that the tradeoff between convergence speed and accuracy can be made by selecting an appropriate value for learning parameter α. That is by selecting smaller values for α the algorithm will converge slowly and accurately.

7 Simulation Environment and Results

In order to simulate the distributed CR network, we use the simulator which we introduced in [33]. This simulator has been developed using C++ language in an extensible modular structure which has flexibility in choosing different parameters and adding additional modules. We extend this simulator by adding learning module to SUs' class. When a packet is departed, the corresponding SU learns from the observation of this packet and changes its channel selection probabilities for next packets. In Fig. 3, a simple block diagram of this simulator including the learning module is shown.

The network has three licensed channels with the service rates $\mu_1 = 0.2$, $\mu_2 = 0.15$, $\mu = 0.25$. The average PUs' packets arrival rates on the corresponding channels are $\lambda_1^{(PU)} = 0.04$, $\lambda_2^{(PU)} = 0.05$, $\lambda_1^{(PU)} = 0.01$. Also, six SUs with average packets arrival rates $\lambda_1^{(SU)} = 0.02$, $\lambda_2^{(SU)} = 0.03$, $\lambda_3^{(SU)} = 0.04$, $\lambda_4^{(SU)} = 0.05$, $\lambda_5^{(SU)} = 0.06$, $\lambda_6^{(SU)} = 0.07$, are considered in the system. The selected penalty responses of the environment are $\beta_{ji}^1 = 1$, $\beta_{ji}^2 = 0.8$, $\beta_{ji}^3 = 0.6$, $\beta_{ji}^4 = 0.4$, $\beta_{ji}^5 = 0.2$, $\beta_{ji}^6 = 0$ and the learning parameter is $\alpha = 0.001$.

First of all, we discuss the self-organized properties of the proposed distributed scheme. We then discuss the performance metrics in terms of the average waiting time, average number of transmission deferring by other SUs, and average interruptions by PUs in comparison with a

