Wireless Pers Commun (2013) 71:629–649
DOI 10.1007/s11277-012-0834-9

Joint Admission Control and Channel Selection Based on Multi Response Learning Automata (MRLA) in Cognitive Radio Networks

Hannaneh Bizhani · Abdorasoul Ghasemi

Published online: 17 September 2012

Abstract We use multi response learning automata (MRLA) to control how secondary users should access the licensed primary channels in cognitive radio networks. We seek two aims in this paper: (1) estimating the availability probability of each primary channel and (2) admission control of secondary users to decrease the rate of collisions between them. We consider single and multiple secondary user scenarios. In the first scenario, the secondary user deploys learning automata to estimate the primary channel availability probability for efficient exploitation. In the second scenario, each secondary user deploys an algorithm based on MRLA to estimate primary traffic as well as the behavior of other secondary users in order to control the rate of collisions. Then, to have a better control on the rate of secondary collisions, when the number of secondary users is greater than the number of primary channels, we proposed an admission control scheme. In this scheme, some of secondary users are blocked in each time slot and do not have any interaction with the environment. The convergence of the proposed algorithms with and without admission schemes is analyzed. Simulation results are provided to show the improvement in the secondary users' total throughput and switching cost while maintaining the fairness between them.

Keywords Cognitive radio networks · Dynamic spectrum access · Multi response learning automata · Channel selection · Admission control

1 Introduction

Due to increasing demands for wireless communication services, the available radio frequency spectrum has become more scarce. Cognitive radio (CR) technology is emerged to

H. Bizhani · A. Ghasemi (✉)
Faculty of Electrical and Computer Engineering, K.N. Toosi University of Technology,
P.O. Box 163151355, Tehran, Iran
e-mail: arghasemi@eetd.kntu.ac.ir

H. Bizhani
e-mail: hanabijani@ee.kntu.ac.ir

improve the bandwidth usage efficiency by opportunistically using the traditional licensed bandwidth allocation by dynamic spectrum access (DSA) techniques. In a hierarchical DSA model [1], the aim is to share the allocated licensed spectrum of primary users (PUs) among secondary users (SUs) without causing harmful interference. Two common schemes for spectrum utilization are underlay and overlay approaches. The former imposes severe constraints on the transmission power of SUs while the latter aims to find when and where the SUs should transmit. In overlay approach, which is adopted in this paper, there are three online cognitive tasks [2]: (1) radio-scene analysis, (2) channel identification, and (3) transition power control and dynamic spectrum management. Through interaction with the environment, these three tasks form a cognitive cycle for DSA [2]. The SU can be considered as an agent that should interact with the environment to sense, learn from the feedbacks, and adjust its transmission parameters to efficiently exploit the spectrum holes.

In this work, we seek two main objectives: (1) estimating the behavior of PUs on each channel for efficient utilization of spectrum holes and (2) selecting channels appropriately considering the collisions with other SUs. Since the primary traffic distributions is not available to SUs a priori, some algorithms are required to find the availability probability of PUs in each channel. On the other hand, this algorithm should also account for the competition between SUs to exploit the channels' spectrum holes simultaneously. Therefore, the SU acts as a decision maker which selects channel sequentially to maximize its utilization of spectrum holes. We deploy learning automata (LA) in each SU for decision making and channel selection. LA helps to aggregate the received feedbacks of SUs' previous interactions with the environment. The main contributions of this paper are summarized as follow:

1. We use LA for single SU scenario of CR networks. In this scenario, the SU learns the PU's traffic distribution using an adjusted LA which interacts with the environment.
2. The multi response learning automata (MRLA) [3] for multiple SUs scenario is then discussed. With this type of learning automaton we can model the primary traffic and secondary collisions as a set of environmental feedbacks to learn the behavior of PUs and other SUs. The aim is to minimize the incurred interferences for PUs and other SUs considering the total SUs' throughput, switching cost and fairness as performance metrics.
3. When the number of SUs are greater than the number of primary channels, an admission control scheme is proposed and integrated with MRLA to restrict the number of competing SUs.

   The rest of this paper is organized as follows: the related works are presented in Sect. 2. Network model and problem statement are discussed in Sect. 3. Section 4, introduces some basic concepts on LA and MRLA. The proposed algorithm based on LA for single SU scenario and MRLA for multiple SUs scenario are presented in Sects. 5 and 6, respectively. Section 7 presents the new admission control mechanism for multiple SUs scenario. Simulation results and performance evaluation are presented in Sect. 8. Finally, we conclude the paper in Sect. 9.

2 Related Works

The practical implications of artificial intelligence (AI) for CR designs are reviewed in [4]. The cognitive term implies awareness, perception, reasoning and adjustment, and hence *learning* is an essential part of CR user. The authors in [5] model cognitive radio with different reasoning and learning engines and describe applications of these models for DSA.

Since DSA problem in single SU model is equivalent with the classical multi armed bandit problem, cognitive medium access can use tools from reinforcement machine learning [6]. In [7], a channel selection algorithm is presented which assumes particular frequency channels are free to use all the times. This assumption decreases network performance. In [8], Exp3 algorithm is used to develop an adaptive DSA protocol. An enhancement to Exp3 is also derived in [8], which uses a weighting factor that adaptively changes based on channel statistics. In [9], Rule1 algorithm is developed which is discussed to be the optimal scheme for channel selection in single SU scenario. In this algorithm, a logarithmic term in updating the SU's strategy is used to guarantee enough sampling time for each channel. If the availability probability of channel i is θ_i, the probability of accessing channel i based on Rule1 converges to the θ_i in probability. That is, each channel is selected according to its availability probability. Therefore, the SU will select the channel which has maximum spectrum holes. For the multiple SUs scenario, Rule3 is proposed. It is proved that if the number of SUs is large, the scheme in Rule3 converges to Nash equilibrium $\frac{\theta_i}{\sum_{i=1}^{N}\theta_i}$, where N is the number of primary channels. When a collision occurs and K users select the same channel i, Rule3 allocates $\frac{B_i}{K}$ bits of bandwidth to each user, where B_i is the total available bandwidth of channel i. In a practical scenario the allocation of bandwidth to competing users is not perfect as it is assumed in Rule3. In [10,11], LA is used for DSA problem assuming a stationary environment for PUs traffic distribution. That is, the traffic distribution of PUs does not change over time. However, in practical scenarios the SUs interact with non stationary environment. In addition, just single SU scenario is considered and the competition between SUs is not discussed. In [12], a LA based algorithm is developed for dynamic environments for a single SU scenario. This algorithm starts learning again when it detects a change in PUs channel's traffic distribution, which is time wasting.

3 Network Model and Problem Statement

Consider a primary network where the set of channels is denoted by $\mathcal{M} = \{1, \ldots, M\}$, where the bandwidth of channel $m \in \mathcal{M}$ is B Hertz.

We assume a time slotted system, i.e., at each time slot $t \in \{1, \ldots, T\}$, the SU selects a channel to sense for possible exploitation. At the end of each time slot, receiver sends back ACK packets to transmitter for successful transmission. The sensing is assumed to be perfect and the SU can correctly infer the presence of PU on a channel which is explored. For the multiple SUs scenario, the set of SUs is denoted by $\mathcal{N} = \{1, \ldots, N\}$.

Let channel m be free with probability θ_m and busy with probability $1 - \theta_m$. We define a Bernoulli random variable $\zeta_m(t)$, which equals 1 if channel m is free at time slot t and equals 0 otherwise. Note that the SUs are not aware of the channel availability vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$. For single SU scenario, the objective is to explore the channels and exploit their free spaces according to their availability parameters. For multiple SUs scenario, the objective is exploring the channels by the SUs simultaneously considering their competition and possible collisions. Let $C_m(t)$ be the set of SUs that select channel m at time slot t.

