The CSI Journal on
Computer Science and Engineering
Vol. 16, No. 1, 2018
Pages 45-59
Regular Paper

Channel allocation using Learning Automata in Cognitive Radio Networks

Panteha Ghorbani Hagh [1] Parisa Rahmani[2]

[1] Department of computer engineering Damavand branch, Islamic Azad University, Damavand, Iran
[2] Department of computer engineering Pardis branch, Islamic Azad University, Pardis, Iran

Abstract

The lack of frequency, low utilization and static allocation of spectrum have been important problems in wireless network in prior methods. To solve this problem, a concept called Cognitive Radio Network was introduced to allow the use of empty spaces of licensed spectrum. The purpose of this paper was to provide an intelligent method for detecting and allocating spectrum in cognitive radio network. In this method, Hidden Marcov model is used to predict the status of free or occupied channels, then some types of learning automata are used to allocate channel to secondary users. Also, it is a way to reduce the waiting time of users who were simultaneously requesting a channel to use a mechanism for fairness in this algorithm. The simulation results indicated that the proposed method is more effective in channel allocation to secondary users thanks to using the proposed mechanisms whose results have a greater convergence speed.

Keywords: Cognitive radio Networks, Spectrum Allocation, Learning Automata, Hidden Markov Model, Pursuit algorithms

1. Introduction

The cognitive radio is a new method for improving the usage of a very valuable natural resource, called frequency spectrum. This method, which is based on environmental learning, can gain an understanding of the environment. One of the most important aims of cognitive radio is spectrum access. Cognitive radio is an effective approach to improve spectrum use and overcoming the spectrum deficiency problem.

This approach is based on division of spectrum between primary and secondary users. Primary users have permission to use the spectrum and have access to it at any time. Secondary users, on the other hand, have access to the spectrum in a dynamic and opportunistic manner which does not cause unbearable interference with primary users. The cognitive radio system is one of the newest technologies that can overcome the problem of spectrum deficiency [1].

Since in telecommunication systems, spectrum is used more as a rare resource and considering that the wireless network environment is an environment with unknown parameters, use of learning automata that take decisions is based on the feedback received from the network environment, makes the algorithm simpler and also better respond to the network and environment changes around the agent [2].

The radio spectrum is one of the most valuable natural resources which in recent years has changed into static spectrum allocation policies and contributed to elevation in the number of wireless services to more meaningful subjects, as a response to physical spectrum deficiency issue in many bands. The Federal Communications Commission has reported evidence that the scarcity of the spectrum is not a result of heavy use of spectrum. Rather, it is because of the static frequency allocation, and hence a suitable method should be developed for better employing spectrum resources [1].

Cognitive Radio technology is aware of frequencies of its surroundings and can enhance the quality of using spectrum

by recognizing it. In order not to decrease the quality service of primary users, spectrum holes that are unused should be utilized. Awareness of spectrum and presence of primary users is the primary concern of cognitive radio systems (secondary users). For this reason, adaptive transfer to the white space is suggested without interference with the primary users. To maximize the efficiency without causing any damage to the quality of primary users, the cognitive radio system must intelligently and professionally find the useless frequency band and enter it without interfering with primary users. Further, within the shortest time during the presence of primary user, it should leave the frequency band.

A rapid diagnosis to prevent interference is one of the most important cognitive radio challenges. Previous spectrum allocation methods do not have learning algorithms, while the proposed method has novel intelligent methods for optimal use of the frequency spectrum. The purpose of this paper is to allocate channels in cognitive radio networks using intelligent methods including learning automata. In this paper, a new method is presented for assigning bands in cognitive radio network systems. Our proposed method is based on use of Markov rule to estimate the presence or non-presence of primary users and then execute the fairness index scheme for assigning free channels to the secondary users with the same priority. Finally, it involves using some kind of learning automata to allocate a band to secondary users in cognitive radio. We list the article's innovations as follows:

1. Integration of Hidden Markov Model and Learning Automata
2. The algorithm used in spectrum allocation is intelligent and responds well in dynamic environments such as cognitive radio networks (CR).
3. Learning algorithms are used to establish a fair mechanism for spectrum allocation.

The simulation results have indicated that the proposed method is effective in improving the efficiency and reducing the cost of switching channel in total throughput. Also in this paper, considering the collaboration between secondary users, the method of access to dynamic spectrum has been proposed based on pursuit learning automata. The proposed method has a significant improvement over other existing methods, but the cost of interaction between secondary users is a matter to note.

2. Related Work

In the solution presented in [3], researchers have perused the quality of Mac[1] protocols in cognitive radio networks. In this paper, MAC protocols were studied for cognitive radio networks and various parameters such as system performance scale were examined. The research concluded that this method can be used to increase the rate of throughput practically. Recently, in the strategy presented in [4], researchers have undertaken a study called strengthening instruction based on cognitive scheme to achieve an opportunistic spectrum. Cognitive radio technology enables secondary users to have access to the communication channels of primary users. As we know, the information about channel characteristics cannot be obtained with practice. So, secondary users try to have fast and independent access to discover free channels in a geographic area. This article proposes a reinforcement learning pattern which determines the order of sensing the existing channels and uses two alternative upgrade rules. Under both choices, secondary users as independent agents obtain processing information solely from their own senses for channel evaluation:

1. Occupancy probability 2. Average duration of vacancies

The ability to accurately estimate the channel characteristics without proper knowledge of the traffic pattern by the primary user is tracked in both dynamic and static transferring environments. The proposed method was compared with two channel selection schemes. The simulation indicated that the proposed scheme could succeed in determining the priority of channel selection in terms of channel characteristics, which was in terms of channel design and energy efficiency compared to previous schemes.

Recently, in the strategy presented in [5], researchers conducted a research entitled “channel allocation discovering plan using cognitive radio”. Radio channels are scarce resources utilized with the license of the judicial authorities. It tries to use channels in the best way. Studies have suggested that occupied channels can hardly be used due to many restrictions. Use of channels is based on recognition of radio derivations. This is a new heuristic method which improves the use of channels through employing the cognitive radio concept. Compared to new contact services, priority is given to old services. The simulations have indicated that the effect of channel usage manifests itself as blocked and abandoned services. In this paper, an exploratory method is introduced which uses the cognitive radio concept in channel allocation in cellular systems. To this end, the services are divided into two categories: primary services and secondary services. Primary services are used by primary users while secondary services are employed by secondary users. Secondary users are opportunistic users who use primary services of shared channels when they are free. The exploratory model uses the cognitive radio concept to reduce the speed of blocking and dropping the initial services. Meanwhile, when the channels are free, they can work well for secondary users, and thus secondary users respond better. In this regard, empirical experiments have shown better use of the channel.

In the solution presented in [6], researchers conducted a research entitled “cognitive Mac protocol in multiple channels of wireless networks”. In this section, spectrum allocation protocols based on time slots have been investigated in cognitive radio networks.

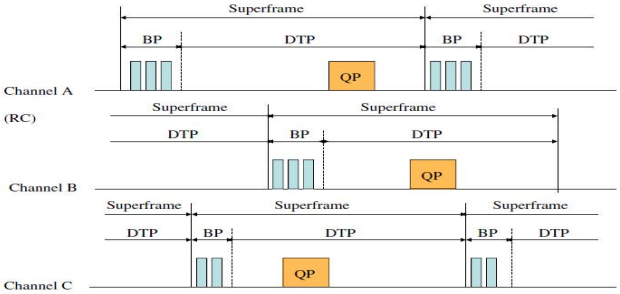

Figure1. Multi-channel super frame structure in C-MAC [6]

Multi-channel super frame structure in C-MAC: Each channel is constructed in the form of super frames whose

[1] Media Access Control

Beacon Periods (BP)[2] are located in the direction of separate channels (without overlapping).

The limitation of RC[3] usage in the distributed slot protocol presented in [7] has been resolved. It not only generates beacon periods and data transmissions, but also provides an in-band signal using dedicated control windows. Throughout this window, bridge nodes are allowed to use multiple channels to optimize the protocol performance by gaining access to more than one compatible group in each super frame.

In the solution presented in [8], researchers performed a research called "cognitive radio based on learning automata in clustered wireless ad hoc networks". In current wireless networks, radio systems are regulated by a static allocation strategy of spectrum. This allocation policy, which is exclusively dedicated to a particular user, leads to undesirable conditions, as some systems use only a small fraction of the allocated spectrum, while others duffer from a very serious lack of spectrum. This paper presents a dynamic-frame based on TDMA according to learning automata for slots assigned in a cluster of ad-hoc wireless networks with unknown traffic parameters, in which intra-cluster communications are scheduled with TDMA pattern. and CDMA scheme covers TDMA scheme to reach an intra-cluster relationship without any interference. In this method, each cluster head is responsible for assigning the slot without collision in the cluster and in the input traffic parameters of its cluster members. Then, we use traffic parameters to consider the desirable channel access schematization in the cluster. The MAC layer in each cluster is based on a TDMA programmed to allocate a portion of the TDMA frame to each host which is appropriate for its traffic load.