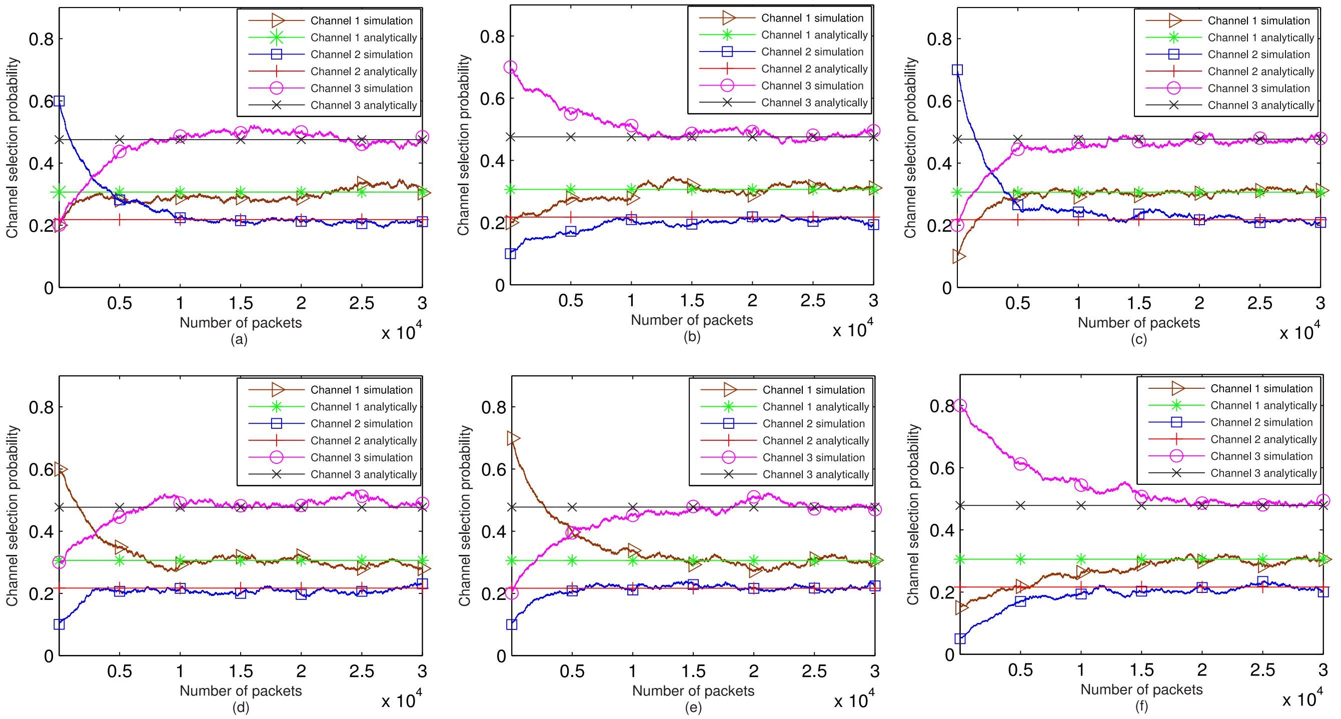

Fig. 4. Variation of the strategy profiles until convergence for a) SU_1, b) SU_2, c) SU_3, d) SU_4, e) SU_5, and f) SU_6.

centralized solution which minimizes the expected penalty for the whole system.

7.1 Self-Organized Property in Dynamic Environment

Solving (24) numerically, the final strategy profiles for six SUs are approximately equal and given by $\mathbf{p}_j = (0.30, 0.21, 0.47), j = 1, \ldots, 6$. In the simulation, the strategy profile for each SU is initialized by a random feasible strategy which selects each channel according to the unit uniform distribution.

Generating packets for each SU and processing the system events the variations in the strategy profile of each SU until convergence is shown in Fig. 4. We find that the final strategy profile of each SU is consistent with the corresponding computed strategy profile. Also, since we use SL_{RP} scheme in the learning process, starting from different initial strategies all SU approximately reach the same profile which guarantees the fairness among SU. Also, channel 3 which has the maximum service rate and the minimum PU utilization, is selected by SUs with a higher probability.

In order to investigate the adaptivity of the proposed multi-agent learning, in the next scenario SU_4, SU_5, and SU_6 leave the system. According to (24), the expected final strategy profiles of the remaining three SUs are approximately equal and given by $\mathbf{p}_j = (0.25, 0.17, 0.57), j = 1, \ldots, 3$. In this case, SU_1, SU_2, and SU_3 receive less penalty from the environment and hence update their strategies profile. In Fig. 5, the variations in the strategy profiles of these SUs are depicted until convergence which is consistent with corresponding computed strategy profiles.

In the next simulation, we assume the PUs's utilization on channels are changes. That is the packet arrival rates of PU_1, PU_2, and PU_3 are changed to $\lambda_1^{(PU)} = 0.01$, $\lambda_2^{(PU)} = 0.01$, and $\lambda_3^{(PU)} = 0.1$ respectively. According to (24) we expect that the new strategy profiles of SU_1, SU_2, and SU_3 are approximately equal and given by $\mathbf{p}_j = (0.46, 0.38,$

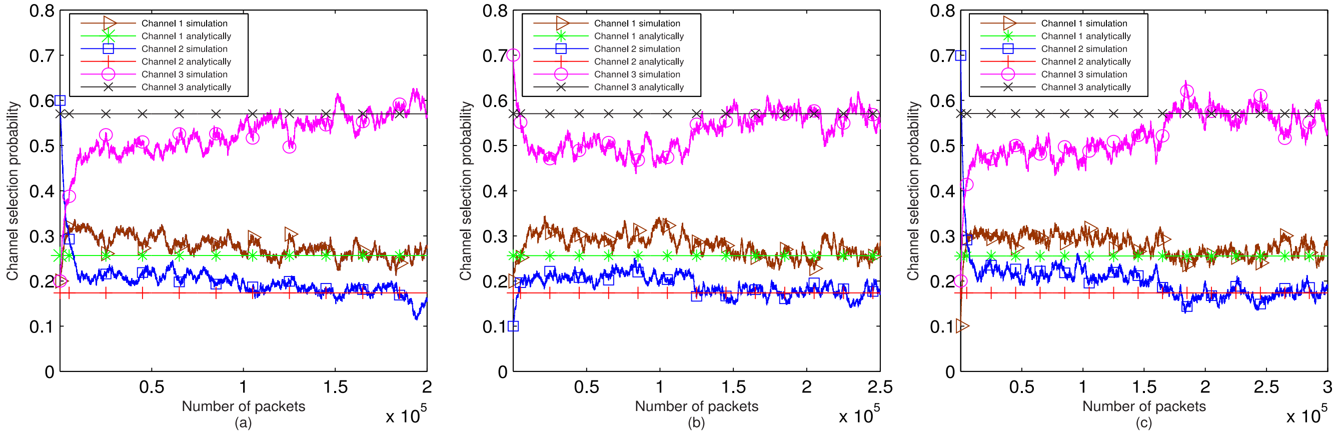

Fig. 5. Variation of the strategy profiles after SU_4, SU_5, SU_6 leave the system for a) SU_1, b) SU_2, and c) SU_3.