The SU n can send B bits over channel m at time slot t, after exploring it, if $\zeta_{m(t)} = 1$ and $|C_{m(t)}| = 1$. If $\zeta_{m(t)} = 0$, the SU will wait until the next time slot for channel exploration. If $\zeta_{m(t)} = 1$ and there is a collision with other SUs, $|C_{m(t)}| > 1$, none of the competing SUs can exploit the bandwidth of the selected channel in that time slot. Let $NonCol_{m(t)}$ be a Bernoulli random variable which equals 1 if there is not any secondary collision on channel $m(t)$ in time slot t and 0 otherwise. The total number of bits that SU $n \in \mathcal{N}$ is able to send during T time slots is given by:

$$W_n = \sum_{t=1}^{T} B.\zeta_{m(n,t)}(t).NonCol_{m(t)}(t) \tag{1}$$

where $m(n,t)$ is the selected channel at time slot t by SU n. The goal is to maximize the expectation of the total throughput of all SUs which is given by (2).

$$W = \sum_{n=1}^{N} W_n = \sum_{n=1}^{N} \sum_{t=1}^{T} B.\zeta_{m(n,t)}(t).NonCol_{m(t)}(t) \tag{2}$$

Switching from channel i to channel j is possible during time slots and incurred switching cost c. The total channel switching cost is computed by adding the number of times that each SU switches from a channel to another one during T slots. This cost should not be high in a reasonable channel access strategy.

4 Basic Concepts of Learning Automata

4.1 Learning Automata (LA)

An agent which is empowered with LA interacts with the environment and adjusts its actions according to the received response of the environment. The actions are chosen according to a probability distribution which is updated based on environment responses that the automaton obtains by performing a particular action. Details and formal definitions of LA can be found in [13]. A block diagram of a learning automaton is presented in Fig. 1.

In this figure, at time instance t, the automaton selects an action $a(t) = a_i$ from its action set $\{a_1, \ldots, a_r\}$ as an input to the environment. The action is selected based on a probability vector $\mathbf{P}(t) = \{P_1, \ldots, P_r\}$, which is updated during time slots. The initial value of this vector is typically set as $P_i(t) = \frac{1}{r}$, $\forall i$, where r is the number of actions. The environment then responses to the input by a reinforcement signal $X(t)$. Three kinds of environment models can be defined according to response $X(t)$. P-Model, in which $X(t) \in \{0, 1\}$, Q-Model, in which $X(t)$ takes discrete values in the range [0,1], and S-Model, in which $X(t)$ takes continuous values in the range [0,1]. The automaton then updates its probability vector for the next time slot $\mathbf{P}(t+1)$ based on the reinforcement scheme. $\mathbf{P}(t+1) = T[\mathbf{P}(t), a(t), X(t)]$ represents the learning algorithm. Let i be the index of selected action in time slot t, then the recurrent equation for updating $\mathbf{P}$ is defined by (3) and (4).

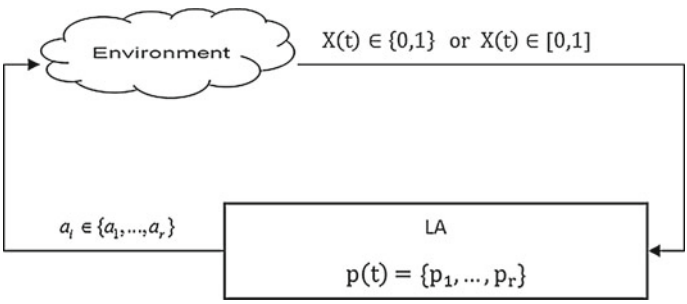

Fig. 1 Block diagram of a learning automaton

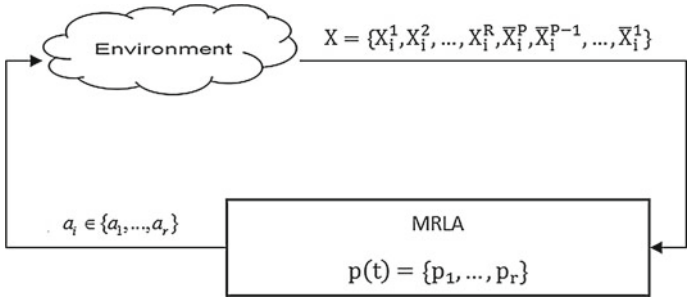

Fig. 2 Block diagram of MRLA

For reward response:

$$\begin{aligned} P_i(t+1) &= P_i(t) + \alpha.(1 - P_i(t)) \\ P_j(t+1) &= P_j(t) - \alpha.P_j(t), \quad \forall j, j \neq i \end{aligned} \tag{3}$$

For penalty response:

$$\begin{aligned} P_i(t+1) &= (1-\beta).P_i(t) \\ P_j(t+1) &= \frac{\beta}{r-1} + (1-\beta).P_j(t), \quad \forall j, j \neq i \end{aligned} \tag{4}$$

where α and β are reward and penalty parameters, respectively. In linear schemes including linear reward-penalty, L_{R-P}, linear reward-inaction, L_{R-I}, and linear reward- ϵ penalty, L_{ReP}, we have $\alpha = \beta$, $\beta = 0$, and $\beta \ll \alpha$, respectively. In [13], other non linear learning algorithms are also discussed.

One of the main advantages of LA is that, it needs no information about the environment in which it operates. Therefore, it is a good framework for a single SU in CR network, which does not have any knowledge about the traffic distribution of PUs. Another advantage of LA is its adaptive behavior which is well suited for dynamic environments like CR networks.

4.2 Multi Response Learning Automata (MRLA)

Like LA, MRLA [3] selects an action $a(t) = a_i$ from its action set $\{a_1, \ldots, a_r\}$ as an input to the environment. Figure 2 shows a block diagram of MRLA. The MRLA algorithm extends L_{R-P} scheme to Q-Models by introducing different reward and penalty parameters for different environment responses. These parameters are adjusted according to how favorable or unfavorable the environment response is.

That is, the environment response for action i is an element of the set $X = \{X_i^1, X_i^2, \ldots, X_i^R, \bar{X}_i^P, \bar{X}_i^{P-1}, \ldots, \bar{X}_i^1\}$ Where $\{X_i^1, X_i^2, \ldots, X_i^r\}$ is the reward set and $\{\bar{X}_i^P, \bar{X}_i^{P-1}, \ldots, \bar{X}_i^1\}$ is the penalty set. The corresponding reinforcement signals for the actions set X are $\alpha_i^1, \ldots, \alpha_i^r$ and $\beta_i^1, \ldots, \beta_i^r$. Also, R and P are the number of rewards and penalties defined for the environment, respectively. Assume that $0 \leq X_i^1 < \cdots < X_i^R < m_i < \bar{X}_i^P < \cdots < \bar{X}_i^1 \leq 1$, where m_i is the threshold for a response to be considered as reward or penalty. It is clear that X_i^1 is the best reward, VeryGood response, and $\bar{X}_i^1$ is the worst penalty, VeryBad response. The probability vector is updated like linear reinforcement schemes in LA.

According to [3], as a special case, reward and penalty functions for action i can be defined as $g_i^r = \eta * \alpha_i^r$ and $h_i^p = \eta * \beta_i^p$, where $0 < \eta \leq 1, 0 < \alpha_i^r, \beta_i^p < 1, r = 1, \ldots, R$,

and $p = 1, \ldots, P$. Based on α_i^r, β_i^p, r and p we have QMRLA$_{R-P}$, QMRLA$_{R-I}$ and QMRLA$_{ReP}$ schemes, where the reward parameter α_i^r, $r = 1, \ldots, R$ and the penalty parameter β_i^p, $p = 1, \ldots, P$ satisfy $1 > \alpha_i^R > \cdots > \alpha_i^2 > \alpha_i^1 > 0$, and $1 > \beta_i^P > \cdots > \beta_i^2 > \beta_i^1 > 0$. The action probability vector is updated according to the learning algorithm which is deployed.

The MRLA based learning algorithm have two useful properties [3]. First, this algorithm preserves the feasibility of the action probabilities which are always nonnegative and sum to one. The second is that, the MRLA based algorithm with positive penalty function is nonabsorbent, i.e., it is not trapped in a specific action and no action is selected with probability one. This is a desirable property for dynamic environments where the optimal action is changing over time. In other words, an action which is optimal in a specific time may not be optimal any more.