The simulation experiment reveals the superiority of the design of memory allocation algorithm over available methods in terms of the channel usability, control overhead and throughput, especially under explosive traffic conditions.

In this paper, we have proposed a slot allocation algorithm based on learning automata in clustered wireless ad-hoc networks when the traffic input parameters are unknown. This method is recommended for a cluster CDMA / TDMA network in which the cluster head is responsible for assigning slots without collisions. The purpose of this paper is to demonstrate the capability of learning automata as a random probability learning method for recognizing unknown traffic distribution parameters and finding an optimal memory allocation strategy.

To demonstrate the superiority of the proposed channel assignment scheme to the existing methods, we made two sets of simulation experiments. Firstly, we compared the proposed method with two well known CDMA / TDMA designs called CS-DCA and DCA hybrid. Then in the second set, we compared it with two slot assignment schemes with dynamic frame-length called DTSA and DFLCA. The results showed that our proposed method has been better than others in most cases.

In the solution presented in [9], researchers conducted a research entitled "Control of permission to enter and select channels based on multi-response learning automata (MRLA)[4] in cognitive radio networks".

We use multi-response learning automata to control how secondary users should gain access to the main channels licensed in cognitive radio networks. We seek two goals in this article:

1- Estimating the probability of each primary channel availability

2- Controlling the entry of secondary users in order to reduce the collision rate between them.

We consider single-user and secondary multi-user scenarios. In the first scenario, the secondary user dispatches the learning automata to estimate the probability of the availability of the main channel for efficient utilization. In the second scenario, to control the collision rate, each secondary user sends a multi-response learning automata algorithm to estimate the primary traffic. Then, to better control the secondary collision rate, when the number of secondary users is greater than that of the main channels, we recommend an input control scheme. In this scheme, some secondary users are blocked at some time and have no interactions with the environment. The convergence of the proposed algorithm is analyzed with or without the input permission scheme. Simulation results are provided to demonstrate total throughput improvement of secondary users and costs of switch, as long as the fairness index scheme is supported between them.

In the solution presented in [10], researchers conducted a research called “Optimizing Channel Selection for Cognitive Radio Networks using a Distributed Bayesian Learning Automata-based Approach”.

Consider a multi-channel CRN[5] with multiple Primary Users (PUs), and with multiple Secondary Users (SUs) competing for the access to the channels. In this scenario, it is essential for SUs to avoid collision among one another while keeping an efficient usage of the available transmission opportunities. We investigate two channel access schemes. In the first model, an SU selects a channel and sends a packet directly without Carrier Sensing (CS) whenever the PU is absent on this channel. In the second model, an SU invokes CS in order to avoid collision among co-channel SUs. For each model, we analyze the channel selection problem and prove that it is a so-called “Exact Potential” game. We also formally state the relationship between the global optimal point and the Nash Equilibrium (NE) point as far as system capacity is concerned. Thereafter, to facilitate the SU to select a proper channel in the game in a distributed manner, we design a Bayesian Learning Automaton (BLA)-based approach. Unlike many other Learning Automata (LA), a key advantage of the BLA is that it is learning parameter free. The performance of the BLA-based approach is evaluated through rigorous simulations and this has been compared with the competing LA-based solution reported for this application, whence we confirm the superiority of our BLA approach.

Recently, in the strategy presented in [11], researchers performed a research with the title “A learning automata

[2] Beacon Period
[3] Rendezvous Channel
[4] Multi Response Learning Automata
[5] Cognitive Radio Network

based spectrum prediction technique for cognitive radio networks."

This paper introduces an application of artificial intelligence in the cognitive radio networks. The Cognitive Radio Network (CRN) provides a suitable environment for Secondary Users (SUs) to share the spectrum with Primary Users (PUs) in a non-interfering manner. In order to determine the availability of PUs bandwidth, SU can sense the spectrum in the channel. But, accurate and constant spectrum sensing consumes the energy of the SUs significantly. In these conditions, to discover the spectrum holes in the absence of PUs, predictive techniques can be one of the solutions which can reduce the consuming energy of the SUs. The simplicity and reliability of predictive techniques play an important role in the practice. In this paper, we utilize a Learning Automata technique to predict the spectrum hole in the cognitive network based on the statistical behavior of the PUs. Simple structure and acceptable prediction rate are two important features of the proposed technique. In order to compare the performance of the proposed method with similar predictive techniques in CRNs, we design a predictor model using multilayer perceptron artificial neural networks and test the performance of these two methods on the same conditions. The results of modeling confirm that the Learning Automata with simple structure is more reliable than neural network.

In [12], the authors explored the scenario where multiple SUs competed for channel access in multiple channels. In that work, as the number of SUs in the system is assumed to be larger than the number of channels, CS was utilized, by default, in order to avoid co-channel collision among SUs. Based on the system configuration, the channel selection problem was formulated as a game. Thereafter, more mathematical insights were provided from the aspects of both the game itself and its potential solutions.

Although the analytical and simulation results in [12] had shown the efficiency of LA in solving problems of this kind in CRNs, there were a few unresolved issues by which the performance could be potentially improved:

1. To allow the SU communication pairs to converge in a distributed manner, the LR−I scheme was utilized to play the game, which, in turn, requires a learning parameter to be configured in advance.

As the applicability and efficiency of the learning-parameter-free LA, i.e., the BLA, in game playing were earlier demonstrated for solving the Goore game [13], we were motivated to incorporate the BLA to solve the multi-SU scenario in CRNs, with the ultimate hope that the system's overall performance could be further improved by its inclusion.

2. As mentioned earlier, if the number of channels is greater than the number of SUs, a scheme that did not invoke CS is an interesting option. This option could be considered with the hope that the learning process can successfully resolve the potential collisions among SUs.

3. The CS process in [12] was assumed to be ideal, meaning that a single SU will certainly win the competition among multiple co-channel SUs. In other words, the event of collision among co-channel SUs, which, indeed, exists in reality, was ignored. To model the impact of the collision between potential co-channel SUs, we foresee the need for a more precise function that can describe the CS process. This is because a different model of the CS process will result in a distinct utility function for the game. Consequently, the property of the game under the new model, begs investigation. Based on the above observations in the state-of-the-art, we are motivated here to investigate the above unresolved issues, and to propose BLA-based distributed approaches to solve the multi-user multi-channel problem in CRNs, and expect to contribute to the state-of-the-art. In the following sections, we will detail the system configurations, analyze the various problems encountered, design the algorithms, and evaluate their performances by rigorous simulations.

In the solution presented in [14], researchers have undertaken a study called "Learning automata based multipath multicasting in cognitive radio networks". Cognitive radio networks (CRNs) have emerged as a promising solution to the problem of spectrum under utilization and artificial radio spectrum scarcity. The paradigm of dynamic spectrum access allows a secondary network comprising of secondary users (SUs) to coexist with a primary network comprising of licensed primary users (PUs) subject to the condition that SUs do not cause any interference to the primary network. Since it is necessary for SUs to avoid any interference to the primary network, PU activity precludes attempts of SUs to access the licensed spectrum and forces frequent channel switching for SUs. This dynamic nature of CRNs, coupled with the possibility that an SU may not share a common channel with all its neighbors, makes the task of multicast routing especially challenging. In this work, we have proposed a novel multipath on-demand multicast routing protocol for CRNs. The approach of multipath routing, although commonly used in unicast routing, has not been explored for multicasting earlier. Motivated by the fact that CRNs have highly dynamic conditions, whose parameters are often unknown, the multicast routing problem is modeled in the reinforcement learning based framework of learning automata. Simulation results demonstrate that the approach of multipath multicasting is feasible, with our proposed protocol showing a superior performance to a baseline state-of-the-art CRN multicasting protocol.

In [15], the authors discussed the issue of determining the circumstances under which a new round of learning had to be triggered in CRNs, and the learning-parameter-based DGPA was again adopted as the learning scheme in the SU pairs.

Other works in this field are listed in References 38 to 42.

3. Learning automata

Learning automata can be considered as an abstract object with a finite number of actions. The learning automata act by choosing an action from their set of operations and applying it to the environment. The operation is evaluated by a random environment where the automata use the response of the environment to select its next action. During this process, the automata learn to select the optimal operation. The method of using environment's response to the selected action of automata which is used to select the next action of automata is employed by learning automata algorithm. In other words, learning automaton is an abstract model which randomly chooses an action from its finite set of operations and applies it to the environment. The environment evaluates the selected action of automata and announces its evaluation

result by an amplifier signal to the learning automata, so that the automata select the next action [16].