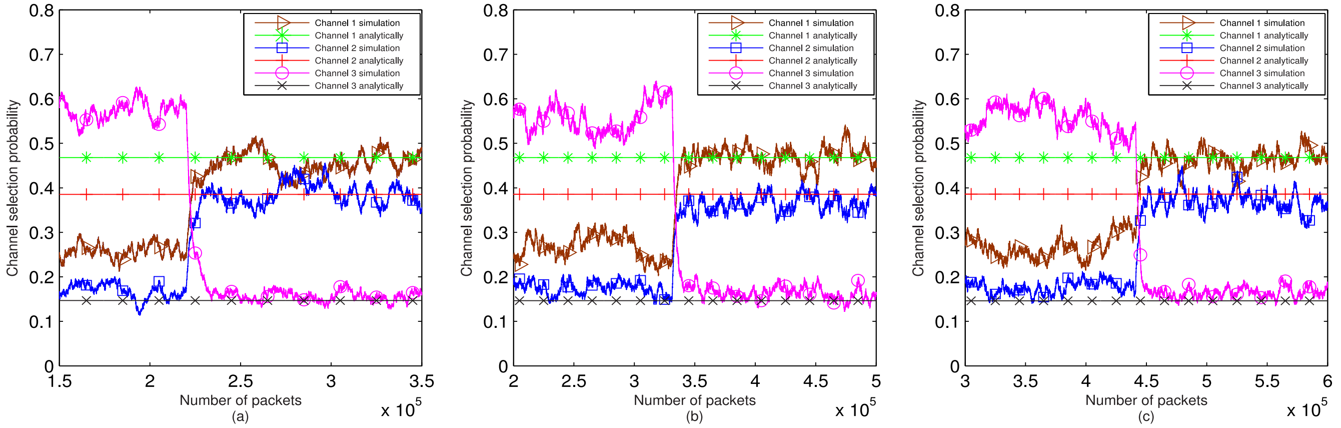

Fig. 6. Variation of the strategy profiles after PUs' arrival rates are changed for a) SU_1, b) SU_2, and c) SU_3.

$0.14), j = 1, \ldots, 3$. In Fig. 6, the variations in the strategy profiles of these SUs until convergence are depicted which shows that the agents adapt their strategies in a self-organized manner if the environment is changed.

7.2 Efficiency of the Final Strategy Profiles

Other performance metrics of the proposed multi-agent learning scheme include the average waiting time, average number of transmission deferring by other SUs, and the average number of interruptions by PUs for each packet.

Also, we compare the efficiency of the proposed scheme in these metrics with a Global Optimum Solution (GOS) which is achieved by a centralized strategy maker. This strategy maker minimizes the penalty responses of the environment for the whole system. It is assumed that this central strategy maker knows all system parameters and aims to find the optimum channel selection strategy for each SU such that the system-wide punishment is minimized. The normalized incurred expected penalty for all SUs in the system is given by (25). The logic behind defining this objective is rooted in evaluating the optimality of distributed solutions in distributed systems [33], [34], [35]. The designed global optimum solution is typically aimed to optimize a weighted sum of the objective functions of the individual agents taking into account the possible cross decisions effects. It can be assumed as the best achievable central solution and can be compared with the proposed distributed solution

$$G = \frac{1}{\sum_{j=1}^{N} \lambda_j^{(SU)}} \sum_{j=1}^{N} \lambda_j^{(SU)} \left(\sum_{i=1}^{M} p_{ji} C_{ji}(\mathbf{P}) \right), \tag{25}$$

where $\sum_{i=1}^{M} p_{ji} C_{ji}(\mathbf{P})$ is the expected penalty of SU_j on all channels. The normalizing factor $\frac{\lambda_j^{(SU)}}{\sum_{j=1}^{N} \lambda_j^{(SU)}}$ reliefs the effect of SU_j's rate on the received penalty.