5 LA for Single SU Scenario

5.1 LA Based Dynamic Spectrum Access

In this scenario, there is only one SU in the CR network, i.e., $N = 1$. A learning automaton is configured in the SU, and sequentially selects one of the primary channels. Therefore, the automaton has M actions equivalent to the M primary channels to be selected, i.e., $a = \{a_1, a_2, \ldots, a_M\}$ and the response of the environment is $X \in \{0, 1\}$. When the SU selects a channel which is free of primary traffic, then the environment response is a reward, $X = 0$, and when the selected channel is busy with primary traffic, the response is a penalty, $X = 1$. The learning automaton of the SU selects a channel in time slot t, based on probability vector $\mathbf{P}(t) = (P_1(t), \ldots, P_M(t))$, where $P_i(t)$ is the probability of selecting channel i at time slot t. After receiving the response from the environment, the learning automaton uses a reinforcement scheme to update the probability vector $\mathbf{P}(t)$ for the next time slot. The objective of the SU is to minimize the received average penalty from the environment. In the proposed method, we use a linear scheme which uses Eqs. (5) and (6) for updating probability vector $\mathbf{P}(t+1)$, where the SU selects channel i.

$$\begin{aligned} P_j(t+1) &= P_j(t) - g_j(\mathbf{P}(t)) \text{ channel } i \text{ is free at time slot } t, \quad \text{for all } j \neq i \\ P_j(t+1) &= P_j(t) + h_j(\mathbf{P}(t)) \text{ channel } i \text{ is busy at time slot } t, \quad \text{for all } j \neq i \end{aligned} \tag{5}$$

For preserving probability measure, we should have $\sum_{j=1}^{M} P_j(t) = 1$, so that

$$\begin{aligned} P_i(t+1) &= P_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{r} g_j(\mathbf{P}(t)) \text{ when channel } i \text{ is free at time slot } t, \\ P_i(t+1) &= P_i(t) - \sum_{\substack{j=1 \\ j \neq i}}^{r} h_j(\mathbf{P}(t)) \text{ when channel } i \text{ is busy at time slot } t \end{aligned} \tag{6}$$

Also, $g_j(.)$ and $h_j(.)$ are the reward and penalty functions respectively which are continuous and nonnegative, satisfying (7) [13].

Algorithm 1 The single SU LA based DSA

Initialization:
Select α and β according to the LA scheme
$P_m(1) = \frac{1}{M}, m = 1, \ldots, M$
for $t = 1$ to T **do**
  $i =$ The selected channel based on $\mathbf{P}(t)$
  Sense channel i
  if channel i is free **then**
    Exploit channel i
    $P_i(t+1) = P_i(t) + \alpha[1 - P_i(t)]$
    $P_j(t+1) = P_j(t) - \alpha P_j(t),\ j \neq i$
  else
    $P_i(t+1) = (1-\beta).P_i(t)$
    $P_j(t+1) = \frac{\beta}{M-1} + (1-\beta).P_j(t),\ j \neq i$
  end if
end for

$$0 < g_j(P) < P_j, 0 < \sum_{\substack{j=1 \\ j \neq i}}^{M} [P_j + h_j(P)] < 1 \tag{7}$$

For all $i = 1, \ldots, M$. This assumption ensures that all the components of $\mathbf{P}(t+1)$ remain in (0,1). As an special case, in linear reinforcement schemes the reward and penalty functions are given by (8).

$$g_j(\mathbf{P}(t)) = \alpha P_j(t), h_j(\mathbf{P}(t)) = \frac{\beta}{M-1} P_j(t) \tag{8}$$

where α and β are reward and penalty parameters and $0 < \alpha < 1, 0 \leq \beta < 1$ [13]. Using (8) in (5) and (6), one can obtain the updating scheme which is presented by (3) and (4). Pseudo code of the proposed LA based single SU algorithm for channel selection is presented in Algorithm 1.

In the initialization phase, we set the probability of all channels for the first time slot to $\frac{1}{M}$. This is because at this time slot, the automaton does not have any information about the availability probability of channels. It will attain information about these probabilities in the consecutive time slots by interacting with the environment and updating vector $\mathbf{P}(t)$.

5.2 Analysis of LA Based DSA

The convergence properties of LA directly depends on the kind of reinforcement scheme which is used. For example, linear reward-penalty (L_{R-P}) and linear reward-ϵ penalty ($L_{R\epsilon P}$) schemes converge in distribution but linear reward-inaction (L_{R-I}) scheme converges with probability one to the optimal action. According to [13], L_{R-P} and $L_{R\epsilon P}$ are ergodic schemes and they have no absorbing states. An ergodic scheme is characterized by the property that the Markov process $\{P(t)\}_{t \geq 0}$ which is generated by the scheme is an ergodic process.

According to [13], if L_{R-P} based channel selection scheme is used in Algorithm 1, the final channel selection probabilities converge in distribution to a random variable with mean (9), independent of its initial value $\mathbf{P}(0)$.

$$\lim_{t \to \infty} E[P_i(t)] = \frac{1/1-\theta_i}{\sum_{i=1}^{M} 1/1-\theta_i} \tag{9}$$

If L_{ReP} scheme is used, Algorithm 1 converges in distribution to a normal process with mean (10), independent of its initial value $\mathbf{P}(0)$ [13].

$$\bar{P}_i = \varepsilon \frac{(1-\theta_H)}{(M-1)(\theta_i+\theta_H)}, i = 1, \ldots, M \text{ and } i \neq H$$
$$\bar{P}_H = 1 - \sum_{i \neq H} \bar{P}_i \tag{10}$$

where H is the index of channel with highest availability probability, i.e., $\theta_H \geq \theta j, j = 1, \ldots, M, j \neq H$. We should note that, if ε is selected sufficiently small the best channel is most explored by the SU. For L_{R-I} learning scheme, one of the elements of the action probability vector converges to one. The selected channel depends on the initial value of $\mathbf{P}(0)$. This scheme is not appropriate for dynamic environment of CR because of the existence of absorbing states.

6 Multiple SUs Scenario

6.1 MRLA Based Dynamic Spectrum Access

We use MRLA to control both primary and secondary collisions, in multiple SUs scenario. Each SU deploys a MRLA algorithm. These automata try to reduce both primary and secondary collisions. We should note that, avoiding primary collisions is more important than avoiding secondary collisions. When the number of SUs in the network is sufficiently larger than the number of primary channels, i.e., $N \gg M$, the secondary collision on channels with higher probability of being free, is inevitable. Therefore, in the proposed method, the environment response is a reward when a SU selects a channel which is free of PUs, even if other SUs select this channel. However, the value of the reward parameter is greater when just one SU selects this channel. The SU receives a penalty when it selects a channel which is busy by PUs. Therefore, in the proposed scenario, we have two rewards and one penalty. In other words, if a SU selects channel i, the environmental feedbacks are defined as follow:

- VERYGOOD: Channel i is free of PUs and no other SUs selects this channel (rewarded heavily with α^1)
- GOOD: Channel i is free of PUs and is also selected by other SUs (rewarded marginally with α^2)
- BAD: Channel i is busy by PUs (penalized with β)

Figure 3 illustrates different SUs which sense primary channels and receive different kinds of response from the environment. One can define more reward and penalty elements in each set by considering parameters such as power, channel switching, throughput and etc. In MRLA scenario, the environment response is an element of $\{\alpha^1, \alpha^2\}$ or is a penalty β. Note that a VERYGOOD response is more favorable than GOOD, hence the reward parameter α^1 is greater than α^2, i.e., $\alpha^1 > \alpha^2$.

Let $Res(t)$ be the response of the environment at time slot t when action i is selected. We can define (11) and (12) to be the probabilities for the reward and penalty responses, respectively.

$$d_i = \Pr[Res(t) \in \{\alpha^1, \alpha^2\} | a(t) = a_i] = \theta_i \tag{11}$$
$$c_i = \Pr[Res(t) = \beta | a(t) = a_i] = 1 - \theta_i \tag{12}$$

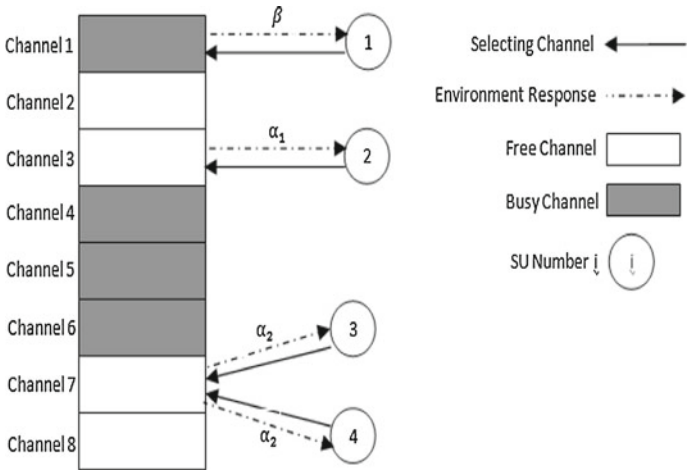

Fig. 3 Accessing primary channels. User #1 gets BAD response (β) from the environment because it selects channel 1 which is busy by PU. User #2 selects channel 2 which is free, therefore it gets VERYGOOD response (α^1). User #3 and #4 select channel 7 which is free, but because of collision between them, the environment response is good (α^2)