Figure 2. The relationship between learning automata and environment

Learning automata consist of two main parts:

A: A random automata with limited numbers of operations and a random environment with which the automaton is associated.
B: Learning automata which learn the optimal operation using the learning algorithm.
A random automata can be considered as a finite machine, which is defined as five dimensional $SA \equiv \{\alpha, \beta, F, G, \phi\}$ where represents the number of automata acts، $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ shows the set of automata acts ، $\beta \equiv \{\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \ldots, \boldsymbol{\beta_r}\}$ denotes the automata input set، $F \equiv \varphi \times \beta \rightarrow \varphi$ the new state generation function (a function that maps the input and current state to the next mode) ، $G \equiv \boldsymbol{\varphi \rightarrow \alpha}$ shows the exit function which maps the active status to the next output and $\varphi(n) \equiv \{\varphi_1, \varphi_2, \ldots, \varphi_k\}$ denotes the set of automata internal state at the moment n. The set (α) contains the outputs of automata, in which automata select an action from the r actions of this set to act upon the environment in each step. The input set (β) identifies the automata entries. Functions F and G map the current input state to the next output (the next action) of automata. If the mappings F and G are definite, the automaton is called deterministic automaton. On the other hand, when F and G mappings are random, the automaton is called random automaton.
In this case, only the probabilities for the next state and the corresponding outputs can be determined. The random automata are divided into two-dimensional automata with a fixed structure and automata with a variable structure.
In the first type, the probabilities for the various operations are constant, but in the second type, the probabilities are updated at each time. The possible environment can be expressed mathematically with triplets E = {α, β, c} Where $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents the set of environment inputs، $\beta = \{\beta_1, \beta_2, \ldots, \beta_q\}$ shows the set of environment outputs, and $c = \{c_1, c_2, \ldots, c_q\}$ constitutes the set of penalty probabilities of random automata output actions. If B is a two-member set, the type of environment is P. In such an environment, $B_1 = 1$ is considered as a penalty while $B_2 = 0$ is considered as a reward. In a Q-type environment, the set B has a finite number of members, and in a S-type environment B has an infinite number of members. C_1 is the penalizing probability of a_1 action.
The internal state of automata φ (n) at the moment n is shown with the probability vector of automata operations P (n) according to Eq.(1), Eq.(2).

$$p(n) \equiv \{p_1(n), p_2(n), \ldots, p_r(n),\} \quad (1)$$

So that

$$\sum_{i=1}^{r} p_i(n) = 1 \,, \forall n \,, p_i(n) = prob[\alpha(n) = a_i] \quad (2)$$

At the beginning of automata activity, the probability of its operations is equal to $\frac{1}{r}$ (where r is the number of automata) [17].
The main idea of all learning algorithms is as follows:
If the learning automata in the nth repeat choose an action such as α_i and receive an optimal response from the environment, $p_i(n)$ (the probability of action (α_i) grows, while the probability of other actions drops. Conversely, if the response from the environment is undesirable, the probability of α_i is reduced while the probability of other automata operations increases. However, the changes are made in such a way that the sum of $p_i(n)$ is always stable and equal to 1.
There are some standard and estimation algorithms that are described below: [18], [43-45], [46-47], [49]

3.1. L_{R-I} : The (SL_{R-I}) reinforcement learning algorithm updates the action probability vector in accordance with Eq. (3). α is a learning parameter and $0 \leq \alpha \leq 1$.

$$\boldsymbol{p_i(n+1) = p_i(n) + a\big(1 - \beta_i(n)\big)\big(1 - p_i(n)\big)} \quad (3)$$
$$\boldsymbol{p_j(n+1) = p_j(n) - a\big(1 - \beta_i(n)\big)p_j(n) \;\; \forall j, j \neq i}$$

3.2. SL_{R-P} : The (SL_ (R-P)) reinforcement Learning Algorithm updates the probability vector of automata operations with Eq. (4). If r is the number of actions in repetition n, α_i is the selected action and β_i is the environment response.

$$\boldsymbol{p_i(n+1) = p_i(n) + a\big(1 - \beta_i(n)\big)\big(1 - p_i(n)\big) - a.\beta_i(n).p_i(n)}$$
$$\mathbf{p_j(n+1) = p_j(n) - a\big(1 - \beta_i(n)\big)p_j(n) + a.\beta_i(n).\left[\frac{1}{r-1} - p_j(n)\right] -}$$
$$\mathbf{a.\big(1 - \beta_i(n)\big).p_j(n) \quad \forall j, j \neq i} \quad (4)$$

3.3. $L_{R\in P}$: The learning automaton $SL_{R\in P}$ with the number of actions, r, the reward parameter, a, and the penalty parameter, b, updates the probability vector using Eq. (5).

$$\boldsymbol{p_i(n+1) = p_i(n) + a\big(1 - \beta_i(n)\big)\big(1 - p_i(n)\big) - b.\beta_i(n).p_i(n)}$$
$$\boldsymbol{p_j(n+1) = p_j(n) - a\big(1 - \beta_i(n)\big)p_j(n) + b.\beta_i(n).\left[\frac{1}{r-1} - p_j(n)\right] -}$$
$$\boldsymbol{a.\big(1 - \beta_i(n)\big).p_j(n) \quad \forall j, j \neq i} \quad (5)$$

3.4. Introduction of Estimation Algorithms

In order to increase the speed of convergence in learning algorithms, a new class of algorithms was introduced that were called estimation algorithms. The main feature of these algorithms is that the probability of reward is maintained for each operation, which is used to update the probability vector. The method is that in each repeat cycle, the learning automaton chooses an action based on its reward. Then the environment supposes a response for this action. Based on this response, the estimation algorithm updates the reward probability of that action. Therefore, the change in the probability vector is based on the received feedback of the environment and the reward estimation of actions. Descriptions of the detail of estimation algorithms are in- [46-49].
If the updating scheme in learning automaton is the same as Eq. (6), T is the update plan, α (n) is the selected action and β (n) is the response that is received from the environment.

$$\boldsymbol{p(n+1) = T\big(p(n), \alpha(n), \beta(n)\big)} \quad (6)$$

In learning automaton with a pursuit design, updating the probability vector is performed as Eq. (7).

$$\boldsymbol{Q(n+1) = T\big(Q(n), \alpha(n), \beta(n)\big)} \quad (7)$$

Q (n) is the pair of <P (n), d (n)>, where dˆ(n) is the vector of reward probability estimation.

3.4.1. $(\mathbf{CP_{R-p}})^{6}$: This learning algorithm uses a reward-penalty pattern. That is, when the environment gives a reward or penalty to the selected action, the probability vector of the action is updated. The first step in the algorithm is to select the action α (n) based on the probability distribution of the probability vector p (n). So, when the automaton receives a reward or penalty from the environment, it increases the probability of the action that its corresponding reward estimation is maximum. Then, it reduces the probability of the other actions.

According to formula (8), the vector of reward estimation is shown with dˆ(n). This vector consists of two vectors w (n) and z (n). Z (n) is the number of times that the operation of α (n) has been chosen and W (n) is the number of times that has been rewarded to α (n). The reward estimation vector for action i is calculated using Eq. (8).

$$\widehat{d}_i(n+1) = \frac{W_i(n+1)}{Z_i(n+1)} \quad (8)$$

Updating the probability vector is based on Eq. (9).

$$p(n+1) = (1-\lambda)p(n) + \lambda e_m \quad (9)$$

Where e_m is the unit vector [0, 0, 0, ..., 1, ..., 0,0,0]. In this vector the position 1 belongs to the action that has the maximum reward estimation. Formula (9) shows that the probability vector p(n), moves to the action with the maximum current reward estimation. The algorithm CP_{R-P} is similar to the algorithm L_{R-P}. The only difference is that the L_{R-P} algorithm moves the probability vector in the direction of the action that is received the most recent reward, but the algorithmCP_{R-P}, moves P(n) in the direction of the action with maximum reward estimation.