On the other hand, the centralized decision maker should take into account the fairness of the system in resource allocation. The decision maker uses Jain's fairness index [36] to measure the SUs' share from the system spectrum opportunities where $\rho_j = \frac{\sum_{i=1}^{M} p_{ji} \lambda_j^{(SU)}}{\sum_{i=1}^{M} \mu_i}$ is used as the total utilization of SU_j on all channels.

The GOS is achieved by solving the optimization problem in (26) in which the system normalized expected penalty is minimized under the constraint of fairness among SUs

$$\begin{cases} \operatorname{argmin}_{\mathbf{P}} \frac{1}{\sum_{j=1}^{N} \lambda_j^{(SU)}} \sum_{j=1}^{N} \lambda_j^{(SU)} \left(\sum_{i=1}^{M} p_{ji} C_{ji}(\mathbf{P}) \right) \\ \textit{subject to} \quad \frac{\left(\sum_{j=1}^{N} \rho_j\right)^2}{N \sum_{j=1}^{N} (\rho_j)^2} \geq F, \end{cases} \tag{26}$$

where F is the fairness index which should be met. We compare the final strategy profiles of the proposed

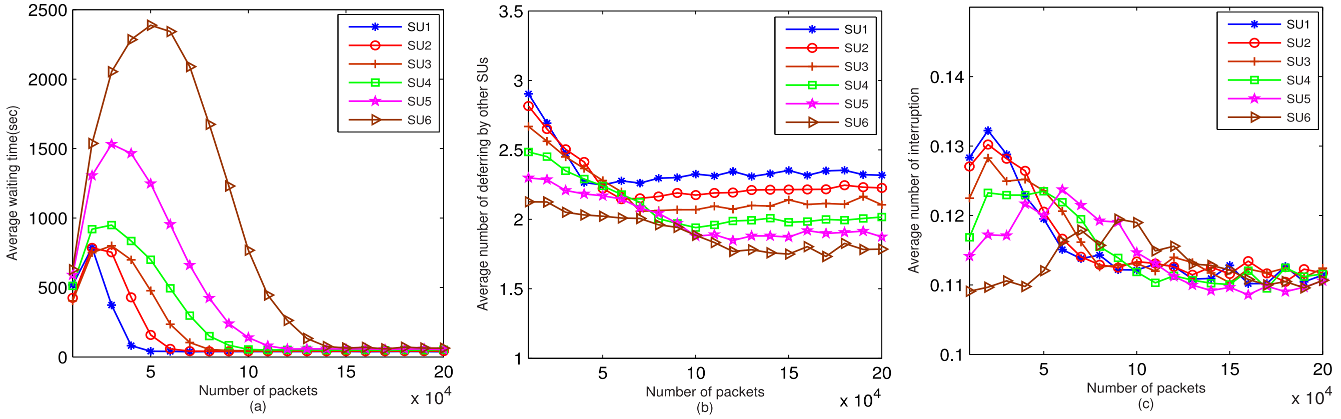

Fig. 7. Decreasing trend during the learning process: a) SUs' packets average waiting time, b) average number of transmission deferring per packet, and c) average number of PUs' interruptions per packet.

TABLE 1
Performance Evaluation and Comparison Between GOS/Proposed Scheme

SUs	Average waiting time	Average number of deferring by other SUs	Average number of interruptions by PUs
SU_1	22.9321 / 37.9831	1.4850 / 2.3177	0.1090 / 0.1114
SU_2	25.0855 / 42.5641	1.4950 / 2.2261	0.1130 / 0.1117
SU_3	25.2064 / 45.9520	0.6450 / 2.1044	0.1240 / 0.1124
SU_4	34.1601 / 51.7269	0.5400 / 2.0162	0.1750 / 0.1116
SU_5	60.2818 / 58.5037	1.5255 / 1.8734	0.0270 / 0.1106
SU_6	92.7285 / 70.2131	1.3770 / 1.7850	0.0230 / 0.1107

self-organized scheme with the GOS that has the same fairness in the mentioned performance metrics.