On the other hand, since at least one of these responses is occurred, we should have:

$$d_i^1 + d_i^2 + c_i = 1 \tag{13}$$

Reward and penalty functions are defined $g_i^r(.) = \eta\alpha^r$ and $h_i(.) = \eta\beta$, respectively. Where η is a random variable in the range (0,1]. Using the specified reward and penalty functions, if action i is selected, the probability vector is updated based on (14) and (15) for reward and penalty responses, respectively.

$$\begin{aligned} P_i(t+1) &= P_i(t) + \eta\alpha^r[1 - P_i(t)] \\ P_j(t+1) &= P_j(t) - \eta\alpha^r P_j(t), \quad \forall j \neq i \end{aligned} \tag{14}$$

$$\begin{aligned} P_i(t+1) &= P_i(t) - \eta\beta^p P_i(t) \\ P_j(t+1) &= P_j(t) + \eta\beta^p * \left[\frac{1}{M-1} - P_j(t)\right], \quad \forall j \neq i \end{aligned} \tag{15}$$

Using (14), the SU increases the probability of selecting channel i and decreases the probabilities of other channels for reward response. (15) is also interpreted for penalty response. The pseudo code of the proposed MRLA based channel selection algorithm for multiple SUs scenario is presented by Algorithm 2.

6.2 Analysis of MRLA Based DSA

According to [3], MRLA has the ergodicity property and the sequence $\{\mathbf{P}(t)\}$ converges in distribution to a random variable p^* which its distribution function is independent of $\mathbf{P}(0)$. Therefore, the proposed MRLA based DSA converges according to Proposition 1.

Proposition 1 *The MRLA based DSA for multiple SUs scenario when* $\alpha_i^1 d_i^1 + \alpha_i^2 d_i^2 + \beta_i c_i = const, \quad \forall i = 1, \ldots, M$ *is ergodic and* $P^n(t), n = 1, \ldots, N$ *converges in distribution to a random variable with mean*

Algorithm 2 The multiple SUs MRLA based DSA

Initialization:
Select $\{\alpha^1, \alpha^2\}$ and β parameters according to MRLA scheme
$P_m^n(1) = \frac{1}{M}, m = 1, \ldots, M, n = 1, \ldots, N$
for $t = 1$ to T **do**
  for $n = 1$ to N **do**
    $i =$ The selected channel based on $\mathbf{P}^n(t)$
    Sense channel i
    if channel i is free **then**
      Exploit channel i
      if no secondary collision exist **then**
        $r = 1$
      else
        $r = 2$
      end if
      $P_i^n(t+1) = P_i^n(t) + \eta\alpha^r[1 - P_i^n(t)]$,
      $P_j^n(t+1) = P_j^n(t) - \eta\alpha^r P_j^n(t), j \neq i$
    else
      $P_i^n(t+1) = P_i^n(t) - \eta\beta P_i^n(t)$
      $P_j^n(t+1) = P_j^n(t) + \eta\beta * [\frac{1}{M-1} - P_j^n(t)], j \neq i$
    end if
  end for
end for

$$\lim_{t\to\infty} E[P_i(t)] = \frac{1/\beta(1-\theta_i)}{\sum_{j=1}^{M} 1/\beta(1-\theta_i)} = \frac{1/(1-\theta_i)}{\sum_{j=1}^{M} 1/(1-\theta_i)} \tag{16}$$

Independent of the initial probability $\mathbf{P}^n(0)$.

Proof Please see [3]. □

Comment 1 Equation (16) is for MRLA$_{R-P}$ scheme. Because in linear reward-penalty scheme $\alpha_i^r = \beta_i^p$, which implies ergodicity condition,

$$\sum_{r=1}^{R} \alpha_i^r d_i^r + \sum_{p=1}^{P} \beta_i^p c_i^p = const, \quad \forall i,$$

which says that, the sum of the reward probability rates, $\alpha_i^r d_i^r$, plus the sum of penalty probability rates, $\beta_i^p c_i^p$, should be the same for all actions [3].

Comment 2 For other types of MRLA schemes like MRLA$_{R-I}$ and MRLA$_{ReP}$ convergence is exactly the same as one described in Sect. 5.2.

7 Admission Control Mechanism

7.1 ψ-MRLA Based Dynamic Spectrum Access

In this section, a new method is introduced for admission control in order to decrease the rate of secondary collisions. In this mechanism we use $Action$ parameter to block some of SUs in each time slot, that is, at each time slot the algorithm admits each SU with probability ψ ($Action$ mode) and blocks it with probability $1 - \Psi$ ($NoAction$ mode). The aim of using this parameter is to block some SUs when the rate of secondary collisions is high, in order

to improve the total performance. Therefore, there is another step before channel selection. If $NoAction$ mode is selected by a SU, it should try to access channels in the next time slot. Otherwise, it uses Algorithm 2. In the following, we explain how to adaptively adjust the blocking probability.

The blocking probability should be adjusted in each time slot according to the number of SUs which are competing for each channel. In order to adjust this parameter, the SUs change their ψ parameter based on the rate of collisions in the network. When the rate of collision is high, the SU should decrease Ψ in order to decrease the number of SUs choosing $Action$ mode. On the other hand, the SU should increase it when it senses the channel with no secondary collision. The updating equation for Ψ is given in (17).

$$\begin{cases} \psi(t+1) = \min(\psi(t) + \mu, 1)\,, & \text{no secondary collision} \\ \psi(t+1) = \max(\psi(t) - \mu, 0)\,, & \text{secondary collision} \end{cases} \tag{17}$$

where μ is a uniform random variable between (0,1) selected by each SU when it plugs into the network and min and max are for bounding $\psi(t)$ between 0 and 1.

This algorithm is operating as the same as MRLA based multiple SUs DSA. The only difference is that, ψ-MRLA uses ψ to block some of SUs in each time slot. In order to consider the fairness between SUs, when a SU selects $NoAction$ mode it also increases its Ψ parameter. This will increase the contribution probability of this SU in the next time slot. In $NoAction$ mode, SUs do not update their channel availability probability because they do not interact with and receive feedback from environment.

7.2 Analysis of ψ-MRLA Based DSA

It can be proved that the sequence $\{\mathbf{P}(t)\}$ in ψ-MRLA algorithm converges in distribution to a random variable p^* given by (16), in MRLA_{R-P} reinforcement scheme. Other kinds of reinforcement schemes like MRLA_{R-I} and MRLA_{ReP} are the same as Sect. 5.

The main difference of Ψ-MRLA is that the SUs do not update their channel access probabilities in all time slots. That is, each SU updates its access probabilities according to its Ψ parameter. In the following, using the asynchronous algorithm model of [14], we show that the final value which $\psi - MRLA$ converges to is the same as MRLA. Let $\mathbf{T} = \{1, 2, \ldots, T\}$ denotes the set of all time slots. Also, let $\tilde{\mathbf{T}}_i \subseteq \mathbf{T}$ be the set of time slots at which SU i is in $Action$ mode and updates its probability vector based on environment feedbacks. At time slots $t \notin \tilde{\mathbf{T}}_i$, user i is in $NoAction$ mode, and its probability vector is left unchanged. The probability updating equation for SU i is now given by:

$$\mathbf{P}(t+1) = \begin{cases} f(\mathbf{P}(t)) & t \in \tilde{\mathbf{T}}_i \\ \mathbf{P}(t) & otherwise \end{cases} \tag{18}$$

where $f(.)$ is the updating function defined by (14) and (15).

Proposition 2 *The ψ-MRLA algorithm converges in distribution to a random variable which MRLA converges to, independent of the initial probability* $\mathbf{P}(0)$.