3.4.2. $(\mathbf{DP_{R-I}})^{7}$: A discretized version of a pursuit algorithm is presented in [18]. This algorithm is based on L_{R-I} algorithm. This means that the probability vector p (n) is updated when the reward is received and will not be executed in the event of penalties. The differences between the discrete and CP_{R-P} occur only in the updating rules for the action probabilities. This algorithm makes changes to the probability vector in discrete steps. So, when an action is rewarded, all the actions that do not correspond to the highest estimate are decreased by a step Δ. In order to keep the sum of the components of the vector p(n) equal to unity , the probability of the action with the highest estimate has to be increased by a coefficient. Updating the action of probability vector is performed as Eq. (10).

$$\boldsymbol{p_j(n+1) = \max_{j \neq m}\{p_j - \Delta, 0\}}$$
$$\boldsymbol{p_m(n+1) = 1 - \sum_{j \neq m} p_j(n+1)} \quad (10)$$

By combining different learning algorithms with pursuit principles, we can introduce four categories of algorithms: DP_{R-I} , DP_{R-P} [8], CP_{R-I} [9], CP_{R-p}

3.4.3. pursuit algorithms (PST)[10]

In algorithms such as L_{R-I} algorithm, updating the probability of operation at a constant point depends only on the selected operation and the gain obtained, while it is independent of the history of past operations and reinforcements. However, pursuit algorithms are a set of learning algorithms performing operation probability updating based on the history of operations and past reinforcements. These algorithms estimate the characteristics of the random environment in which FALA[11] operates. These estimates are calculated online during FALA operations. The main idea of this method is to use these estimates to improve the performance in terms of speed and accuracy of learning. One type of pursuit algorithms is BPST[12], which we will discuss below.

3.4.3.1. Bayesian pursuit algorithm (BPST)

This new method was introduced by Zhang and Oommen in 2013 [19], [20]. As described in the previous section, the fastest algorithms for learning automata are pursuit algorithms. These algorithms use the maximum likelihood similarity to follow the desirable action. The BPST algorithm combines the basic principles of two learning automata algorithms including BLA [13] and PST. In this algorithm, estimates are based on BLA model and the nature of variables like Bayesian. Unlike the ML[14] estimation, which is usually a single value, these estimates are calculated as Bayesian. This algorithm allows selecting the last spectrum values to be used for more accurate estimation by a subsequent distribution. On the other hand, the PST model is utilized for extraction purposes. In the previous section, the PST algorithm was fully described. In this section, we give a brief overview of the BLA algorithm.

3.4.3.2. Bayesian learning automata (BLA)

This automaton is based on Bayesian argument. In BLA, there is a beta distribution in accordance with Eq. (11). In this algorithm, the beta distribution is used in two ways: firstly, creation of Bayesian estimation for appraising the probability of rewards of corresponding actions and secondly, the statistical arrangement based on the random selection mechanism.

$$f(x; a, b) = \frac{x^{a-1}(1-x)^{b-1}}{\int_0^1 u^{a-1}(1-u)^{b-1}\,du}\ ,\ x \in [0,1] \quad (11)$$

Where, a and b represent two positive parameters and f is a probability density function.

[6] Continuous Pursuit reward-penalty Automaton
[7] Discretized Pursuit Reward-Inaction Algorithm
[8] Discretized Pursuit reward-Penalty Automaton
[9] Continuous Pursuit reward-Inaction Automaton
[10] Pursuit Algorithm
[11] Finite Action Learning Automata
[12] Bayesian Pursuit Algorithm
[13] Bayesian Learning Automata
[14] Maximum Likelihood

The BPST algorithm considers the probability of rewards of corresponding actions based on beta distribution. This estimate helps the reward probability of action i, which is shown with xi, to be correctly calculated using Eq. (12).

$$\frac{\int_0^{xi} v^{a-1}(1-v)^{b-1}}{\int_0^1 u^{a-1}(1-u)^{b-1}\,du} = 0.95 \qquad [19], [20] \qquad (12)$$

Where, a represents the number of rewards received from the environment by the selected action and b shows the number of failure. Also, Xi is the reward probability of action i. The value of 0.95% means that 0.95% of the last values have proven to follow the best practice. Using this formula, the reward of action i is estimated. After estimating the reward of actions in automata, updating the vector probability is exactly the same as PST algorithm. It means that it follows an action whose reward estimation is maximum.

4. The proposed method

The purpose of this paper is to offer a method for detecting spectrum holes and allocating spectrum to secondary users with regard to PU[15] activity. Indeed, the proposed method identifies the times when a primary user is present in the spectrum. For this purpose, the primary user's behavior is first identified and predicted using Hidden Markov Model. Then, the proposed scheme uses this predicted parameter to perform the spectrum allocation algorithm by learning automata. The advantage of using learning methods to detect the primary user activity and spectrum allocation is that in dynamic environments such as cognitive radio where the behavior of network is constantly changing, the prediction of the primary users' behavior and learning of how the spectrum is used by primary and secondary users will lead to performance improvement of cognitive radio networks. Accordingly, the scenario of this paper involves three parts: 1. detecting the presence of a primary user or PU in the channel using Markov rule 2. reviewing the simultaneous requests of secondary users or SU using the plan of fairness index 3. allocating frequency spectrum by BPST automata.
Regarding how spectrum allocation takes place, the spectrum availability in a region using BPST learning automata is learned by a node. Then, this learning improves and leads to enhanced performance of the frequency spectrum allocation algorithm over time. Meanwhile, use of learning automata in spectrum allocation and identification of licensed spectrum result in diminished switching between channels, bearing in mind that frequent switching between existing channels develops many problems such as increased delays, loss of packet, and increased communication costs over time. This is especially true in the environments that are very dynamic as cognitive radio networks, resulting in a significant reduction in network performance and undesirable use of the licensed spectrum. However, as mentioned previously, in algorithms such as L_{R-I}, the probability of an operation being updated at a fixed point depend only on the chosen action and is independent of the history of past operations and enhancements. So, BPST algorithms are used to improve the spectrum allocation function. In this category of learning methods, the probability of channels is updated based on the history of actions. The main reason for using this method is to improve performance from the perspective of learning speed and accuracy. Therefore, in the next section, spectrum allocation operations are performed using pursuit Bayesian algorithms, which we describe here. In each secondary user node, a learning automaton is included. We assume that the entire authorized spectrum is divided into a series of identical widths of channels. That is, the number of channels is specified by default $[ch_1, ch_2, \dots, ch_m]$. The algorithm works in the same way as the beginning. Also, the probability of selecting each channel for secondary users is the same. In each repetition, the cognitive radio selects a channel from the list of available channels based on the probability distribution of that action. Now, with the feedback received from the environment, probability vector updates the channels.
The operation of the previous method, MRLA, is compared with the proposed method of the article through simulation. Also another mechanism is presented in the proposed strategy to improve the use of spectrum and execute the fairness in the competition between secondary users. When multiple secondary users simultaneously want to use a frequency band, priority is given to the secondary user who has requested a greater range of spectrum and could not obtain the channel according to its request. In this case, the amount of traffic increases in its buffer. Indeed, the proposed method gives the channel access priority to the node which has waited more than the others.
The methodology of the algorithm in the flowchart is presented in detail in the appendix.
The steps of the proposed algorithm are summarized as follows:

4.1. PU activity prediction

Here, we introduce the learning automata model to predict PU activity in the corresponding frequency bands. In other words, the bandwidth usage scheme of PU is modeled. The proposed learning automata structure and PUs' behavior modeling are described in three steps:

- Generating the Pus' activity pattern
- Modeling training and updating
- Testing the ability of the model to predict the correct behavior of the system

The first step in modeling the system is to provide a method for displaying the activity of PUs. In practice, the PU activity pattern becomes available by sensing the exact spectrum and the collection of statistical behavior of PUs for a certain period of time. However, in order to test the ability of the model to predict the correct state of PUs, the PUs' activity pattern is generated based on the characteristics of PUs' activity.
The PU spectrum usage pattern is a function of time and space. Each PU has a different activity pattern, according to its geographic location and demand range, which varies with time. In order to display the activity of PUs, one can use a dual arrangement in which 0 indicates absenteeism while 1 shows the presence of the user. The number of bits in each sequence represents the accuracy of the spectrum sensitivity for PU activity each day. The ratio of the number of 1 to the total number of bits in each order is considered as a PU activity factor, which is one of the most important parameters

[15] Primary User

in system modeling. To model the obtained pattern of PU bandwidth, it is assumed that each PU occupies the bandwidth more than half of the total time every day. To challenge the PU status prediction model, a random behavior is added to each sequence (Figure 3). In each sequence, the number of bits is randomly assigned to zero or one while PU is constantly active [11].