In Fig. 7, the packets' average waiting time, the average number of deferring by other SUs before transmission, and the average number of interruptions by PUs are shown respectively. The results in these figures are the average of 100 times run. Fig. 7a shows the variations of the packets' average waiting time during the learning process. In this figure by adapting to the environment the average waiting time of SUs' packets has a decreasing trend. The final values of these waiting times are compared with the GOS in Table 1 where both schemes meet Jain's fairness index $F \simeq 0.87$. In GOS, SU_5 and SU_6 which have greater arrival rates are incurred the greater delay to meet the fairness. Fig. 7b shows the average number of deferring by other SUs before transmission per each packet for each SU during the learning process which shows a decreasing trend. Also, the comparison with GOS is presented in Table 1. In GOS, SUs encounter less transmission deferring since the effect of collision and contention between SUs is explicitly minimized in centralized decision maker. Fig. 7c shows the average number of PU's interruption per packet for each SU which again has a decreasing trend. Also, Table 1 shows that the number of interruptions by PUs in the GOS for SU_5 and SU_6 which have greater arrival rates are less than other SUs while in the self-organized scheme these values are approximately equal. The reason is that effects of handoffs are explicitly minimized in the GOS.

8 Conclusion

We propose a distributed learning automata for spectrum management in CR networks in which SUs as intelligent agents interact with the RF environment and learn to select appropriate spectrum by the different responses of the environment in a self-organized manner. There is no prior information about the environment, PUs' arrival rates, other SUs' arrival rates, and action selection of other SUs. Also, there is no information exchange among multiple SUs. We investigate the convergence behavior and the stability property of the proposed distributed scheme analytically which are justified by simulation. In future work, we study admission control problem in CR networks when the arrival rates of SUs may exceed the channels service rates.

References

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw. J.*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[2] Q. Zhao and B.M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[3] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[4] B. Wang and K. J. R. Liu, "Advances in cognitive radio networks: A survey," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 5–23, Jan. 2011.

[5] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tutorials*, vol. 15, no. 3, pp. 1136–1159, Jul. 2013.

[6] A. He, et al., "A survey of artificial intelligence for cognitive radios, "*IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1578–1592, May 2010.

[7] M. van der Schaar and F. Fu, "Spectrum access games and strategic learning in cognitive radio networks for delay-critical applications," *Proc. IEEE*, vol. 97, no. 4, pp. 720–740, Apr. 2009.

[8] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202–3212, Jun. 2013.

[9] L. Rose, S. M. Perlaza, C. J. Le Martret, and M. Debbah, "Self-organization in decentralized networks: A trial and error learning approach," *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 268–279, Jan. 2014.

[10] K. S. Narendra, and M. A. L. Thathachar, *Learning Automata: An Introduction*. Englewood Cliffs, NJ, USA: Prentice Hall, 1989.

[11] L. Busoniu, R. Babuska, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst. Man Cybern.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[12] H. Li and Z. Han, "Competitive spectrum access in cognitive radio networks: Graphical game and learning," in *Proc. Wireless Commun. Netw. Conf.*, Apr. 2010, pp. 1–6.

[13] M. Azarafrooz and R. Chandramouli, "Distributed learning in secondary spectrum sharing graphical game," in *Proc. IEEE Global Telecommun. Conf.*, 2011, pp. 1–6.

[14] Y. Xu, Q. Wu, L. Shen, and J. Wang, "Opportunistic spectrum access with spatial reuse: Graphical game and uncoupled learning solutions," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4814–4826, Oct. 2013.

[15] M. Maskery, V. Krishnamurthy, and Q. Zhao, "Decentralized dynamic spectrum access for cognitive radios: Cooperative design of a non-cooperative game," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 459–469, Feb. 2009.

[16] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in cognitive radio networks: Global optimization using local interaction games," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 2, pp. 180–194, Apr. 2012.

[17] H. P. Shiang and M. van der Schaar, "Queuing based dynamic channel selection for heterogeneous multimedia application over cognitive radio networks," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 896–909, Aug. 2008.

[18] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *Proc. IEEE Symp. Dynamic Spectrum Access Netw.*, Apr. 2010, pp. 1–9.

[19] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1380–1391, Apr. 2012.

[20] J. Zheng, Y. Cai, N. Lu, and Y. Xu, "Stochastic game-theoretic spectrum access in distributed and dynamic environment," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4807–4820, Oct. 2015, doi: 10.1109/TVT.2014.2366559.