Proof According to [15], when $f(\mathbf{P}(t))$ is a standard function, the iterative algorithm using $\mathbf{P}(t+1) = f(\mathbf{P}(t))$ is called the standard algorithm. We proved in Appendix that ψ-MRLA algorithm is a standard algorithm. Also, it preserves the feasibility of the action probability space, i.e., at each iteration of ψ-MRLA, the action probabilities are always nonnegative and sum to 1. According to [15], if $f(\mathbf{P})$ is feasible, then from any initial probability vector $\mathbf{p}$, the asynchronous standard algorithm converges to $\mathbf{p}^*$. Since ψ-MRLA is feasible and standard, then starting with any initial probability $\mathbf{P}(0)$, ψ-MRLA converges to $\mathbf{p}^*$ like MRLA. □

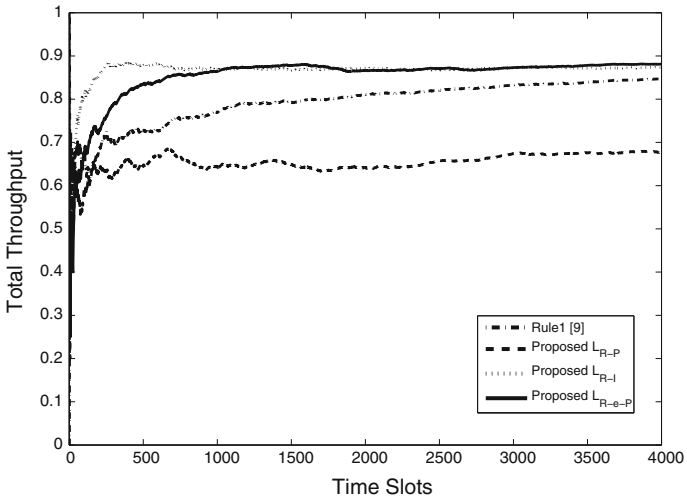

Fig. 4 Total throughput of LA–DSA methods compared to Rule1 in stationary environment

8 Performance Evaluation

Three performance criteria are considered in our simulation and are compared with other methods. The first is the total throughput which is defined by (2). The second criterion is Jain's fairness index [16]. We use this index in order to show how fairly the algorithm is performing. Jain's index is used to rate the fairness of different schemes on how the SUs exploit the spectrum white spaces. That is, if the throughput of SU i is x_i, then the Jain's fairness index is given by:

$$J(x_1, x_2, \ldots, x_N) = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N.\sum_{i=1}^{N} {x_i}^2} \tag{19}$$

This index ranges from $\frac{1}{N}$ (worst case) to 1 (best case), and its maximum value is achieved when all SUs receive the same allocation. We use the total channel switching cost as the third criterion which is defined in Sect. 3.

We assume 10 primary channels in all simulations and the simulations are done for T = 4,000 slots. We use θ = [0.90 0.300.48 0.21 0.48 0.67 0.36 0.40 0.23 0.86] for stationary network. It means that for all T time slots, the traffic distribution for primary channel i, is fixed and equal to θ_i. For the dynamic environment we use θ = [0.10 0.20 0.30 0.90 0.19 0.10 0.19 0.39 0.49 0.19] for the first T/2 time slots and θ = [0.90 0.11 0.41 0.10 0.10 0.29 0.10 0.10 0.10 0.10] for the second T/2 time slots.

8.1 Single SU Scenario

We use L_{R-P} with $\alpha = \beta = 0.09$, L_{R-I} with $\alpha = 0.09$, and L_{ReP} with $\alpha = 0.09$ and $\beta = 0.009$. The throughput of these three schemes as well as Rule1 [9] in a stationary environment is shown in Fig. 4. This Figure shows that L_{R-I} and L_{ReP} outperform Rule1 in total throughput. L_{R-I} is an absolutely expedient scheme which converges to the optimal action with probability one. The SU which employs L_{R-I} finds the best available channel

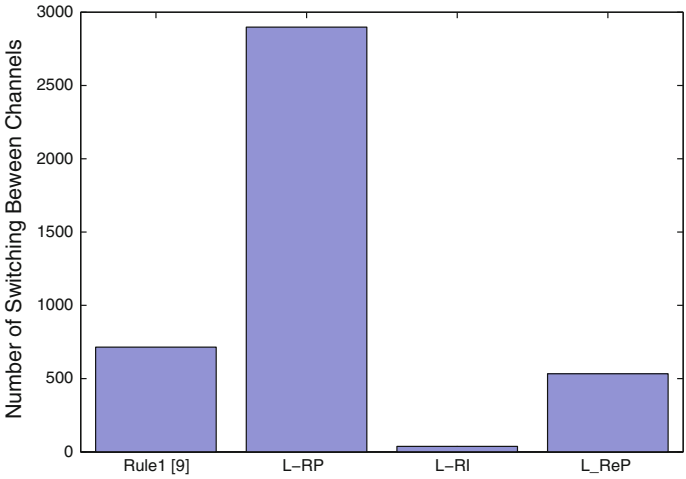

Fig. 5 Number of channel switching for LA–DSA methods compared to Rule1 in a stationary environment, with 4,000 time slots

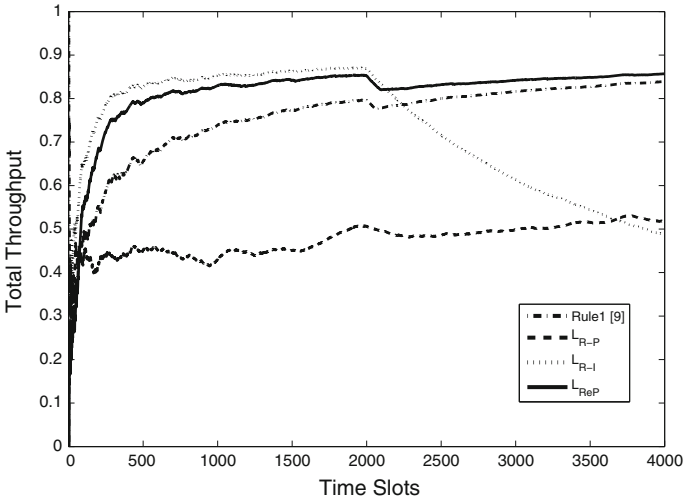

Fig. 6 Total throughput of LA–DSA methods compared to Rule1 in a dynamic environment

after some time slots. L_{R-I} sticks to the best channel until the end of time slots. This leads to lower switching cost, which is presented in Fig. 5. In this figure the total number of switching between channels is depicted.

If the primary traffic distribution changes over time slots, then L_{R-I} performance degrades, because of absorbing state. It can not find the new best channel because $\beta = 0$ and there is no updating whenever the best channel is changed. This fact is presented in Fig. 6, when the total throughput is decreasing after time slot 2,000.

On the other hand, L_{ReP} performs the same as L_{R-I} but with a very small value of penalty parameter, i.e., $\beta \ll \alpha$. This reinforcement scheme is ϵ—*optimal*. We find from Fig. 6 that L_{ReP} also performs well in dynamic environment and it can adapt quickly to the

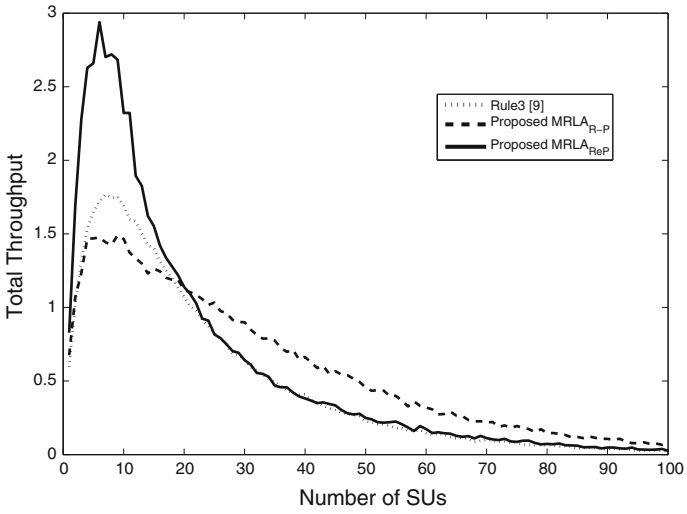

Fig. 7 Total throughput of MRLA–DSA methods compared to Rule3 with different number of SUs in the network

environment. Switching cost in L_{ReP} is more than L_{R-I}, because in contrast to L_{R-I}, L_{ReP} gives a chance to other channels to be selected.

L_{R-P} has different behavior compared to L_{R-I} and L_{ReP}. As $\alpha = \beta$ in L_{R-P}, it updates action probabilities with the same rates when there is reward or penalty response. As the result, in L_{R-P} algorithm, the probability of selecting a non best channel is higher than other methods. This causes more channel switching as well as increasing the chance of selecting less available channels. Therefore, the total throughput has the lowest value compared to other schemes.