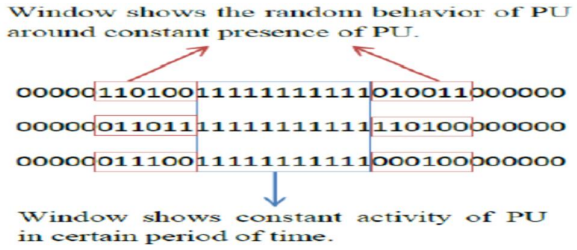

Figure3. Binary sequence showing the random behavior of PUs [11]

4.2. Training the behavior modeling of PU

We can use Markov chain to model the PU behavior. Figure 3 shows the number of bits in the states،total number of states and the complexity of the structure in Markov model. Each bit represents a PU activity in a state. For example, 001 indicates that the PU is idle for two consecutive time slots and is activated in the last slot. In Markov model, the following sets are defined:

A = {a_1, a_2, ..., a_r} is the group of operations on Markov chains; r represents the number of operations (2 <r <∞); and the choice of action at time t is a (t), which can be 0 or 1. Each action is a transition on Markov states.

E = {e_1, e_2, ..., e_r} is defined as the set of possible responses of the environment. e_i = 1 is the PU activity and e_i = 0 is the PU absence in the respective range.

B={$b_1(t)$,$b_2(t)$,…,$b_r(t)$} shows the set of transition probabilities between the modes ($b_i(t)$∈[0,1]) and $0 \leq \sum_{i=1}^{r} b_i(t) \leq r$ [21].

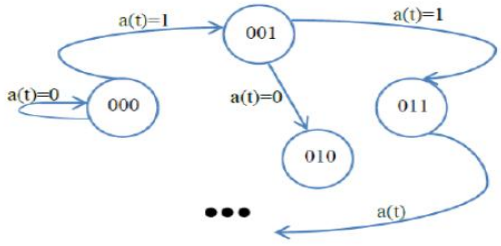

Figure4. An example of Markov chain [11]

Before the training, the probabilities of primary transition are considered as 0.5. This assumption shows that at the start of the training process, no action has priority over other actions. Eq. (13) and Eq. (14) represent the probability updating scheme for the training model. The sum of the output probabilities in each step is 1.

$$\left.\begin{aligned} b_{e1}(t+1) &= b_{e1}(t) + \tfrac{1}{k} \quad if \;\; e(t+1)=1 \\ b_{e0}(t+1) &= b_{e0}(t) - \tfrac{1}{k} \quad if \;\; e(t+1)=1 \end{aligned}\right\} \qquad (13)$$

$$\left.\begin{aligned} b_{e1}(t+1) &= b_{e1}(t) - \tfrac{1}{k} \quad if \;\; e(t+1)=1 \\ b_{e0}(t+1) &= b_{e0}(t) + \tfrac{1}{k} \quad if \;\; e(t+1)=1 \end{aligned}\right\} \qquad (14)$$

Here, k is an integer value, which is the number of bits in each step. Choosing different values for the parameter k affects the threshold value on decision-making (Figure 4).

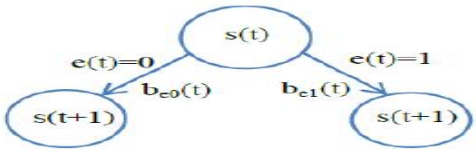

Figure 5. An example of updating scheme at time slot t [11]

In Figure 5, s (t) is the current state of the model. Based on binary sequence of PU activity, the next environment response (PU activity) is known. If PU is active for the next time slot e (t + 1) = 1, then transition probability b (t + 1) is updated using Eq. (13). b_{e0} shows transition probability when PU is inactive (e (t + 1) = 0) and b_{e1} is used when the response of environment is 1(e (t + 1) = 1). If PU is inactive for the next time slot, transition probability is updated using Eq. (14). After calculating b (t + 1), the current state changes to s (t + 1) and updating scheme will continue for the next time slot. This process continues for all set of binary sequence of the PU activity to complete the training phase [17].

One of the most important parameters in training phase is the complexity of learning method. Using a simple low computational method makes a prediction model more executable for online applications. Update methods in most prediction techniques is a function of the number of O (r) operations. However, the proposed updating pattern in Formulas 13 and 14 is independent of the number of actions in the update phase [11].

After training the prediction model, in order to estimate the ability of the model to accurately predict the PU activity, a binary sequence test is produced by using the same method described to generate the PU activity pattern. The random time at the test sequence determines the current state of the model. The next step in PU activity is to predict by comparing the transition probability in each mode. Later, the result of the prediction is compared with the actual PU activity in the test sequence. This process is performed for a large number of repetitions and using Eq. (15), the prediction rate is calculated to predict the correct PU activity [11].

$$\text{Forecast rate} = \frac{\textbf{The number of correct predictions}}{\textbf{Total number of repetitions}} * 100(\%) \qquad (15)$$

4.3. Estimating the magnitude of back off for competition between SUs

$$U(ch_i) = \begin{cases} 1 & if \; ch_i \notin \{ ch_i \mid e_{ij} \in U \} \\ 0 & otherwise \end{cases} \qquad (16)$$

According to Eq. (16), when multiple cognitive users (secondary users) compete at a moment to access the channel, then the node that has received the channel earlier can send its data. The proposed method has offered a mechanism that allows cognitive users who have more traffic in the queue or those who have failed to access the channel gain access to the data channel faster than their competitors.

For this purpose, in each node, the automata with two actions {0 and 1} are positioned such that the primary value of the probability of each action is initially 0.5.

Action 1 shows the probability of obtaining the channel by CR[16]. If CR submits the channel request and the channel was free according to Eq. (17), then the probability of action 1 would be reduced as CR could use the channel and send its data. However, if the channel was not free and was occupied according to Eq. (18), then the probability value grows. Now, if multiple CRs request a data transfer at the same time, they will wait as much as possible given the probability value of action 1 per CR. Then, they submit their request to send data, and thus the cognitive radios whose probabilities are higher (i.e. CRs cannot access the channel for a long time) will take the data channel.

Through this technique among cognitive users to use the channel, fairness can be established in access to the channel, making fairness be observed to access the authorized channels between secondary users.

$$p_{Ti}(t+1) = p_{Ti}(t) - \lambda.(p_{Ti} - \alpha) \qquad if\ the\ channel\ is\ free \qquad (17)$$

$$p_{Ti}(t+1) = p_{Ti}(t) + \lambda.(1 - p_{Ti}) \qquad if\ the\ channel\ is\ busy \qquad (18)$$

λ is the size of learning process. This algorithm runs at time periods that a node requests. The primary value of $p_{Tri}(t) = 0.5$. The parameter α ε [0, 1]. This algorithm ensures that the value of p_{Tri} is always within the range of [α, 1]. The reasons is that if a node does not have a packet for sending, then p_{Tri} may reach zero and lose its chance to get the channel; therefore, the value of α is considered to be a number greater than zero, so that the node i has the chance for not losing the channel.

4.4. Spectrum allocation in the cognitive radio network

Once the proposed method predicted the behavior of primary users by the Hidden Markov model in the previous section, it uses the Bayesian pursuit algorithm to allocate the spectrum in this section. This learning algorithm uses the history of selected operations and feedback received to calculate the probability of operating rewards. In this algorithm, from the channel list, the channel is selected based on its probability distribution, and then its reward is estimated. Note that in the reward estimation from the predicted model of PU behavior, Markov model is used to allocate the spectrum to users more accurately.

At time t, the algorithm calculates the current estimation of the reward probabilities, $\hat{x}_i(t)$. In order to calculate $\hat{x}_i(t)$, with considering the Bayesian nature of pair distribution values $a_i(t), b_i(t)$, which are positive parameters of the beta distribution, are updated based on the received response from the environment in accordance with Eq. (19).

$$\begin{cases} a_i(t) = a_i(t-1) + 1;\ b_i(t) = b_i(t-1) \\ a_i(t) = a_i(t-1);\ b_i(t) = b_i(t-1) + 1 \end{cases} \qquad (19)$$

The probability of the reward of chosen action α_i, which is equal to $x_i(t)$, is calculated according to Eq. (20).

$$\frac{\int_0^{x_i(t)} v^{a_i-1}(1-v)^{b_i-1}}{\int_0^1 u^{a_i-1}(1-u)^{b_i-1}du} = 0.95 \qquad (20)$$

Finally, we update the probability of the action with the highest reward, according to Eq. (21):

$$p(t) = (1-\lambda)p(t-1) + \lambda e_m \quad \text{With} \quad m = \arg\ max_i[x_i(t)] \qquad (21)$$

Note that updating the probability of operations in the Bayesian algorithm does not depend on identifying the selected channel. In this algorithm, the chosen channel and the gain obtained are used to update the estimation of the reward probabilities. Then, the probability of the channels is updated to increase the probability of the best estimated current channel. The other advantage of the Bayesian algorithm is that the updating of P (t) is not directly related to the environment response. Also, the learning algorithm is faster than the other learning algorithms such as L_{R-I}. In this algorithm, if λ is sufficiently small, each channel is allocated to an enough number and the estimations are closed to the actual values of the reward probabilities.