[21] P. S. Sastry, V. V. Phansalkar, and M. A. L. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," *IEEE Trans. Syst. Man Cybern.*, vol. 24, no. 5, pp. 769–777, May 1994.

[22] L. Husheng, "Multi-agent Q-learning of channel selection in multi-user cognitive radio systems: A two by two case," in *Proc. IEEE Conf. Syst. Man Cybern.*, Oct. 2009, pp. 1893–1898.

[23] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1823–1834, May 2010.

[24] T. Jiang, D. Grace, and Y. Liu, "Two-stage reinforcement-learning-based cognitive radio with exploration control," *IET Commun.*, vol. 5, no. 4, pp. 644–651, Mar. 2011.

[25] T. Jiang, D. Grace, and P. D. Mitchell, “Efficient exploration in reinforcement learning-based cognitive radio spectrum sharing,” *IET Commun.*, vol. 5, no. 10, pp. 1309–1317, Jul. 2011.

[26] H. Bizhani and A. Ghasemi, “Joint admission control and channel selection based on multi response learning automata (MRLA) in cognitive radio networks,” *Wireless Pers. Commun.*, vol. 71, no. 1, pp. 629–649, Jul. 2013.

[27] A. Anandkumar, N. Michael, and A. Tang, “Opportunistic spectrum access with multiple users: Learning under competition” in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[28] K. Liu and Q. Zhao, “Distributed learning in cognitive radio networks: Multi-armed bandit with distributed multiple players,” in *Proc. IEEE Int. Conf. Acoustics Speech Signal Process.*, Mar. 2010, pp. 3010–3013.

[29] Q. Zhao, L. Tong, A. Swami, and Y. Chen, “Decentralized cognitive MAC for opportunistic spectrum access in Ad Hoc networks: A POMDP framework,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 589–600, Apr. 2007.

[30] K. S. Narendra, and M. A. L. Thathachar, “On the behavior of a learning automaton in a changing environment with application to telephone traffic routing,” *IEEE Trans. Syst. Man Cybern.*, vol. 10, no. 5, pp. 262–269, May 1980.

[31] L. C. Wang, C. W. Wang, and F. Adachi, “Load balancing spectrum decision for cognitive radio networks,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 757–769, Apr. 2011.

[32] L. C. Wang, C. W. Wang, and C. J. Chang, “Modeling and analysis for spectrum handoffs in cognitive radio networks,” *IEEE Trans. Mobile Comput.*, vol. 11, no. 9, pp. 1499–1513, Sep. 2012.

[33] M. Fahimi and A. Ghasemi, “Joint spectrum load balancing and handoff management in cognitive radio networks: A non-cooperative game approach,” *Wireless Netw.*, vol. 22, no. 4, pp. 1161–1180, May 2016.

[34] C. Kim and H. Kameda, “An algorithm for optimal static load balancing in distributed computer systems,” *IEEE Trans. Comput.* vol. 41, no. 3, Mar. 1992, pp. 381–384.

[35] D. Grosu and A. Chronopoulos, “Noncooperative load balancing in distributed systems,” *J. Parallel Distrib. Comput.*, vol. 65, no. 9, pp. 1022–1034, Sep. 2005.

[36] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer system,” Digit. Equipment Corporation, Maynard, MA, USA, Tech. Res. Rep. TR-301, 1984.

[37] M. Esnaashari and M. R. Meybodi, “Irregular cellular learning automata,” *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1622–1632, Aug. 2015.

Mina Fahimi received the BS degree in computer software engineering from Al-Zahra University, Tehran, Iran, and the MSc degree in artificial intelligence, computer engineering from the K. N. Toosi University of Technology, Tehran, Iran. Her research interests include machine learning, game theory, optimization, distributed algorithms, automation and self-organized systems, and wireless networks.

Abdorasoul Ghasemi received the BS degree (with honors) from the Isfahan University of Technology, Isfahan, Iran, and the MSc and PhD degrees from the Amirkabir University of Technology (Tehran Polytechnique), Tehran, Iran, all in electrical engineering, in 2001, 2003, and 2008, respectively. He is currently an assistant professor in the Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran. His research interests include communication networks, network protocols, resource management in wireless networks, and applications of optimization and game theories in networking.