8.2 Multiple SUs Scenario

For MRLA$_{R-P}$, the parameters are set to $\alpha^1 = \alpha^2 = \beta = 0.09$, and for MRLA$_{ReP}$, we set $\alpha^1 = 0.09, \alpha^2 = 0.01$ and $\beta = 0.01$. The results are compared to Rule3 [9]. Figure 7 shows the total throughput of these schemes for different number of SUs in the network. It is clear that, as the number of SUs is increasing the total throughput decreases due to secondary collisions. Figure 8 shows the Jain's fairness index for these schemes. We do not consider MRLA$_{R-I}$ since the SUs will select the same channel and collide in all time slots.

As there are 10 primary channels the peak of Fig. 8 is happened at 10 SUs. We can explain the behavior of these algorithms using Fig. 9.

Figure 9 shows the probability of selecting each of 10 primary channels by 10 SUs. As Fig. 7 shows, MRLA$_{ReP}$ has higher total throughput than other methods, when the number of SUs is equal to the number of primary channels. MRLA$_{ReP}$ allocates channels to SUs in a distributed manner. Because of this fact, Jain's fairness index decreases at $N = 10$. However, MRLA$_{R-P}$ considers high probability for more available channels and low probability for others. As Fig. 9 shows, channel 1 and 10 have higher probabilities in MRLA$_{R-P}$ model because channel 1 and 10 are the best available channels. This behavior is well performed when there are large number of SUs in the network, i.e., $N \gg M$. In that case, the majority of SUs select channel 1 and 10 and there will be high collision rate on these channels. But

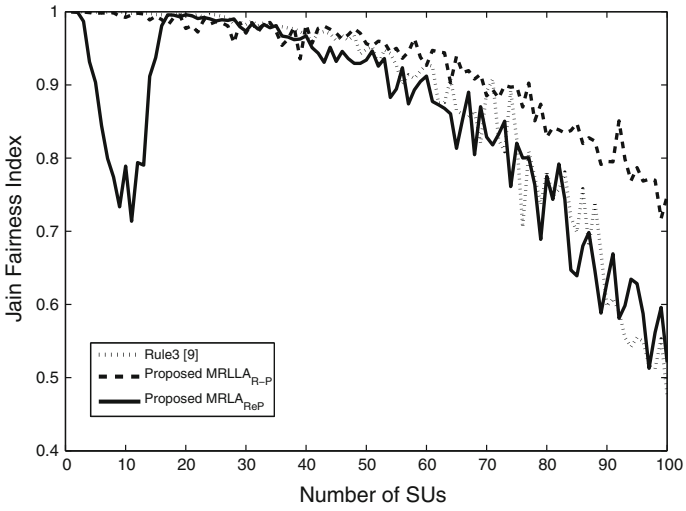

Fig. 8 Jain's fairness index of MRLA–DSA methods compared to Rule3 with different number of SUs in the network

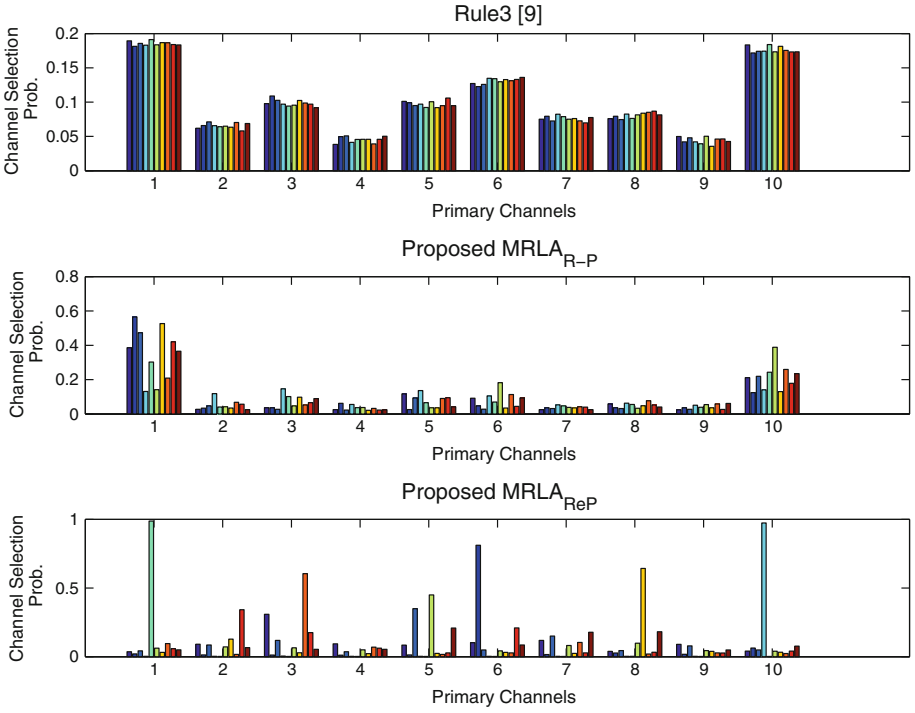

Fig. 9 Probability of selecting 10 primary channels for 10 SUs in Rule3, MRLAR-P and MRLAR-P after convergence

a few SUs will select other channels which have successful transmission if that channel is free. This fact helps MRLA$_{R-P}$ scheme to have more total throughput compared to other schemes, when the number of SUs are greater than the number of primary channels.

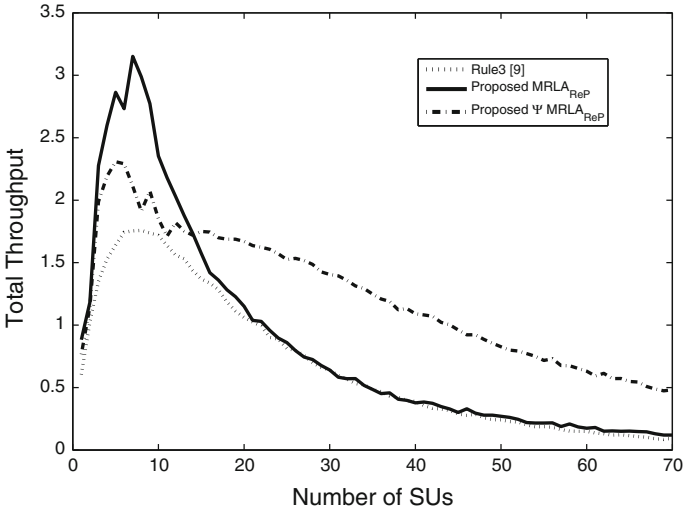

Fig. 10 Total throughput of ψ-MRLA–DSA methods compared to Rule3 and MRLA–DSA, with different number of SUs

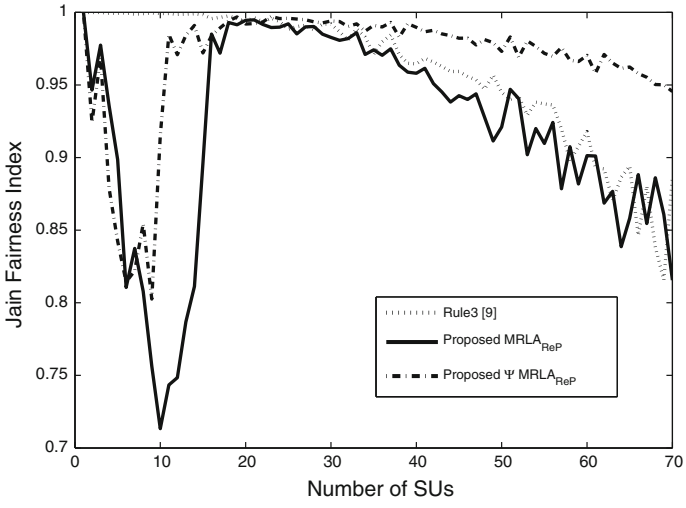

Fig. 11 Jain's fairness index of ψ-MRLA DSA method compared to Rule3 and MRLA DSA, with different number of SUs

8.3 Admission Control Mechanism

In this section, we will explain the behavior of MRLA$_{ReP}$, using $Action$ parameter for admission control. In a stationary environment, we use MRLA$_{Rep}$ in which the reward and penalty parameters are set to $\alpha^1 = 0.09, \alpha^2 = 0.01, \beta = 0.01$ in all cases. Then we consider the proposed admission control in Sect. 7. Decreasing the number of SUs in the network reduces the number of secondary collisions. The proposed ψ-MRLA DSA uses this fact to increase the total throughput in CR networks. Figure 10 shows that, ψ-MRLA DSA. Figure 11 shows the Jain's fairness index of these methods.