4.5. Spectrum allocation algorithm using a Bayesian Algorithm

In the first step of the algorithm, which is initialization step, assumes that the number of channels is r, and the probability of selecting all channels is equal to 1 / r. In steps 2 and 3, the reward probability of each channel (x_i) is estimated using Bayesian algorithm. In fact, steps 2 and 3 compute the reward estimation $x_i(t)$. In these two steps, a and b are reward and penalty parameters for channel i. Steps 2 and 3 are repeated so that a suitable value for x_i (t) is obtained. After $x_i(t)$ (reward estimation of channel at time t) is calculated, in step 4, the probability of the channels is updated according to the estimation of rewards that is obtained in the previous steps, and finally, the probability of selecting a better channel increases.

The details of the BPST algorithm are as follows:

Initialization:

1- Initialize the parameters

- Suppose the set of automata actions in each secondary user is as follows, where r is the number of channels.

$\alpha_i = \{ch_1, ch_2, \dots, ch_r\}$

- Set the probability of each action as follows. At the beginning of the process, the probability of all actions is the same according to Eq. (22).

$$p(t)_j = \frac{1}{r} \quad \text{where } j = 1..r \qquad (22)$$

Repeat steps 2 to 4 to converge to the appropriate solution.

2- Repeat the following steps as frequent as constant r.

- Choose a channel ch_i from the set of actions α_i based on its probability distribution.
- Calculate the feedback received from the environment as below.

✓ if ch_i is busy at the moment and $b_{e1}(t+1) > 0.5$ (in other words, the probability that the next slot will be occupied is more than half of the probability), then the value of $b_i(t)$ is calculated via Eq.(23).

$$b_i(t) = b_i(t-1) + 1 \qquad (23)$$

[16] Cognitive User

✓ If ch_i is occupied at the moment and $b_{e1}(t+1) < 0.5$, (in other words, the probability that the next slot will be occupied is less than half of the probability), then the value of $b_i(t)$ is calculated via Eq.(24).

$$b_i(t) = b_i(t-1) + 0.5 \quad (24)$$

✓ If ch_i is free at the moment and $b_{e0}(t+1) < 0.5$ (in other words, the probability that the next slot will be free less is than half of the probability), then the value of $a_i(t)$ is calculated using Eq.(25):

$$a_i(t) = a_i(t-1) + 0.5 \quad (25)$$

✓ If ch_i is free at the moment and $b_{e0}(t+1) > 0.5$ (in other words, the probability that the next slot will be free is more than half of the probability), then the value of $a_i(t)$ is calculated using Eq. (26).

$$a_i(t) = a_i(t-1) + 1 \quad (26)$$

3- Now estimate the reward of the channel i , $x_i(t)$ with Eq. (27).

$$\frac{\int_0^{xi} v^{a-1}(1-v)^{b-1}}{\int_0^1 u^{a-1}(1-u)^{b-1}\,du} = 0.95 \quad (27)$$

4- Then update the probability vector according to Eq. (28).

$$p(t) = (1-\lambda)p(t-1) + \lambda e_m \quad \text{With } m = \arg\ max_i[x_i(t)] \quad (28)$$

Where, λ represents the learning parameter (or step size) within the range of $0 < \lambda < 1$.

5- If the node i requests the channel, the probability is updated via Eq. (29).

$$\begin{cases} p_{Ti}(t+1) = p_{Ti}(t) - \lambda.\left(p_{Ti} - \alpha\right) & \text{if the channel is free} \\ p_{Ti}(t+1) = p_{Ti}(t) + \lambda.\left(1 - p_{Ti}\right) & \text{if the channel is busy} \end{cases} \quad (29)$$

End of the algorithm

5. Simulation environment

In the evaluated network, the number of channels ranges from ch1 to ch10 while the number of nodes of the simulation environment lies within the range of 10 to 50. It consists of a primary user (PU) and secondary users (SUs). These nodes are distributed randomly and uniformly in a two dimensional simulation of 800 by 800 square meters. The buffer of each node is unlimited and the transmission range of each node is 200 meters. The data transfer rate or CBR[17] is 2 Mbps . The speed of nodes is 0 to 5 m/s and the number of CBRs between the source and destination is randomly 10. The simulation results were iterated over 50 times on average. The software used in this simulation was MATLAB.

In the evaluated network, the number of channels ranges from ch1 to ch10 while the number of nodes of the simulation environment are within the range of 10 to 50. It consists of a primary user (PU) and secondary users (SUs).These nodes are distributed randomly and uniformly in a two dimensional simulation of 800 by 800 square meters. The buffer of each node is unlimited and the transmission range of each node is 200 meters. The data transfer rate or CBR[18] is 2 Mbps . . The speed of nodes is 0 to 5 m/s and the number of CBRs between the source and destination is randomly 10. The simulation results were iterated over 50 times on average. The software used in this simulation was MATLAB.

6. Checking the improvement parameters of channel usage

In the conducted experiments, the efficiency of the channel allocation plan has been measured according to the following criteria:

- Throughput flow chart

This factor represents the network throughput, which is the ratio of the average number of received packets to the total number of sent packets. This metric is optimized using the proposed algorithm, in which the bandwidth allocated to each secondary user is proportional to their number of requests.

Figure 6 illustrates the throughput of the proposed BPST-MAC and MRLA algorithms, which is a function of traffic of nodes. As can be seen from the results, the BPST-MAC algorithm has the best performance via the mechanisms considered such as the prediction of PU behavior via Markov on the spectrum, resulting in faster learning. Furthermore, through BPST-MAC algorithm, the convergence rate rises, as a result our method converges faster than MRLA and its results are better than MRLA due to employing the two fairness mechanisms and Markov we used in the channel allocation. Hence, the response received from the proposed method is better than that from MRLA. Even with increasing the network traffic, it converges faster to the best channel thanks to its high convergence rate. The MRLA method has used L_{Rep}, L_{RI}, L_{RP}, but we used an algorithm whose convergence speed and accuracy have been higher.

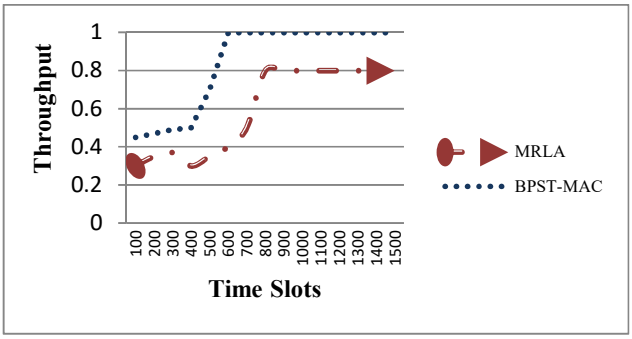

Figure 6. Throughput flow chart based on Timeslot

[17] Constant Bit Rate

[18] Constant Bit Rate

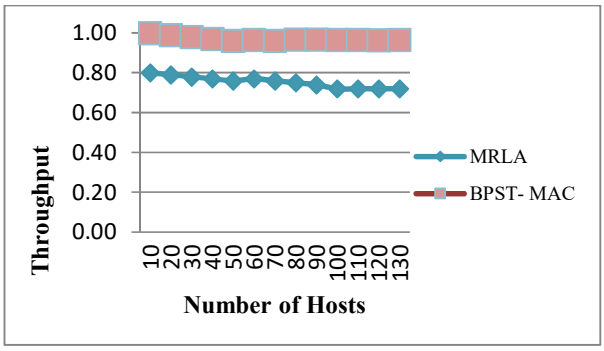

Figure 7. Throughput flow chart based on the number of secondary users (host)

Figure 7 compares the proposed method of BPST-MAC with the MRLA algorithm. The results suggest that the proposed method functions better than MRLA with increasing traffic and data transmission volume. The reason is that the proposed method uses Markov rule and estimation method for spectrum allocation, which enjoys a high convergence rate and more accurate estimation.