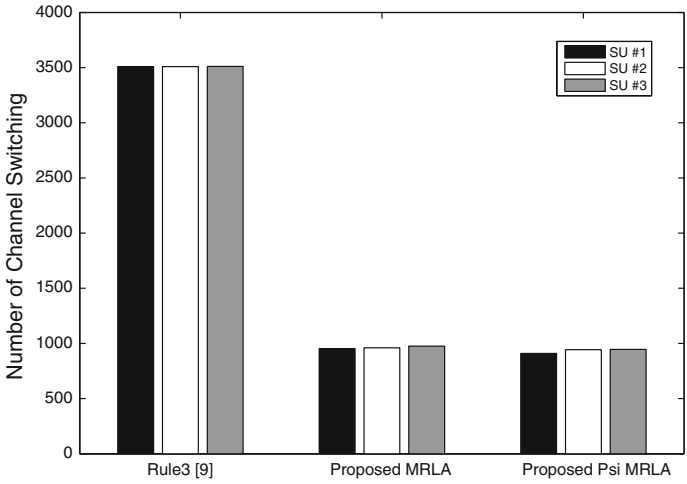

Fig. 12 Number of channel switching for different methods in 4,000 time slots for three SUs

Again, since MRLA$_{ReP}$ allocates channels in a distributed manner, there is a decrease in Jain's fairness index when $M = N = 10$. As it is shown in Fig. 11, ψ-MRLA DSA outperforms Rule3 regarding the Jain's fairness index, when the number of SUs is greater than the number of Primary channels. The Jain's fairness index is $\frac{k}{N}$, when k users equally share the resource and the other $n - k$ users receive zero allocation. As the number of k increases, this index also is increased. Since in ψ-MRLA DSA, there is lower secondary collisions when $N \gg M$, more SUs have successful transmission. Therefore, the Jain's fairness index will decrease.

Figure 12 shows the number of channel switching when three SUs are competing to exploit 10 primary channels. From this figure we find that, MRLA DSA and ψ-MRLA DSA have lower channel switching than Rule3 due to their distributed behavior on selecting channels.

Finally, the total number of collisions is depicted in Fig. 13. we find that the proposed Ψ-MRLA$_{ReP}$ efficiently decreases the number of secondary collisions by admission control mechanism.

8.4 Convergence Issues

Figure 14 shows the convergence behavior of MRLA DSA for MRLA$_{R-P}$ scheme. This results is the same as what is expected by (16). As we expected, Fig. 15 shows that channels 1, 6, and 10 are selected by three existing SUs in MRLA$_{\mathrm{ReP}}$ scheme.

To compare the rate of convergence, we summarized the required number of iteration for convergence in Table 1. Generally speaking, the value of reward and penalty parameters in LA based algorithms play an important rule in convergence speed. These results are computed by averaging 100 times of running each algorithm for three SUs scenario. The results show that the convergence speed are almost the same.

9 Conclusion

We use learning automata in single secondary user scenario and multi response learning automata in Multiple secondary users scenario for dynamic spectrum access algorithm, in

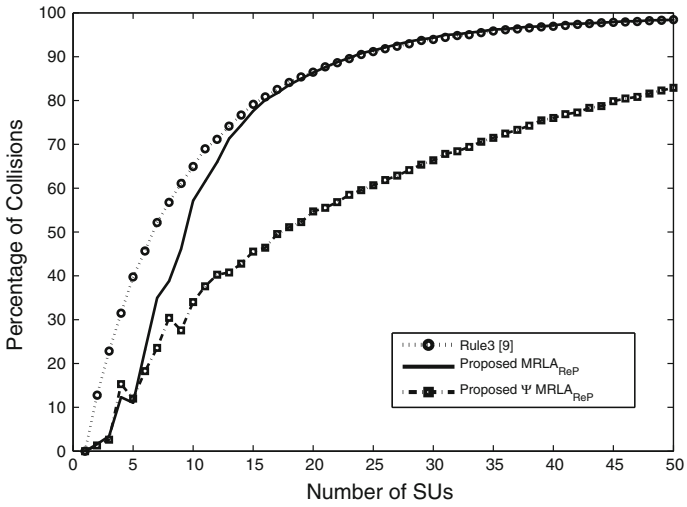

Fig. 13 Percentage of collisions for ψ-MRLA–DSA methods compared to Rule3, with different numbers of SUs

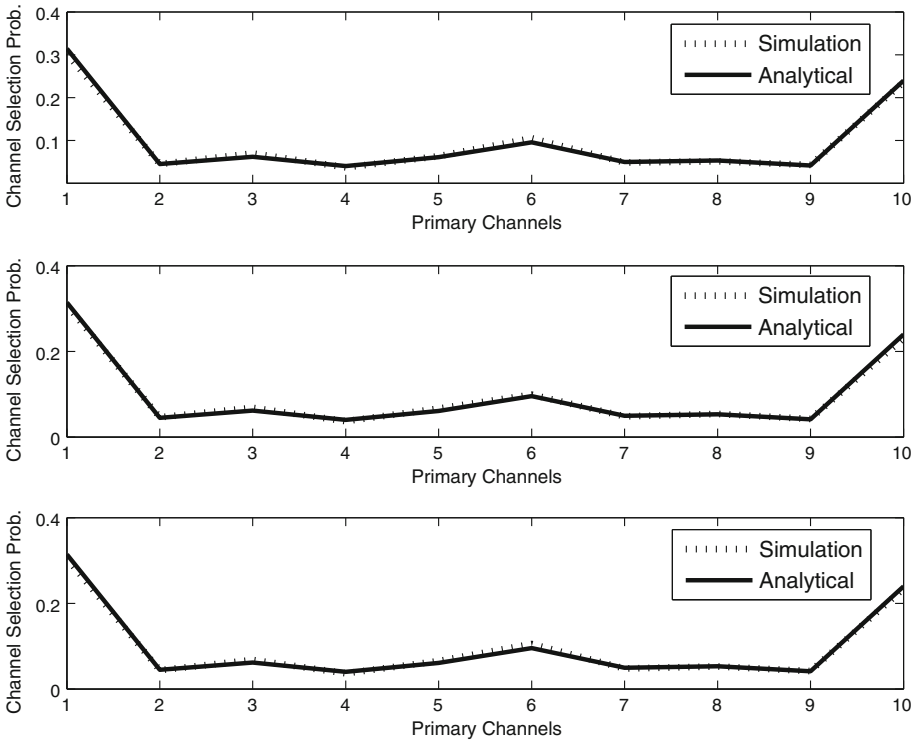

Fig. 14 Convergence of channel selection probabilities for three SUs in proposed ψ MRLA$_{R-P}$ method

cognitive radio networks. By deploying learning automata, the existing secondary user can exploit channels better. It is showed that, learning automata based dynamic spectrum access algorithm performs well when L_{ReP} scheme is used. Also, we show that multi response

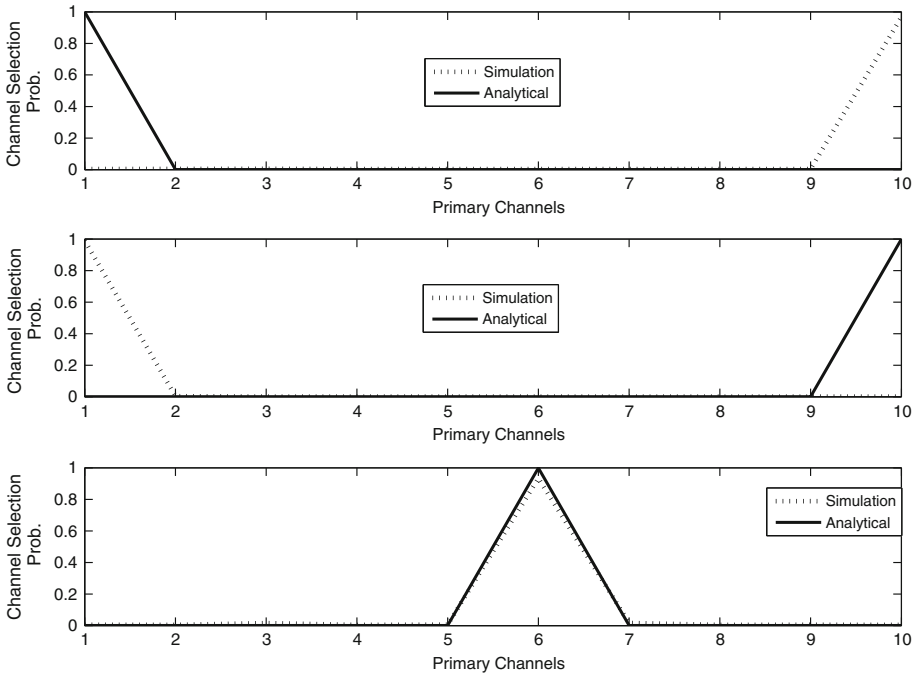

Fig. 15 Convergence of channel selection probabilities for three SUs in proposed ψ MRLA$_{ReP}$ method