- **Channel Switching parameter between channels**

The switching cost in the proposed method which uses the BPST-MAC algorithm is less than that of the other learning algorithms. According to the priority among secondary users which use the fairness mechanism and pursuit algorithm enjoying a high convergence rate and updating the probability of actions based on the estimation of rewards, the proposed method helps preserve the result of the selected channel history. Consequently, the proposed method chooses better channels, and in the subsequent repeats the number of channel switching is reduced. Since we used the Markov model to predict PU behavior, so when our algorithm wants to choose a channel it selects it more quickly and accurately, thereby reducing the extent of switching between channels, since it chooses a good channel faster. When the speed convergence grows because of using the BPST-MAC algorithm, the channel is found earlier, thus lessening the extent of switching between channels. The following results have been obtained after 1000 replications in timeslot = 1000. (Figure 8)

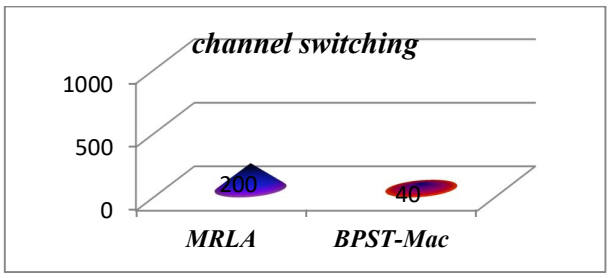

Figure 8. The amount of switching between channels in Time slot = 1000

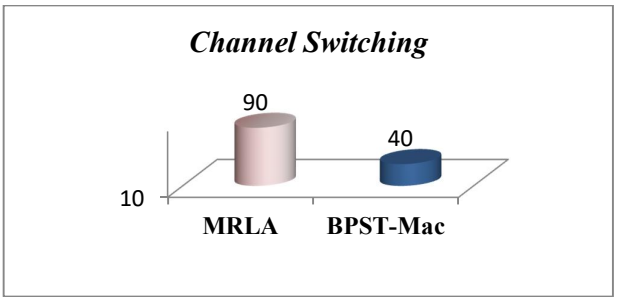

Figure 9. Channel switching rate between channels with five secondary users

Figure 9 shows the channel-switching rate between the proposed method and MRLA with five secondary users. Simulation results show that by increasing the number of hosts due to increased demand for data transmission, the switching rate to find the appropriate channel among secondary users in MRLA method is more than BPST-Mac method.

- **Channel utilization**

The channel availability rate for secondary users is contingent upon their requests. This metric calculates the average channel utilization, with the aims of allocating bandwidth to each node in line to the needs of the host and then improving channel utilization, our proposed channel allocation algorithm is expected to have better performance than MRLA algorithm. Over time, the graph in figure 10 indicates that in the BPST method, the convergence rate increases due to employing Markov mechanism and fairness index. The percentage of channel usage in the proposed method is about 20% better than that of MRLA method. Also, as can be observed, the percentage of the channel usage declines with increasing the number of nodes due to the rise in the number of requests. So, when the number of nodes increases, the efficiency or utilization diminishes. In this test, the number of CBRs varies from 5 to 10 randomly. (Figure 10)

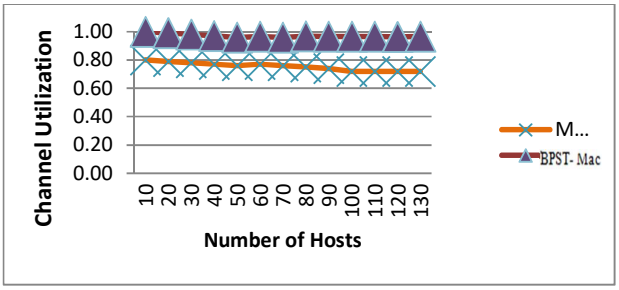

Figure 10. Utilization based on the number of secondary users or host

The switching cost in the proposed method, which uses the Pursuit algorithm, is less than other learning algorithms. Figure 11 shows that the proposed method is considering the priority given to the secondary users whose requests are in the first repetitions, as well as considering the use of pursuit algorithm with a high convergence rate and updating the probability of actions based on the reward

estimation, the selected channel history is also considered. As a result, the proposed method chooses better channels and reduces the number of channel switching in subsequent replications.

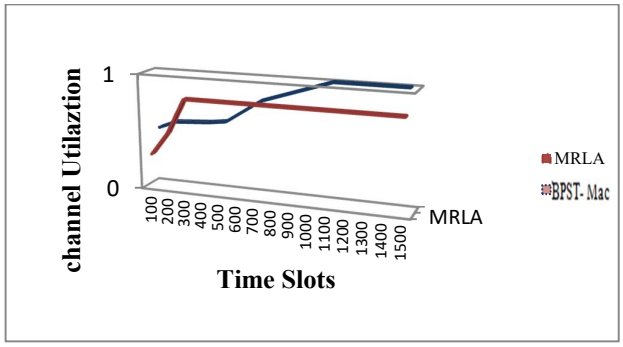

Figure 11.Utilization based on time slot

- The parameter of justice index

According to Figure 12, because of employing fairness mechanism according to the Formula $J(X_1,X_2,\ldots,X_N) = \frac{(\sum_{i=1}^{N} X_i)^2}{N.\sum_{i=1}^{N} X_i^{\,2}}$, if several nodes submit a request simultaneously, priority is given to the node with more requests and less access to the channel. In other words, our algorithm increases the probability value when our node requests more channels, but the channel is not given. Hence, according to the algorithm, we give the channel to the nodes with a high probability value, so we establish fairness in such a way that if the probability value of the node is high, this means that the node has awaited a lot for its requests. Thus, we will give the channel first to this node, while in MRLA, the fairness index is only calculated and there is no mechanism for it.

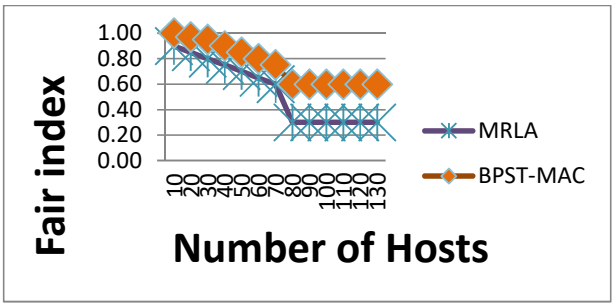

Figure 12.The justice index based on the number of secondary users or host

7. Conclusion and Future work

The cognitive radio based on the learning automata is a potential technology to cope with the challenge of spectrum deficiency in cognitive radio networks. Using the proposed method, we could significantly increase the efficient usage of the frequency spectrum, so we could improve the quality of service parameters. We used learning automata in the scenario of secondary users to access the dynamic spectrum in cognitive radio networks. Also, we showed that the algorithm for access to the dynamic spectrum based on automata can control the competition between secondary users and the use of accessible channels. Further, we used the HMM [19] rule to predict the presence or absence of the primary user in the channel and we used the fairness index mechanism to control the competition between users who simultaneously requested use of the channel. This will lead to reduced collision rate, increased total throughput, and diminished switching cost between channels. We also showed that the Bayesian algorithm uses the history of the selected actions and feedback received to calculate the probability of operating rewards. In this algorithm, we selected the channel from a list of channels based on its value of the probability distribution and then estimated its reward. Unlike MRLA, updating the probability of an operation in Bayesian algorithm does not depend on identification of the selected channel. In this way, we resolved the problem of spectrum deficiency in networks using a cognitive radio network based on learning automata.

Considering the rate of traffic variation in spectrum allocation, as traffic changes in each node, an idea could be using a channel with variable width to optimize the spectrum usage and improve the quality of service. This could be a proposal for a new plan for the future.

References:

[1] Chen, R. and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in Networking Technologies for Software Defined Radio Networks, 2006. SDR'06.1 st IEEE Workshop on. 2006. IEEE.

[2] He. A. et al., "A survey of artificial intelligence for cognitive radios," IEEE Transactions on Vehicular Technology, 2010. **59**(4): p. 1578-1592.

[3] Cormio, C. and K.R. Chowdhury, "A survey on MAC protocols for cognitive radio networks," Ad Hoc Networks, 2009. **7**(7): p. 1315-1329.

[4] Kordali, A.V. and P.G. Cottis, "A reinforcement-learning based cognitive scheme for opportunistic spectrum access," Wireless Personal Communications, 2016. **86**(2): p. 751-769.

[5] Vidyarthi, D.P. and S.K. Singh, "A Heuristic Channel Allocation Model Using Cognitive Radio," Wireless Personal Communications, 2015. **85**(3): p. 1043-1059.

[6] Cordeiro, C. and K. Challapali, "C-MAC: A cognitive MAC protocol for multi-channel wireless networks," in New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on. 2007. IEEE.

[7] Zhao, J., H. Zheng, and G.H. Yang, "Spectrum sharing through distributed coordination in dynamic spectrum access networks," Wireless Communications and Mobile Computing, 2007. **7**(9): p. 1061-1075.

[8] Torkestani, J.A. and M.R. Meybodi, "A learning automata-based cognitive radio for clustered wireless ad-hoc networks," Journal of Network and Systems Management, 2011. **19**(2): p. 278-297.

[9] Bizhani, H. and A. Ghasemi, "Joint admission control and channel selection based on multi response learning automata (MRLA) in cognitive radio networks," Wireless personal communications, 2013: p. 1-21.