Table 1 The required number of iterations until the algorithms convergence

Algorithm	Convergence time
Rule3 [9]	441
Proposed MRLA$_{ReP}$	393
Proposed Ψ MRLA$_{ReP}$	412

learning automata based dynamic spectrum access algorithm can control the competition between secondary users as well as exploiting available channels. This leads to decrease the rate of collisions and increase the total throughput. Also, we proposed an admission control mechanism to restrict the number of competing secondary users. Switching cost and convergence of all algorithms are discussed and compared to recent schemes.

Acknowledgments This research was in part supported by a grant from ITRC.

Appendix

Function $f(\mathbf{P})$ is standard, if for all $\mathbf{P} \geq 0$ three properties are satisfied: (1) *Positivity*: $f(\mathbf{P}) > 0$, (2) *Monotonicity*: If $\mathbf{P} > \mathbf{P}'$, then $f(\mathbf{P}) \geq f(\mathbf{P}')$, and (3) *Scalability*: For all $\nu > 1$, $\nu f(\mathbf{P}) > f(\nu \mathbf{P})$ [15].

Using (14) and (15), the positivity property is implied by a nonzero value of each component of $\mathbf{P}(t)$.

Noting that, if LA has the following property it is absolutely monotonic [17].

$$\frac{\phi_1(t)}{P_1} = \frac{\phi_2(t)}{P_2} = \ldots = \frac{\phi_m(t)}{P_m} = \lambda(\mathbf{P})$$

In Ψ-MRLA, we have $\phi_j(t) = g_j^p(X(t))P_j(t)$, for reward response or $\phi_j(t) = h_j^p(X(t))P_j(t)$, for penalty response, which leads to $\lambda(\mathbf{P}) = \eta\alpha_j^r$ for reward response or $\lambda(\mathbf{P}) = \eta\beta_j$ for penalty response. Therefore, $f(\mathbf{P})$ is absolutely monotonic.

For the scalability property we have:

- For reward function: $f(\nu\mathbf{P}) = \nu\mathbf{P} + \eta\alpha[1-\nu\mathbf{P}]$ and $\nu f(\mathbf{P}) = \nu\mathbf{P} + \nu\eta\alpha[1-\mathbf{P}]$ Therefore, $\nu f(\mathbf{P}) - f(\nu\mathbf{P}) = \eta\alpha(\nu-1) > 0,\ \forall\nu > 1, 0 < \alpha < 1, 0 < \eta < 1$ Which leads to: $\nu f(\mathbf{P}) > f(\nu\mathbf{P})$.
- For penalty function: $f(\nu\mathbf{P}) = \nu\mathbf{P} + \eta\beta\frac{1}{|a|-1} - \nu\eta\beta\mathbf{P}$ and $\nu f(\mathbf{P}) = \nu\mathbf{P} + \nu\eta\beta\frac{1}{|a|-1} - \nu\eta\beta P$ Therefore, $\nu f(\mathbf{P}) - f(\nu\mathbf{P}) = \eta\beta\frac{1}{|a|-1}(\nu-1) > 0,\ \forall\nu > 1, 0 < \beta < 1, 0 < \eta < 1$

Therefore, we have $\nu f(\mathbf{P}) > f(\nu\mathbf{P}),\ \forall\nu > 1$ in all cases and the scalability property of the updating function is proved.

References

1. Sadler, B. M., & Zhao, Q. (2007). A survey of dynamic spectrum access. *IEEE Signal Processing Magazine, 24*(3), 79–89. doi:10.1109/MSP.2007.361604.
2. Haykin, S. (2005). Cognitive radio: Brain-empowered wireless communication. *IEEE Journal on Selected Areas in Communications, 23*(2), 201–220. doi:10.1109/JSAC.2004.839380.
3. Economides, A. A. (1996). Multiple response learning automata. *IEEE Systems, Man, and Cybernetics Society, 26*(1), 153–156. doi:10.1109/3477.484448.
4. Newman, A. K. K. B., Gaeddert, T. R., Menon, J. K. K., Tirado, R. M., Neel, L., Reed, J. J. Y. Z., Tranter, J. H., & He, W. H. (2010). A survey of artificial intelligence for cognitive radios. *IEEE Transactions on Vehicular Technology, 59*(4), 1578–1592. doi:10.1109/TVT.2010.2043968.
5. Hecker, J., Stuntebeck, E., & O'Shea Clancy, T. (2007). Applications of machine learning to cognitive radio networks. *IEEE Wireless Communications, 14*(4), 47–52. doi:10.1109/MWC.2007.4300983.
6. Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge: MIT Press.
7. Tekin, C., Hong, S., & Stark, W. (2009). Enhancing cognitive radio dynamic spectrum sensing through adaptive learning. In MILCOM, IEEE, *Military communications conference* (pp. 1–7, 18–21). doi:10.1109/MILCOM.2009.5379925.
8. Cesa-Bianchi, N., Freund, Y., & Auer, R. S. P. (2003). The nonstochastic multi-armed bandit problem. *SIAM Journal on Communicating, 32*(1), 48–77. doi:10.1137/S0097539701398375.
9. Lai, L., El Gamal, H., Jiang, H., & Poor, H. V. (2011). Cognitive medium access: Exploration, exploitation, and competition. *IEEE Transactions on Mobile Computing, 10*(2), 239–253. doi:10.1109/TMC.2010.65.
10. Song, Y., Fang, Y., & Zhang, Y. (2007). Stochastic channel selection in cognitive radio networks. In *IEEE global telecommunications conference* (pp. 4878–4882, 26–30). doi:10.1109/GLOCOM.2007.925.
11. Li, H., Zhu, G., Jian, L., Liang, Z., & Wang, D. (2009). Stochastic spectrum access based on learning automata in cognitive radio network. In *IEEE international conference on intelligent computing and intelligent systems* (Vol. 3, pp. 294–298, 20–22). doi:10.1109/ICICISYS.2009.5358183.
12. Tuan, T. A., Tong, L. C., & Premkumar, A. B. (2010). An adaptive learning automata algorithm for channel selection in cognitive radio network. In CMC, *International conference on communications and mobile computing.* (Vol. 2, pp. 159–163, 12–14). doi:10.1109/CMC.2010.328.
13. Narendra, K. S., & Thathachar, M. A. L. (1989). *Learning automata: An introduction*. Englewood Cliffs, NJ: Prentice Hall.
14. Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and distributed computation*. Englewood Cliffs, NJ: Prentice Hall.
15. Yates, R. D. (1995). A framework for uplink power control in cellular radio systems. *IEEE Journal on Selected Areas in Communications, 13*(7), 1341–1347. doi:10.1109/49.414651.

16. Jain, R., Chiu, D., & Hawe, W. (1984). A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *Technical Research Report TR-301, Digital Equipment Corporation*.
17. Thathachar, M. A. L., & Ramakrishnan, K. R. (1984). A cooperative game of a pair of learning automata. *Automatica, 20*(6), 797–801. doi:10.1016/0005-1098.

Author Biographies

Hannaneh Bizhani received the B.S. in Computer Software Engineering from UCNA, Tabriz, Iran and her M.Sc. degree from K. N. Toosi University of Technology, Tehran, Iran, in Artificial Intelligence, Computer Engineering. Her research interests include Machine Learning, Communication Networks, and Network Protocols.

Abdorasoul Ghasemi received his B.S. degree (with honors) from Isfahan University of Technology, Isfahan, Iran and his M.Sc. and Ph.D. degrees from Amirkabir University of Technology, Tehran, Iran all in Electrical Engineering in 2001, 2003, and 2008, respectively. He is currently an Assistant Professor with the Electrical and Computer Engineering Faculty of 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.