[10]Jiao L. et al., "Optimizing channel selection for cognitive radio networks using a distributed Bayesian

[19] Hidden Marcov Model

learning automata-based approach," Applied Intelligence, 2016. 44(2): p. 307-321.

[11] Golestanian M. et al., "A learning automata based spectrum prediction technique for cognitive radio networks," International Transaction of Electrical and Computer Engineers System, 2014. 2(3): p. 93-97.

[12] 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. on Wireless Comm., Vol. 11, No. 4, 2012, pp. 1380-1391.

[13] O. C. Granmo and S. Glimsdal, "Accelerated Bayesian learning for decentralized two-armed bandit based decision making with applications to the Goore game," Applied Intelligence, Vol. 38, No. 4, 2013, pp. 479-488

[14] Torkestani, J.A. and M.R. Meybodi, "An intelligent backbone formation algorithm for wireless ad hoc networks based on distributed learning automata," Computer Networks, 2010. **54**(5): p. 826-843.

[15] T. A. Tuan, L. C. Tong, and A. B. Premkumar, "An adaptive learning automata algorithm for channel selection in cognitive radio network," Proceedings of the IEEE International Conference on Communications and Mobile Computing, Shenzhen, China, Apr. 2010, pp.159-163.

[16] Ali, A., J. Qadir, and A. Baig, "Learning automata based multipath multicasting in cognitive radio networks," Journal of Communications and Networks, 2015. 17(4): pp. 406-418.

[17] Thathachar, M.A. and P.S. Sastry, "Networks of learning automata: Techniques for online stochastic optimization," 2011: Springer Science & Business Media.

[18] Oommen, B.J. and M. Agache, "Continuous and discretized pursuit learning schemes: Various algorithms and their comparison," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2001. 31(3): p. 277-287.

[19] Zhang, X., O.-C. Granmo, and B.J. Oommen, "On incorporating the paradigms of discretization and Bayesian estimation to create a new family of pursuit learning automata," Applied intelligence, 2013. 39(4): p. 782-792.

[20] Zhang, X., Granmo, O.-C., Oommen, B. J., "The Bayesian pursuit algorithm: a new family of estimator learning automata," in Proceedings of the 24th international conference on Industrial engineering and other applications of applied intelligent systems conference on Modern approaches in applied intelligence-Volume Part II 2011, pp. 522-531. Springer-Verlag.

[21] Herath, S.P., N. Rajatheva, and C. Tellambura, "Unified approach for energy detection of unknown deterministic signal in cognitive radio over fading channels," in Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on. 2009. IEEE.

[22] Thathachar, M.A. and P.S. Sastry, "Varieties of learning automata: an overview," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2002. **32**(6): pp. 711-722.

[23] Sutton, R.S. and A.G. Barto, "Reinforcement learning: An introduction," Vol. 1. 1998: MIT press Cambridge.

[24] He. A. et al., "A survey of artificial intelligence for cognitive radios," IEEE Transactions on Vehicular Technology, 2010. **59** (4): p. 1578-1592.

[25] Mitola, J. and G.Q. Maguire, "Cognitive radio: making software radios more personal," IEEE personal communications, 1999. **6**(4): p. 13-18.

[26] Xing, Y., Mathur, C. N., Haleem, M. A., Chandramouli, R. and Subbalakshmi, K. P., "Dynamic spectrum access with QoS and interference temperature constraints," IEEE Transactions on Mobile Computing, 6(4), pp. 423–433.

[27] Meybodi, M. R., Learning automata and its application to priority assignment in a queuing system with unknown characteristics, Ph.D. thesis, Department of Electrical Engineering and Computer Science, University of Oklahoma, Norman, Oklahoma, USA.

[28] Mekki, S. et al., "Chi-squared distribution approximation for probabilistic energy equalizer implementation in impulse-radio UWB receiver," in Communication Systems, ICCS 2008, 11th IEEE Singapore International Conference on. 2008. IEEE.

[29] Yucek, T. and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," IEEE communications surveys & tutorials, 2009. **11**(1): pp. 116-130.

[30] Ma, L., C.-C. Shen, and B. Ryu, "Single-radio adaptive channel algorithm for spectrum agile wireless ad hoc networks," in New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2007, 2nd IEEE International Symposium on. 2007, IEEE, pp. 547–558.

[31] Eslamnour, B., M. Zawodniok, and J. Sarangapani, "Dynamic Channel Allocation in Wireless Networks Using Learning Automata (Preprint)," 2009, Missouri Univ-Rolla, Dept. of Mechanical and Aerospace Engineering.

[32] Rastegar, R. et al. "A fuzzy clustering algorithm using cellular learning automata based evolutionary algorithm," in Hybrid Intelligent Systems, 2004. HIS'04. Fourth International Conference on. 2004. IEEE.

[33] Jianli, Z., W. Mingwei, and Y. Jinsha, "Based on neural network spectrum prediction of cognitive radio," in Electronics, Communications and Control (ICECC), 2011 International Conference on. 2011. IEEE

[34] Hossain, E., D. Niyato, and Z. Han, "Dynamic spectrum access and management in cognitive radio networks," 2009: Cambridge university press.

[35] Narendra, K.S. and M.A. Thathachar, "Learning automata- a survey," IEEE Transactions on systems, man, and cybernetics, 1974(4): p. 323-334.

[36] Akyildiz, I., W. Lee, and K. Chowdhury, "Cognitive radio ad hoc networks: research challenges," in Ad Hoc Networks Journal, 2009, Elsevier.

[37] Čabrić, D., et al. "A cognitive radio approach for usage of virtual unlicensed spectrum," in 14th IST mobile and wireless communications summit, 2005.

[38] Tsagkaris, K., A. Katidiotis, and P. Demestichas, "Neural network-based learning schemes for cognitive radio systems," Computer Communications, 2008. **31**(14): pp. 3394-3404.

[39] Granmo, O.-C., "Solving two-armed bernoulli bandit problems using a bayesian learning automaton," International Journal of Intelligent Computing and Cybernetics, 2010. 3(2): pp. 207-234.

[40] Jiao, L., et al., "A Bayesian learning automata-based distributed channel selection scheme for cognitive radio networks," in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. 2014. Springer.

[41] Song, Y., Y. Fang, and Y. Zhang, "Stochastic channel selection in cognitive radio networks," in Global

Telecommunications Conference, GLOBECOM'07. IEEE. 2007. IEEE.
[42] Lei, J., et al., "Optimization channel selection for cognitive radio networks using a distributed Bayesian learning automata-based approach," 2015.
[43] Narendra, K.S. and M.A. Thathachar, "Learning automata: an introduction," 2012: Courier Corporation.
[44] Thathachar, M. and P.S. Sastry, "A hierarchical system of learning automata that can learn die globally optimal path," Information sciences, 1987. 42(2): p. 143-166.
[45] Thathachar, M. and B.R. Harita, "Learning automata with changing number of actions," IEEE transactions on systems, man, and cybernetics, 1987. 17(6): p. 1095-1100.
[46] Lanctôt, J., "Discrete estimator algorithms: A mathematical model of computer learning," 1989, M.Sc. Thesis, Department of Mathematics and Statistics, Carleton University, Ottawa, Canada.
[47] Lanctôt, J.K. and B.J. Oommen, "Discretized estimator learning automata," IEEE Transactions on Systems, Man, and Cybernetics, 1992. 22(6): p. 1473-1483.
[48] Akyildiz, I.F., et al., "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," Computer networks, 2006. **50**(13): p. 2127-2159.
[49] Thathachar, M., "A class of rapidly converging algorithms for learning automata," in IEEE International Conference on Cybernetics and Society, 1984.

Panteha Ghorbani Hagh received a Bachelor's degree in software engineering from Allameh Mohades Noori University in Mazandaran and a Master's degree in software engineering from Islamic Azad University branch of Damavand. Her interests are Cognitive Radio network and Machine learning. She has been working as an instructor in technical and vocational schools.

Email address: ghorbani.hagh@yahoo.com

Parisa Rahmani received a Bachelor's degree from Islamic Azad University branch of Qazvin in 2004, a Master's degree in software engineering from Islamic Azad University branch of Qazvin in 2006 and now she is a Ph.D. candidate in software engineering in Islamic Azad University, Science and research branch of Tehran. Her current research topics include Machine learning, heuristic algorithms, resource allocation in wireless network such as channel and power allocation in ad hoc networks.

Email address: rahmani@pardisiau.ac.ir

Paper Handling Data:

Submitted: 06.02.2018
Received in revised form: 03.09.2018
Accepted: 10.09.2018
Corresponding author: Panteha Ghorbani Hagh
Department of computer engineering Damavand branch, Islamic Azad University, Damavand, Iran

Appendix:

The flowchart of the proposed algorithm: