

# A QoS Aware Learning Automata Based Channel Assignment Method in Cognitive Network

Setareh Rasaneh<sup>1</sup> · Mohsen Jahanshahi<sup>2</sup>

Published online: 29 May 2017  
© Springer Science+Business Media New York 2017

**Abstract** With the rapid growth of wireless services and technologies, and as a result, increasing demand for the spectrum, cognitive network (CN) has been proposed as a promising solution for accomplishing the problem of spectrum scarcity. Channel assignment (CA) problem in CN is NP-complete and hence, in terms of scalability it should be solved through heuristic or meta-heuristic approaches. The main goal of dynamic CA in CN environment is to improve the QoS provisioning for SUs as well as protecting the Primary Users (PUs) from interference. To achieve this goal, a Secondary User may have to change its transmission channel several times during the established communication. Channel switching is a costly task due to the time overhead and interruptions that imposes on the given communication. To overcome these problems, two learning based dynamic CA methods are introduced; In the first method, by proposing a channel allocation scheme based on learning automata a trade-off between different QoS parameters will be made. In the second method, it is trying to decrease the overhead and interruptions because of switching channels of the first algorithm. Simulation results demonstrate the superiority of the proposed method in terms of delay, data delivery ratio, throughput, and data loss rate in comparison with other representative methods.

**Keywords** Cognitive network · Channel assignment · Learning automata · QoS requirement · Channel switching

---

✉ Mohsen Jahanshahi  
mjahanshahi@iauctb.ac.ir  
Setareh Rasaneh  
SRasaneh@iauctb.ac.ir

<sup>1</sup> Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran

<sup>2</sup> Young Researchers and Elite club, Central Tehran Branch, Islamic Azad University, Tehran, Iran

## 1 Introduction

Advances in the widespread wireless services and technologies, have led to a significant increase in demand of the available spectrum. The Federal Communications Commission (FCC) report indicates that, most of the allocated spectrum is significantly underutilized in temporal and spatial due to the inefficient static frequency allocation policy [1]. A cognitive radio network is an emerging technique to mitigate the spectrum scarcity, by allowing the Secondary Users (SUs) access to the underutilized licensed channels of PUs temporarily [2, 3].

The fixed CA strategy can be mapped to graph coloring algorithm, which is known as NP-complete problem [4]. Dynamic CA in CNs is also NP-complete, since this is even harder than the fixed CA scheme. As the size of the problem grows, NP-complete problems cannot be solved in polynomial time. To address this issue, heuristic or meta-heuristic approaches are often used to speed up the process in the event that exhaustive searches are impractical.

Learning and intelligence capabilities are two significant characteristics of the CN. Taking advantage of these features to predict the spectrum condition and the holes positions, can dramatically improve the performance of distributed CNs [5]. LA is one of the meta-heuristic methods that has been recently utilized to solve different NP problems efficiently. Because of close relation between the CA problem requirements in CN and inherent characteristics of learning automata (LA), using LA is a reasonable idea to cope with this problem.

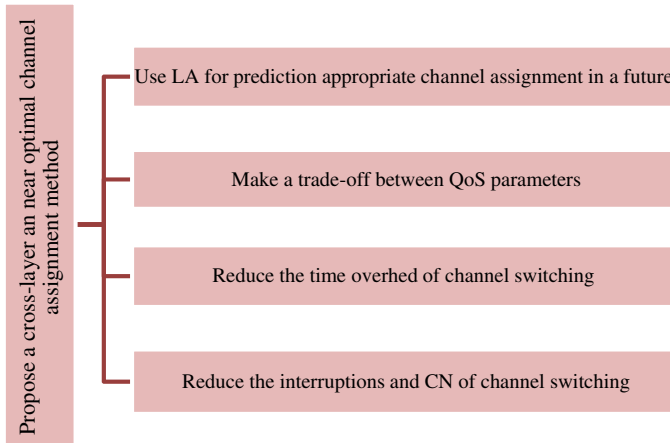
The main goal of dynamic CA in CR environments is to increase the quality of SU's transmission, while protecting PUs from interference. PUs have higher priority than SUs. When a PU appears in the channel, SUs have to vacate this channel and resume their unfinished transmissions on the new selected channels. Therefore, PUs will be protected from SUs interferences and their communication quality will be guaranteed. On the other hand, once the quality of communication is unsatisfied, channel switching can be a helpful task for increasing the SU's QoS.

Channel switching is an expensive task due to the communication time overhead. In addition, in some cases like a real-time data transmission, the interruptions in communication caused by the frequent channel switching, are unacceptable.

Therefore, In this paper, two fully distributed learning based, QoS aware CA methods are proposed with the following main contributions:

- The online feature of the learning automata is used for adapting to the dynamic characteristic of CNs. This, helps to predict the future of spectrum conditions. In addition, here, parameters of the LA are traffic aware and tuned dynamically, in order to increase its adaptability.
- In addition, appropriate using of spectrum holes without interfering with PUs and a tradeoff between several QoS requirements of SUs (i.e., throughput, delay, data delivery ratio, data lost) will be made.
- The channel switching cost will be reduced to a desirable level, while keeping the communication quality in a good condition.
- Interruptions caused by several channels switching will be decreased and hence, a communication with an appropriate level of disruption is setting up.

Figure 1 depicts a summary of the contributions.



**Fig. 1** Summary of the paper contributions

The rest of this paper is organized as follows: In Sect. 2, the related works are discussed in details. The theory of LA will be proposed in Sect. 3. In Sect. 4, system assumptions and models are introduced. Proposed algorithms and their flowcharts are described in Sect. 5. In Sect. 6 a comprehensive analysis of simulation results will be provided. Finally, in Sect. 7 article is concluded and some interesting research issues for future works are presented.

## 2 Related Works

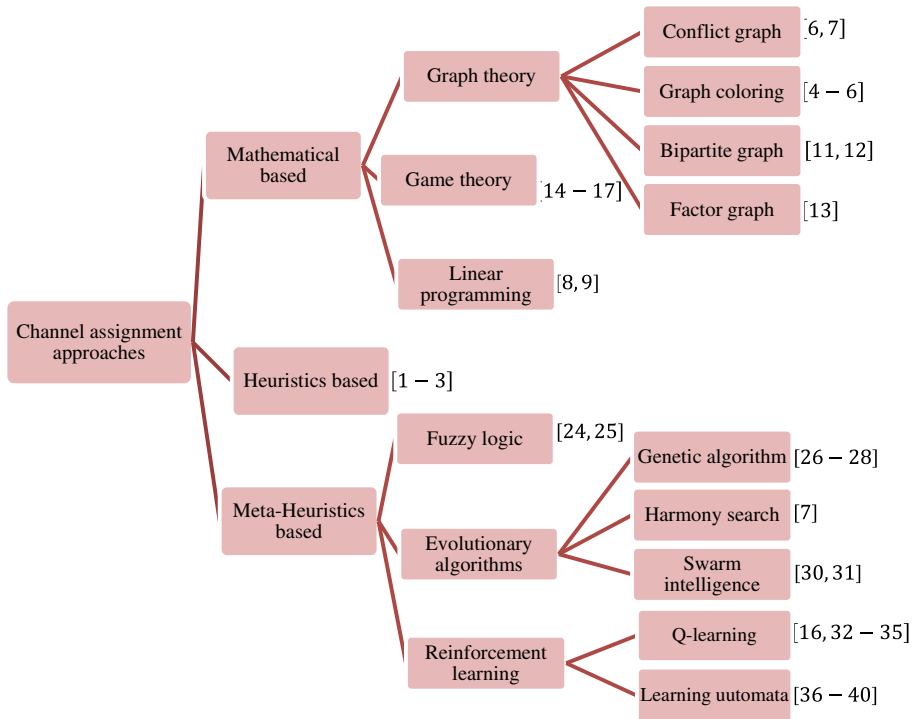
In the context of the CA problem in the CN, many researches have been carried out recently, that can be generally classified as Fig. 2.

Several graph-theory related solutions such as network conflict graph [6, 7], Graph coloring [8–10], bipartite graph [11, 12], and factor graph [13] have been proposed to cope with CA problem in CN. However, these approaches cannot incorporate all of CN parameters such as QoS requirements and Adjacent Channel Interference (ACI).

In the literature, many works have used game theory to deal with the CA problems in CN [14–17]. This powerful mathematical tool can represent the cooperation and competition between SUs quite well. However, is it difficult to structure the game in such a way that always equilibrium is guaranteed.

Another commonly used mathematical technique for solving channel allocation problem in the CN is linear programming [18, 19]. The Advantage of using this approach is that the existing LP techniques can be utilized. But, the channel allocation can only be formulated as a Mixed Integer Non-Linear Programming (MINLP). MINLP is NP-hard and hence is not scalable. To address this, it should be transformed to Binary Linear Program (BLP). This transformation cannot be guaranteed because, it requires several assumptions for converting continuous variables to binary, which may not always be valid [20].

As noted earlier in Sect. 1 CA problem in CN is an NP-complete and hence often heuristic and meta-heuristic approaches should be used to speed-up the process of finding a sub-optimal solution in a scalable way.



**Fig. 2** Classification of CA approaches

Heuristic approaches do not have a specific algorithmic solution. Therefore, many simple algorithms have proposed in this category to tackle the problem of CA in CN [21–23]. Heuristic algorithms are simple and can be easily implemented. In theory, these methods are problem-independent and show less sensitivity to changes in the data quality and problem characteristics. However, most of the proposed heuristic solutions are problem-specific and therefore, cannot be used for accomplishing other problems. In addition, they may trap into local optima that may be far from the global optimal solution. Finally, there is no methodology for studying about the convergence of these methods.

Meta-heuristic based methods can be categorized in three sub-classes. The first one is fuzzy logic [24, 25], that can achieve the solution quickly based on the predefined rules. In addition, learning techniques can help for improving the quality of this solution. However, a fuzzy system is not scalable and needs too many rules for covering all the network parameters. Finally, because of the dynamic nature of CNs, definition of accurate rules is so hard. The second subclass of the reported meta-heuristic based methods is evolutionary algorithms, which include genetic algorithm [26–28], harmony search [29] and swarm intelligence [30, 31]. The arbitrary goals and constraints can be managed by these approaches. In addition, inappropriate solutions will be ignored simply. However, the process of finding a good solution is very slow and can lead to find a local optimal solution rather than the global one. The last subclass is reinforcement learning-based (RL based) algorithms [16, 32–35]. RL helps agents be compatible with the dynamic and uncertain environment. The proposed method in this paper uses the LA which can be placed in this

category. In recent years, some ideas have been proposed to cope with the CA problem by using LA [36–40] as well, that will be discussed as follows:

In [36] the authors proposed a stochastic channel selection algorithm based on LA with the aim of reducing the channel switching times. The action probability vector changes according to the environmental feedback in a way that, a channel with the highest probability of successful transmission will have a higher chance for selection in the future epochs. This algorithm, tries to minimize the collision probability to avoid the costly channel switching. However, since channel quality and throughput are ignored in this algorithm, the QoS satisfaction cannot be guaranteed.

The proposed algorithm in [36] has been extended in [37]. Here, the collision probability and channel quality are considered simultaneously with the aim of maximizing the throughput. Both [36, 37] assume a network with stationary condition for the PU traffic distribution. In other words, the probability that a PU uses the communication channel does not change over the time. However, in the practical networks, SUs should communicate in a non-stationary environment.

In [38] a LA based CA algorithm has been proposed for adapting to the dynamic environment of CN. In this work, with tracking the notable changes in the reward estimator vector, altering the environment is recognized. After detecting this change, the algorithm is reinitiated, which is a very time-consuming task. In addition, this algorithm the same as [36, 37] only is developed for a single SU scenario and the effects of other SUs are not considered in the network behavior. Finally, in [38] like [36] QoS requirements are not guaranteed, because the channel quality and throughput are neglected.

In [39] another LA based spectrum allocation algorithm was proposed with the aim of reducing the delay, packet loss, and communication cost because of the frequently channel switching. In this work, a central broker collects the information about the states of the spectrum and demands of users into a database. Central broker controls the spectrum allocation through this information. This algorithm, unlike [36–38], regards the competition between SUs in the network. In addition, it tries to guarantee some of the QoS requirements (e.g. delay, fairness). For improving the fairness, one user can only use one channel and for avoiding conflict one channel can only be used by one user. Both of these, reduce the channel utilization. In addition, the overhead of obtaining and updating the network information are not negligible. Finally, the broker is a single point of failure and hence the network reliability cannot be guaranteed.

In [40] a multi response LA based algorithm for dynamic spectrum access in CN was introduced. Each SU is equipped with a LA. The LA has several actions equal to the number of primary channels. In addition, an admission control mechanism was proposed in order to restrict the number of competing SUs and reducing the rate of collisions between them. In this scheme, each SU is admitted with probability  $\psi$  (Action mode) and blocked with probability  $1 - \psi$  (No Action mode). When the rate of the collision increases,  $\psi$  should be decreased in order to decrease the number of SUs that are competing for each channel and vice versa. To promise fairness, the  $\psi$  parameter of each SU that is blocked in the current time, will be increased. This algorithm improves some of QoS requirements (e.g. total throughput, the number of switching, fairness). Moreover, in this work the competition among SUs is considered unlike the most of learning based algorithms that have been introduced later. Disadvantages of this algorithm will be discussed in Sect. 6.

In the most of the previous works [33, 36–40], an SU switches the communication channel only when a PU appears in this channel. Their authors argue that, channel switching is a very costly task and should be avoided as much as possible. In some other works [32, 35], cognitive users not only switch the channel when a PU appears in it, but

also vacate the communication channel when their quality becomes unacceptable. This is because, in these works satisfying of some QoS requirement (e.g. blocking and dropping probability) is more important than the overhead of switching. However, they also try to reduce the switching costs as much as possible.

All the above methods only decreased the overhead of channel tuning by reducing the number of switching by selecting the suitable channels for communication. However, even afterwards this overhead may be significant. Therefore, new methods are needed to overcome this problem.

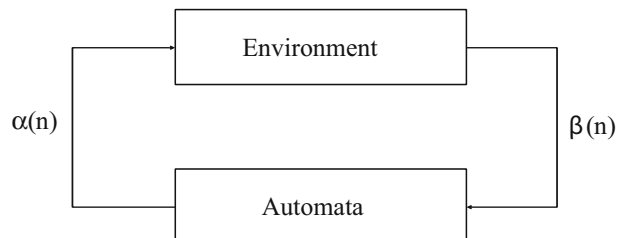
### 3 Learning Automata

A learning automaton is an adaptive decision-making entity that is one of the main fields of artificial intelligence. The automaton attempts to find a solution for the problem without any information about suitable action [41]. LA improves its operation through learning how to choose the optimal action from a finite set of available actions by interacting repeatedly with a stochastic environment. The action is chosen randomly based on a probability distribution. At each instant, the given action serves as an input to the random environment. The environment, responds the taken action in order to the reinforcement signal. Then, the internal information of LA is updated based on the given feedback from the environment [42, 43]. Afterwards, LA adjusts its action repeatedly until a termination condition is satisfied. Figure 3 shows the relationship between a learning automaton and its random environment.

Every environment is represented by:  $E = \{\alpha, \beta, c\}$ , where  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  is a set of inputs,  $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$  is a set of outputs, and  $c = \{c_1, c_2, \dots, c_r\}$  is a set of penalty probabilities, that  $c_i$  denotes the penalty probability of each taken action  $a_i$ . The environments depending on the nature of the reinforcement signal  $\beta$  can be categorized into three types: P-model, Q-model and S-model. If the  $\beta$  set is binary, the model is known as a p-model. In this model,  $\beta_1 = 1, \beta_2 = 0$  represent the penalty and reward, respectively. Similarly, a model is called the Q-model if the  $\beta$  set contains a finite number of values in the interval  $[0, 1]$ . On the other hand, in the S-model,  $\beta$  set has an infinite number of members in the interval  $[0, 1]$ .

LA can be classified into a fixed structure (FSLA) and variable structure (VSLA). In the FSLA, the transition and the output functions are time invariant. A learning automaton with variable structure is represented by  $\{\alpha, \beta, p, T\}$ . In the definition of a VSLA, the LA is completely defined by a set of actions as  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ , (one of which is the output of the automata), a set of inputs as  $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$  (which is usually the response of the environment), action probability vector as  $p = \{p_1, p_2, \dots, p_r\}$  and finally learning algorithm as  $p(n + 1) = T[\alpha(n), \beta(n), p(n)]$ . The learning automata randomly chooses one

**Fig. 3** Learning automata diagram



action according to its probability vector ( $p_i$ ). The LA chooses action  $\alpha_i$  and applies it to the environment. Thereafter, LA receives a reinforcement signal from the environment and then updates its action probability vector. If the response from the environment is desirable, automata updates it's action probability vector as follows:

$$\begin{aligned} p_i(n + 1) &= p_i(n) + a(1 - p_i(n)) & a(n) &= a_i \\ p_j(n + 1) &= p_j(n) - ap_j(n) & a(n) &= a_i, \forall j, j \neq i \end{aligned} \tag{1}$$

In contrary, if the environmental feedback is unfavorable, the action probability vector will be updated as follows:

$$\begin{aligned} p_i(n + 1) &= (1 - b)p_i(n) & a(n) &= a_i \\ p_j(n + 1) &= \frac{b}{r - 1} + (1 - b)p_j(n) & a(n) &= a_i, \forall j, j \neq i \end{aligned} \tag{2}$$

In formula (1) and (2), a and b denote reward and penalty parameters respectively. If a and b are equal, the algorithm is called  $L_{R-P}$ . Whereas, for  $b \ll a$  the algorithm is  $L_{R\epsilon P}$ . Finally, if  $b = 0$  then the algorithm is said to be  $L_{R-I}$ .

In what follows, some principles will be expressed to clarify the reasonable use of LA to solve the problem of CA in CN; The first reason is that, CA problem is an NP problem [4] which can't be solved in polynomial time order. On the other hand, learning automaton is a Meta-heuristic algorithm that can overcome this problem. In addition, for CA in CN, the behavior of PUs and other SUs in the network should be accurately followed. Since, the nature of CN is unpredictable, an appropriate manner is needed to adapt to this dynamicity. LA is a dynamic and adaptive entity that tries to find an online solution, which can be helpful for optimization of CA problem in the non-stationary environment. Moreover, because of the dynamic nature of CNs, it is so hard to gather accurate and updated information about the network. The main characteristic of an LA is that, it learns the optimal action through interactions with the environment, without any prior knowledge. The environment can be treated as a "black box", and LA only needs the feedback signal.

Table 1 summarizes the aforementioned principles which can map the requirements of the CA problem to capabilities of LA.

### 4 System Model

In this Section, the system model and the basic policies, assumptions and limitations of the proposed method are presented.

**Table 1** Requirements of the channel assignment problem and capabilities of the learning automata

Capabilities of learning automata	Requirements of channel assignment in CN
Can provide a sub-optimal solution in polynomial time	Is an NP problem
Is online and dynamic	CN is a very dynamic environment
Does not require any prior information about the environment (only needs reinforcement signal)	It is so hard to gather accurate and update information about CN

A fully distributed and non-cooperative cognitive radio network is considered, which is consisting of  $M$  SUs (that are located in the fixed and random manner),  $N$  PUs and  $N + 1$  channels ( $N$  data channels and 1 control channel). Every SU is equipped with  $R + 1$  radio interfaces ( $R$  radios for data and 1 radio for control communication). Each of the  $R$  data radio can only transmit, receive or sense in data channels at one time. Control radio is devoted for operating over the dedicated control channel.

Traffic model of PU in the channel (presence or absence of the PU signal) follows a continuous time Markov chain with two states: idle (OFF) and busy (ON). This model has been used widely in the literature (e.g., [40]).

SUs use an overlay manner [5] for utilizing the PU’s channel. In this method, an SU only transmits opportunistically over the licensed spectrum once the PU is in idle period. As soon as a PU is detected on the channel (transits to ON state), SUs are forced to vacate the occupied channel. In other words, PUs have the preemptive priority to access the channels.

For quantifying the quality of a link, signal to interference plus noise ratio (SINR) is applied. Equation (3) calculates the SINR of a packet that is sent from a sender  $SU_i$  to the receiver  $SU_j$  when  $SU_k$  and  $SU_i$  are using the same channel [44].

$$SINR = \frac{P_i G_{ij}}{\sum_{k \neq i} P_k G_{kj} + \sigma^2} \tag{3}$$

where  $P_i$  is the transmission power by transmitter  $i$  and  $\sigma^2$  is the noise.  $G_{ij}$  is the channel gain and can be mathematically expressed as Eq. (4) [44]. Here,  $d_0$  the far-field crossover is distance and  $\eta$  is the isotropic path loss exponent.

$$G_{ij} = \left[ \frac{d_0}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \right]^\eta \quad \forall i, j \text{ and } i \neq j \tag{4}$$

According to Eq. (3) noise of the communication channel is inversely proportional to the SINR of received packets. On the other hand, it is clear that, noise is an inevitable problem in wireless network’s channels. Hence, to avoid the wrong decision influenced by sudden noises, a new metric for decision making about the quality of communication channels is considered. This metric is called as SINR-history (SINR\*).  $SINR^*(n + 1)$  is calculated as follows:

$$SINR^*(n + 1) = SINR(n + 1) + (1 - \alpha)SINR^*(n) \quad 0 \leq \alpha \leq 1 \tag{5}$$

where  $0 \leq \alpha \leq 1$  is the historical rate. In Eq. (5) the greater amount of  $\alpha$  gives more weight to the last packet’s SINR. On the other hand, the smaller value of  $\alpha$ , grants more importance to the SINRs of the past epochs.

$SINR_{Thr}$  is the minimum level of SINR for acceptable communication signal quality. In other words, the channel quality in transmission from  $SU_i$  to  $SU_j$  is acceptable if the SINR of the received packet at node  $j$  is greater than  $SINR_{Thr}$ . Packets with unfavorable quality should be discarded. In addition, in this paper, another level of SINR is introduced for decision making when the channel quality is degraded. This level is called  $SINR_{Critical}$  and can be expressed as follows:

$$SINR_{Critical} = SINR_{Thr} + \Delta \tag{6}$$



where  $\Delta > 0$  is a small value. This level, helps us predict about quality of channels in the near future. Critical range can be expressed as follows:

$$SINR_{Thr} < SINR^* \leq SINR_{Critical} \tag{7}$$

In other words, if after receiving a packet,  $SINR^*$  exists in the critical range, it can be predicted that this channel quality is not desirable for continuing this communication. In addition, this degradation is not influenced by the sudden noise.

### 4.1 Channel Switching Model

Channel switching is the process of changing the operational channels by an SU. In this paper, the communication channel will be changed in two states. As mentioned earlier, the overlay model is applied for spectrum sharing among SUs and PUs. Hence, at first, when the PU appears in the channel, SUs have to vacate the occupied channel. Second, continuing communication will be prevented in the low-quality channel by channel switching.

Channel switching process requires sensing time, handshaking time and exchanging time [45].  $T_{switching}$  can be calculated as follows:

$$T_{switching} = T_{sensing} + T_{handshaking} + T_{exchanging} \tag{8}$$

where  $T_{sensing}$  denotes a time that a SU is needed to select a new suitable channel.  $T_{handshaking}$  is the required time for negotiation about the selected channel between sender and receiver.  $T_{exchanging}$  is the duration time, which an SU needs for changing frequency band from current channel to the new selected channel.

Channel switching may degrade the SU’s performance by incurring longer delay and disrupts temporary communication [45]. In the first proposed algorithm, this delay and interruption will be imposed to the network for all channels switching. However, the second algorithm tries to prevent from delay and interruptions of channels switching through utilization of Backup Radio (BR).

In what follows, the properties and operation mechanism of BR are explained.

### 4.2 Backup Radio Model and Mechanism

In a multi radio network, all of radios are not often active simultaneously in all the times. During the communication, if an SU predicts that channel quality will not be acceptable in the near future and the SU has some idle radios, the BR is selected randomly from these radios. Under the assumption of every radio will be active (sending, receiving or sensing) with independent probability  $\mu$ , the probability that at least one BR is available can be expressed as follows:

$$P_{UnusedRadio} = 1 - \mu^{R-1} \tag{9}$$

where R is the number of data radios. In other words,  $P_{UnusedRadio}$  is the probability of finding a BR when the parallel operation is needed. For example, if an SU equips with five data radios and each of the radios is active in 75% of the time, then in 70% of the time BR can be found.

In the second proposed algorithm, when the critical condition occurs (according to (7)) communication in the current channel is not stopped. Instead, the algorithm tries to find a BR, activates it and then starts the process of selecting a new channel with sending operation (which is carried out by main radio) simultaneously. The BR activity includes

selecting a new channel (by automata) and then sensing selected channel to ensure that PU is not active in it. The time of these activities is equal to  $T_{sensing}$  in the channel switching process. After selecting a desired channel, handshaking process will be done with  $T_{handshaking}$ . Controlled radio performs the handshaking process in parallel to the main radio. Through this, controlled data will be exchanged between sender and receiver. After handshaking, main radio stops sending operation and after a negligible delay ( $T_{exchanging}$  on receiver radio), BR continues to data sending operation in a new suitable channel.

Figure 4 shows how the BR helps the main radio reduce the costs of channel switching.

### 4.3 Learning Model

In proposed model, every SU is equipped with one LA. The action set of learning automata is represented by  $ch = \{ch_1, ch_2, \dots, ch_N\}$  and Action probability vector by  $p = \{p_1, p_2, \dots, p_N\}$ , in which  $N$  is the number of channels. The probability of selecting channel  $i$  ( $i = 1..N$ ) is represented by the value of the corresponding index in action probability vector. When an SU has to select a new suitable channel to resume its unfinished transmission, LA randomly chooses one channel according to its probability vector. When data transmission is finished through the selected channel, the learning automaton updates its action probability vector proportional to communication quality.

As mentioned in Sect. 3, LA uses two parameters  $a$  and  $b$  for updating action probability vector. In this paper, these parameters are calculated dynamically, with the aim of adapting to the dynamic nature of CNs.  $a$  and  $b$  can be expressed as follows:

$$0 < a = \frac{SINR}{SINR_{max}} < 1 \quad \forall SINR \geq SINR_{Thr} \tag{10}$$

$$0 < b = \frac{SINR_{Thr} - SINR}{SINR_{max}} < 1 \quad \forall SINR < SINR_{Thr} \tag{11}$$

As shown in Eqs. (10) and (11)  $a$  and  $b$  dynamically change with the quality of communication. Therefore, Eqs. (1) can be updated as (12) in the case that the received signal is desirable.

$$\begin{aligned} p_i(n+1) &= p_i(n) + \frac{SINR}{SINR_{max}}(1 - p_i(n)) & a(n) &= a_i \\ p_j(n+1) &= p_j(n) - \frac{SINR}{SINR_{max}}p_j(n) & a(n) &= a_j, \forall j, j \neq i \end{aligned} \tag{12}$$

Similarly, Eq. (2) is updated as (13) when feedback from the environment is undesirable.

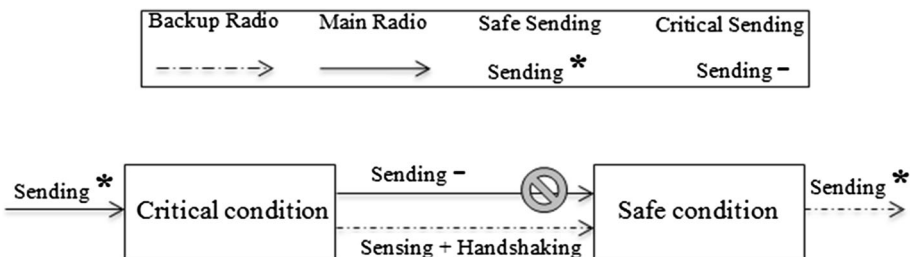


Fig. 4 BR mechanism

$$\begin{aligned}
 p_i(n + 1) &= \frac{SINR}{SINR_{max}} p_i(n) & a(n) &= a_i \\
 p_j(n + 1) &= \frac{SINR}{(r - 1)SINR_{max}} + \left(1 - \frac{SINR}{SINR_{max}}\right) & a(n) &= a_i, \forall j, j \neq i
 \end{aligned}
 \tag{13}$$

Thus, channels receive a punish or a reward corresponding to their quality. Using this method we are able to accurately differentiate the channels from each other in terms of quality.

### 5 Proposed Algorithms

In this Section, two dynamic channel assignment algorithms based on LA are proposed in a fully distributed fashion. These algorithms will be run on the all individual transmitter–receiver pairs and no information will be exchanged between every two users that already don’t communicate. The first algorithm is called QASA (Qos aware learning Automat base Spectrum Allocation) and the second algorithm is named DQASA (Delay and Qos aware learning Automat base Spectrum Allocation).

#### 5.1 Preparation Phase

In the preparation phase, two operations should be taken. First, the action probability vector is set with the same initial value for all actions *i* as follows:

$$p_i(t) = \frac{1}{N} \forall i; t = 0
 \tag{14}$$

where *N* is the number of channels. With this initialization, all channels have an equal chance to be selected at the first time. Second, the SINR<sub>k-max</sub> matrix should be calculated for SU<sub>k</sub> that is expressed as follows:

$$SINR_{k-max} = \begin{bmatrix} SINR_{max\ 11} & \cdots & SINR_{max\ 1n} \\ \vdots & & \vdots \\ SINR_{max\ m1} & \cdots & SINR_{max\ mn} \end{bmatrix} \quad 0 < k < m + 1
 \tag{15}$$

This matrix will be calculated and saved in the memory of each node *k*, in the preparation phase. The columns and rows of SINR<sub>k-max</sub> indicate channels and receiver nodes, respectively. Every element of this matrix shows the SINR<sub>max ij</sub> (*i* = 1..*m*, *j* = 1..*n*). That is, the maximum SINR in transmission from node *k* to receiver node *i* (*i* = 1..*m*) through the channel *j* (*j* = 1..*n*). For calculating SINR<sub>k-max</sub> matrix, this scenario is considered: one SU is a sender and other secondary nodes are receivers. The sender *k* transmits (*m* – 1)<sup>*n*</sup> packets to (*m* – 1) receivers through all *n* channels one by one. Upon receiving a packet, SINR is calculated and saved in the given matrix. This process will be repeated for all SUs. Since at one time only one communication occurs, interference is minimized and calculated SINR gets its maximum value.

#### 5.2 QASA: First Proposed Algorithm

In this Section, the operation and stages of the first proposed algorithm will be illustrated. The flowchart of QASA is shown in Fig. 6. The algorithm will be started after receiving

---

```

handle_message_QASA (msg, SINR*, channel, radio)
{
    if (channel = 0) then
    {
        channel ← select a channel by LA Algorithm;
        SINR* ← Initialize_SINR();
        radio.channel ← channel;           // The radio is tuned on the selected channel
    }
    if (PUs are active in the selected channel) then           // The channel must be changed
    {
        Penalize (channel);
        if ( All channels are occupied by PUs) then
            Wait (Random());
        channel ← 0;
    }
    else
    {
        Send (msg, radio);
        SINR ← radio.Get_SINR ();
        SINR* ← Calc_SINR* (SINR);
        if (SINR ≥ SINRThreshold) then           // Message is sent successfully
        {
            Reward (channel);
            msg ← Next_Packet();           //Next message of data flow is prepared
        }
        else           // Message is discard in receiver and must be send it again
        {
            Penalize (channel);
            if (SINR* < SINRThreshold + Δ) then           // The quality of selected
                channel ← 0;           //Channel is not suitable and must be changed
            }
        }
    }
    if (msg ≠ NULL) then           // Checks that any packet is remained
        handle_message (msg, SINR*, channel, radio);
}

```

---

**Fig. 5** Handle message algorithm of DQASA

the first packet of a new data flow. Then, SU tries to select a suitable channel by the LA. This operation will continue until the SU finds a free channel. Then, packets will be sent by SU through selected channel. Afterwards, in destination the quality of packet will be evaluated and accordingly, the following scenario will occurs:

In case of ( $SINR < SINR_{Thr}$ ) and ( $SINR^* \geq SINR_{Thr} + \Delta$ ), the packet is discarded; selected channel is penalized; action probability vector is updated, and communication will continue through the channel.

- In case of ( $SINR < SINR_{Thr}$ ) and ( $SINR^* < SINR_{Thr} + \Delta$ ), the packet is discarded; selected channel is penalized; action probability vector is updated; communication through the used channel is stopped, and SU retries to select a suitable channel.
- In case of ( $SINR \geq SINR_{Thr}$ ), used channel is rewarded; action probability vector is updated, and finally communication will continue through the channel.

This algorithm will be run for any generated data flow in senders and finishes when the last packet of this flow is sent successfully. The corresponding algorithms and the flowchart of QASA is shown in Figs. 5 and 6 respectively.

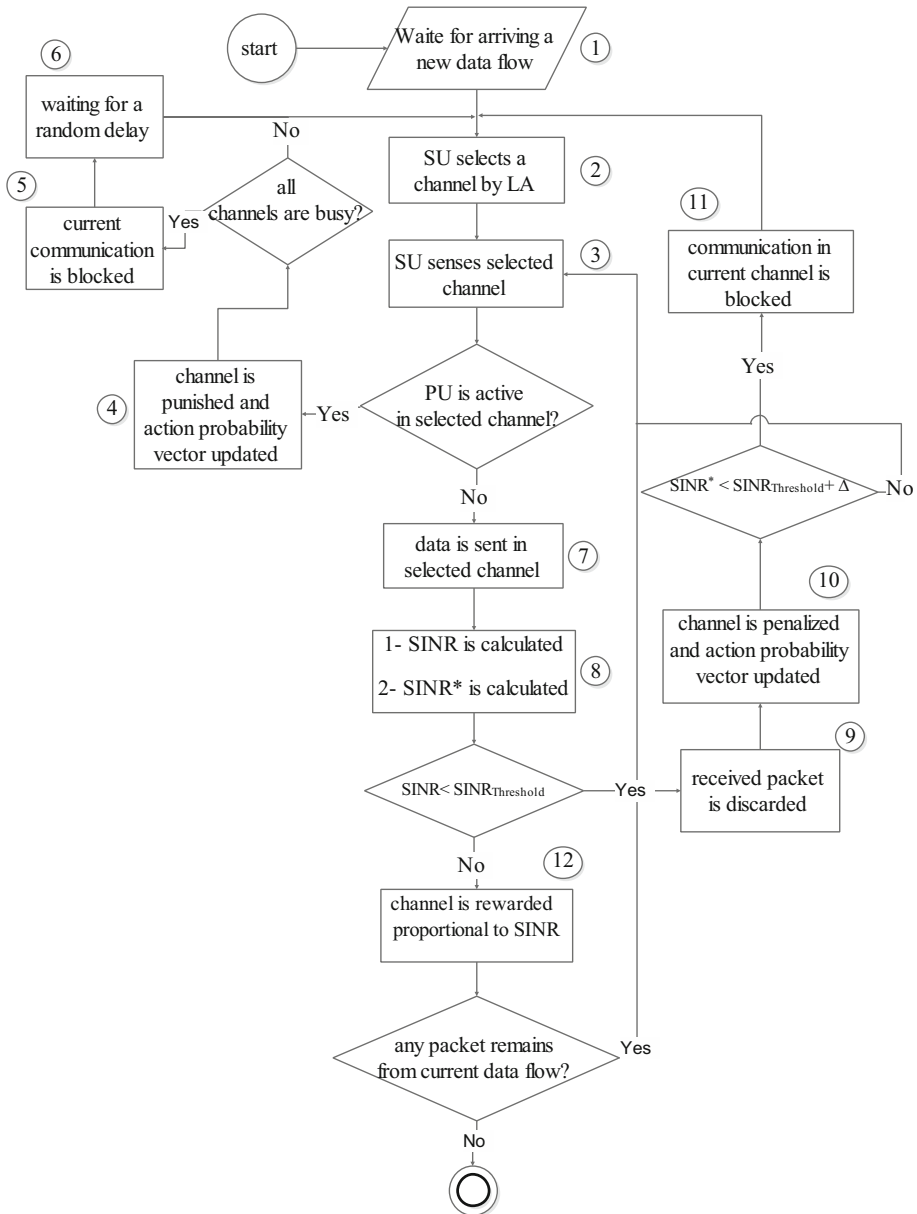


Fig. 6 Flowchart of the first proposed algorithm

### 5.3 DQASA: Second Proposed Algorithm

In this Section, the operation and stages of the second proposed algorithm will be explained. This algorithm is suggested with the aim of reducing the channel switching overhead of the first algorithm. The flowchart of DQASA is depicted in Fig. 8. This algorithm like QASA will be started after receiving the first packet of a new data flow.

Afterwards a new suitable and free channel is selected by LA and packet is sent through it. In the receiver SINR and  $SINR^*$  will be calculated and accordingly, the following scenario will occur:

- In case of  $(SINR < SINR_{Thr})$  and  $(SINR^* \geq SINR_{Thr} + \Delta)$ , the packet is discarded; selected channel is penalized; action probability vector is updated, and communication will continue through the channel.
- In case of  $(SINR < SINR_{Thr})$  and  $[(SINR^* < SINR_{Thr}) \text{ or } [(SINR_{Thr} - \leq SINR^* < SINR_{Thr} + \Delta) \text{ and } (\text{all radios are active})]]$ , the packet is discarded; selected channel is penalized; action probability vector is updated; communication through the used channel is stopped, and SU retries to select a suitable channel.
- In case of  $(SINR < SINR_{Thr})$  and  $(SINR_{Thr} \leq SINR^* < SINR_{Thr} + \Delta)$  and all radios are not active, BR will be chosen randomly among inactive radios; LA selects a new channel; BR senses the selected channel; control information is exchanged between control radios; sending operation through main radio is stopped and communication will be continued by BR.
- In case of  $(SINR \geq SINR_{Thr})$ , used channel is rewarded; action probability vector is updated, and finally communication will continue through the channel.

This algorithm like QASA will be run for any generated data flow in senders and finishes when the last packet of this flow is sent successfully. In Figs. 7 and 8 the corresponding algorithms and the flowchart of DQASA is depicted.

## 6 Evaluation of Performance

Performance of the proposed method is evaluated through extensive simulations of the algorithm using OMNeT++ [46] which is a component-based, open-architecture, modular and discrete event network simulator. In addition, INET-MANET framework feature was considered for simulating multi radio multi channel networks.

Network nodes were deployed on a plane of  $500 * 500 \text{ m}^2$ . The number of SUs is set to  $M = 50$ , that are stationary and distributed uniformly. In the proposed model, 11 channels were considered with one dedicated control channel. The number of PUs is equal to the number of data channels and was set to  $N = 10$ . Every SU was equipped with one control radio and the number of data radios varies between  $1 \leq R \leq 10$ . Simulations run for 100 s.

Extensive simulations carried out under different scenarios and network settings. The results are compared for the following network parameters:

- *Channel switching rate (CSR)*: is the rate of switching caused by degradation of channel quality. This rate is calculated as the total average number of secondary channels switching (except those are due to the presence of PUs), divided by the simulation period.
- *Data delivery ratio (DDR)*: is the ratio of the average amount of data that all the senders send to that of all the receivers receive on the network during the simulation period.
- *Throughput*: is defined as the rate of data received successfully by the MAC layer of all the secondary receivers over the simulation period.
- *Data loss (DL)*: is the difference between the average amount of data sent and average amount of data received.

```

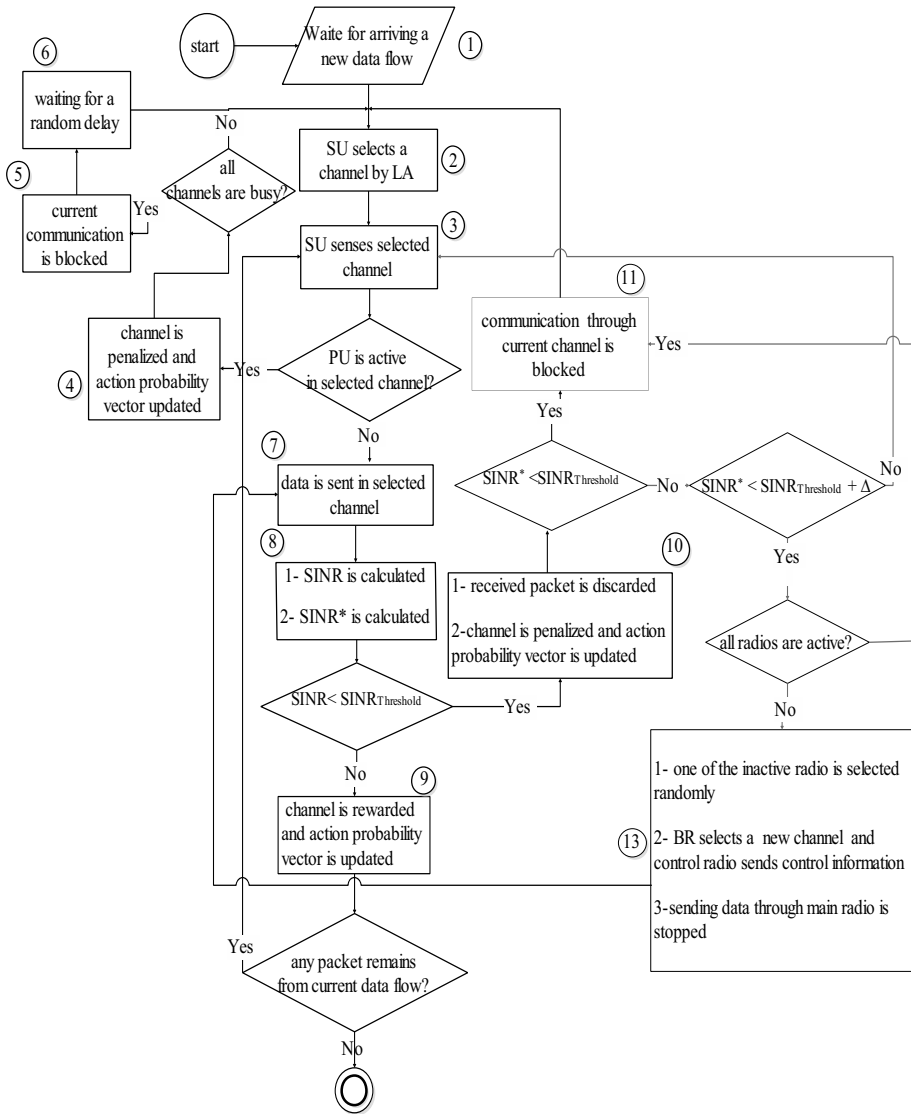
handle_message_DQASA ( msg, SINR*, channel, radio )
{
  if (channel = 0) then
  {
    channel  $\leftarrow$  select a channel by LA Algorithm;
    SINR*  $\leftarrow$  Initialize_SINR();
    radio.channle  $\leftarrow$  channel;           // The radio is tuned on the selected channel
  }
  if (PUs are active in the selected channel) then
  {                                     // The channel must be changed
    Penalize (channel);
    if ( All channels are occupied by PUs) then
      Wait (Random());
    channel  $\leftarrow$  0;
  }
  else
  {
    Send (msg, radio);
    SINR  $\leftarrow$  radio.Get_SINR ();
    SINR*  $\leftarrow$  Calc_SINR* (SINR);
    if (SINR  $\geq$  SINRThreshold) then           // Message is sent successfully
    {
      Reward (channel);
      msg  $\leftarrow$  Next_Packet();           // Next message of data flow is prepared
    }
    else                                     // Message is discard in receiver and must be send again
    {
      Penalize (channel);
      if (SINR* < SINRThreshold ) then           // The quality of selected
        channel  $\leftarrow$  0;           //channel is not suitable and must be changed
      else if (SINR* < SINRThreshold +  $\Delta$ ) then
      {
        if ( there is an inactive radio ) then           //A new radio is selected
        {
          radio  $\leftarrow$  A radio is selected among inactive radios;
          channel  $\leftarrow$  A channelis selected by LA;
          SINR*  $\leftarrow$  Initialize_SINR();
          radio.channle  $\leftarrow$  channel;
        }
      }
    }
  }
  if (msg  $\neq$  NULL) then           //Checks that any packet is remained
    handle_message (msg, SINR*, channel, radio);
}

```

**Fig. 7** Handle message algorithm of DQASA

- *Communication delay (CD)*: is the time between a packet's creation and its delivery. One of the main resources of the CD is the queuing delay, which is the time a packet waits in the queue to be scheduled and is inversely proportional to the DDR. The lower amount of DDR is proportional to more dropped packets. By increasing the number of dropped packets, the queue length and as a result queuing delay will be increased.

DQASA and QASA are evaluated in terms of the aforementioned QoS parameters compared to other representative approaches namely MRLA [40], RND and QRSA (QoS aware Random Spectrum Allocation). In RND like MRLA channel are switched only when a PU appears in it. However, RND selects a new channel randomly. Also, QRSA operates



**Fig. 8** The flowchart of second proposed algorithm Evaluation of performance

like QASA but it selects a new channel randomly. It should be noted that, all of these algorithms are classified into two categories. In the methods of the first category (i.e., QRSA, QASA and DQASA), SU not only switches the channel in the presence of PU, but also changes it when the channel quality is degraded. In the second category (i.e., MRLA and RND), different methods tend to switch the channels after a PU appears.

In what follows, the results of conducted experiments are represented to investigate the performance of two proposed schemes. Three sets of simulations are considered in these experiments. In the first one, the results are represented during first 100 s in which every SU is equipped with 5 radios. The second one presents the results in time instance 100 s in



which the number of radios vary from 1 to 10 per each SU. Finally, in the third set, the cumulative results are shown in time instance 100 s when each SU is equipped with 5 radios.

### 6.1 Experiment 1: Channel Switching Evaluation

This experiment was conducted to evaluate the channel switching rate as well as total number of channel switching in two proposed methods compared to the other methods. It should be noted that, in DQASA when there is a chance to find a BR the channel switching delay is negligible. From the results reported in Fig. 9 in all of compared algorithms, in the beginning of the simulation, the network traffic is light, the quality of the channels is desirable and therefore, the CHR is low. The CHR in different algorithms will be stable around 50 s. The results of Figs. 9 and 10 demonstrate that the rate and total number of channel switching in learning based algorithms are significantly less than other comparative algorithms especially after stability. The reason is that, learning based schemes get a higher selection chance to the channels with successful experiences of transmission in the past epochs. Hence, these algorithms gradually tend to select the channels with better quality and so SUs are not forced to change communication channels frequently. In addition, as seen in Fig. 10, although MRLA provides lower total channel switching than QASA, DQASA not only recovers this weakness (by using BR) but also outperforms the MRLA. This superiority thanks to better decision making metric in DQASA (SINR) in contrary to the decision making metric in MRLA (presence or absence of other SUs in channel). By increasing the number of radios per each SU, the transmission capacity, number of channel switching and the network traffic will be increased. With the enhancement of the radios, DQASA has a higher chance to find a BR and as a result, the number of channel switching that are due to the channel quality degradation will be decreased and gradually will be converged to 0. That is why the results of Fig. 11 shows that increasing the number of radios increases the number of channels switching in all algorithms, except in DQASA.

### 6.2 Experiment 2: Throughput Evaluation

The objective of this experiment is to assess the resultant throughput and sent data rate of two proposed algorithms compared to the other algorithms. From the Fig. 12, the gap between the chart associated with these two parameters in the methods of the second

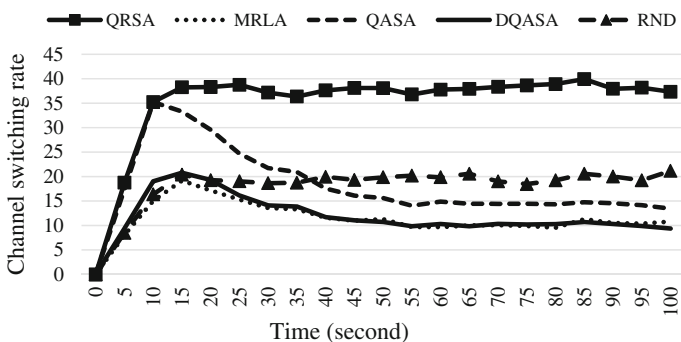
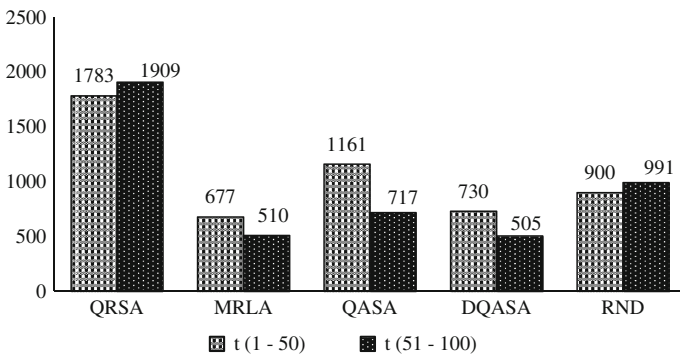


Fig. 9 Channel switching rate versus time (R = 5)

aforementioned category is larger than that of other methods. This large gap is due to the large number of dropped packets caused by the communication through bad quality channels in the methods of this category. In addition, DQASA shows higher throughput than QASA. This superiority thanks to higher sent data rate due to the lower switching delay in DQASA. From the results reported in Fig. 13 both proposed methods provide more attained throughput compared to the other approaches. This superiority is due to channel switching in channel degradation quality times that guarantees the quality of received packet (in comparison with RND and MRLA) and employing LA (in comparison with RND and QRSA). As shown in Fig. 14, by increasing the number of radios, DQASA shows the better attained throughput than other methods. The reason is that, by increasing the number of radios per each SU, increasing the BRs activity leads to a significant degradation in blocking and dropping probability in the DQASA. Therefore, the greater amount of data will be successfully received at the destination.

### 6.3 Experiment 3: Data Delivery Ratio (DDR) Evaluation

In this experiment, the aim is to evaluate the attained DDR in proposed algorithms in comparison with other algorithms. As it can be seen in Fig. 15, in the beginning of the simulation, DDR in all compared methods are close to each other. But in the methods of the first category DDR increases and converges to 1 gradually. The reason is that, In the methods of this category, when the quality of channel degrade, communication via this channel will be spotted and continued through a suitable channel and so the higher percentage of the sent data are received successfully at the destination. By increasing the number of radios the traffic in the methods of the first category, will be better distributed among the channels and so the quality of network communications will be enhanced. Therefore, DDR will be increased and converged to 1. But in the methods of the second category, by increasing the network traffic due to utilization of the more radios per each SU, the channels quality and as a result DDR will be decreased. That is why the results of Fig. 16 depict the ascending trend in the methods of the first category in contrary to the descending trend in other methods. In addition the superiority of two proposed methods in comparison with QRSA in Fig. 16b thanks to inherent characteristics of LA, that leads to selecting the channels with higher chance of successful communication even after enhancement of the traffic of the network.



**Fig. 10** Total number of channel switching in Seconds between (1–50) and between (51–100) (R = 5)

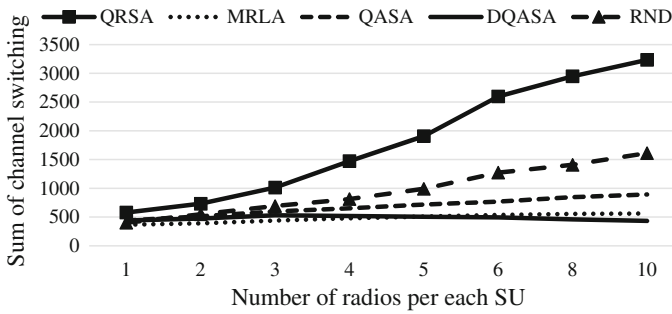


Fig. 11 Total number of channels switching after stability versus number of radios per each SU

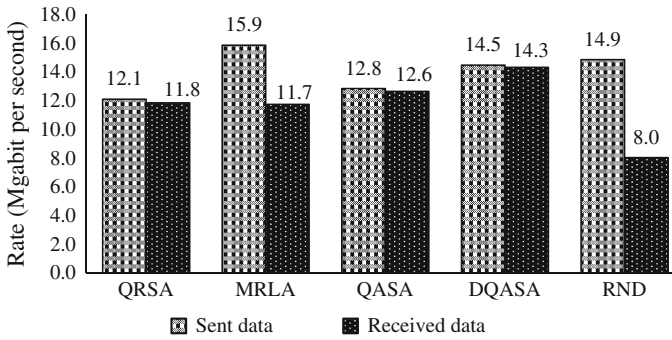


Fig. 12 The rate of the sent and received (throughput) data (R = 5)

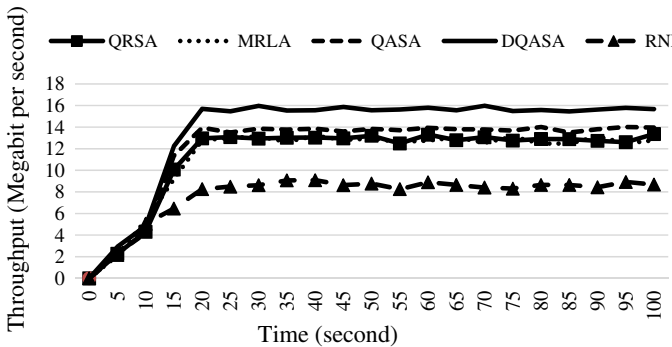


Fig. 13 Throughput versus time (R = 5)

### 6.4 Experiment 4: Data Lost (DL) Evaluation

This experiment is mainly conducted to evaluate the achieved DL in proposed methods compared to the other methods. It is clear from Fig. 17 that, the total DL in the first category is significantly lower than second category methods. This large difference is because, in the methods of the second category, when the packet is dropped due to the channel’s quality degradation SU tries to resend this packet, again through the current bad

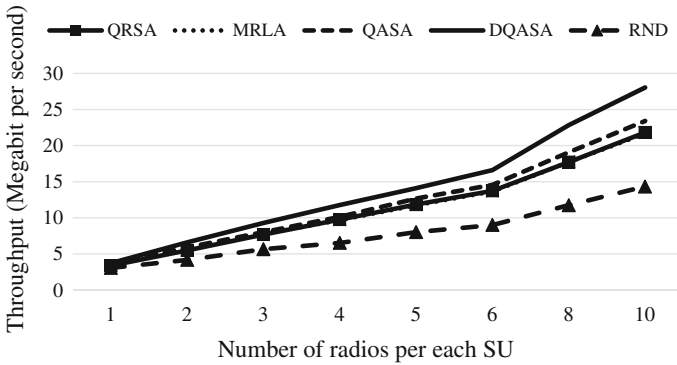


Fig. 14 Throughput versus number of radios per each SU

quality channel. As a result, this channel will become more congested, and therefore, data loss will be increased. This happens while, many high-quality channels may be available in the network. In contrary to the methods of the first category, lost packets will be resent via a new suitable channel and so, will be successfully delivered with higher probability. similarly, as seen in Fig. 18a, b by increasing the number of radios per each SU in the methods of the first category, packets will be better divided between channels. Hence, not only the amount of lost data will be degraded, but also, resent packets will be delivered with higher chance.

### 6.5 Experiment 5: Delay Evaluation

In this experiment the objective is to study the achieved delay in proposed methods compared to the other methods. By comparing Figs. 9, 15 and 19 the following points can be found:

- Although MRLA and DQASA have almost the same channel switching curves, DQASA shows lower delay than MRLA. This superiority thanks to higher DDR in the DQASA that leads to lower queue length and delay.
- Despite QASA shows higher channel switching rate than MRLA because of the higher DDR, QASA has the less average delay than MRLA.

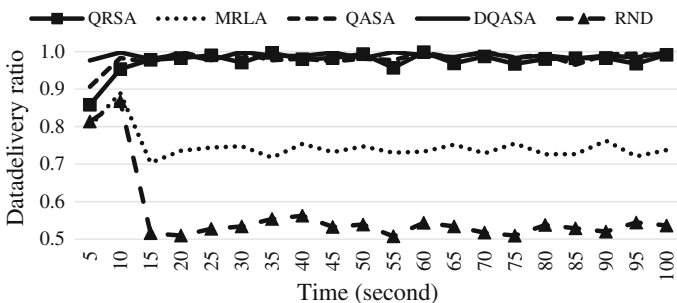
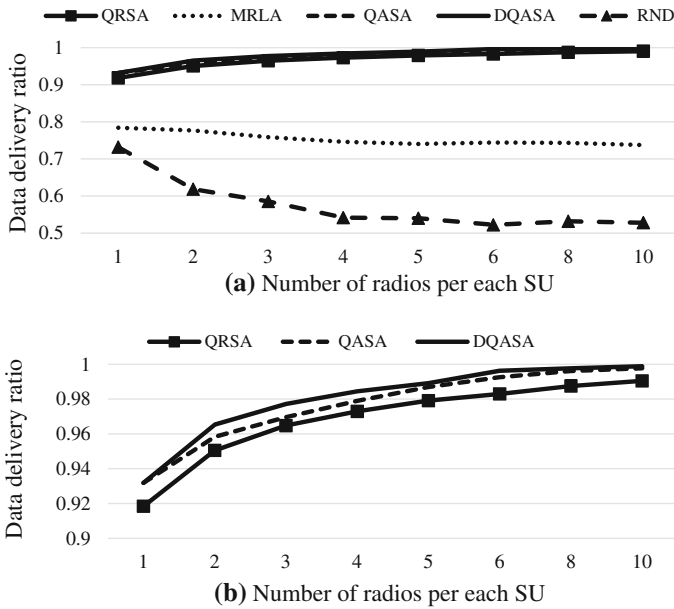
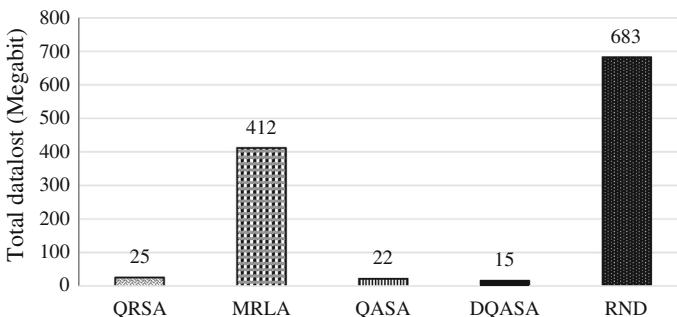


Fig. 15 Data delivery ratio versus Time (R = 5)

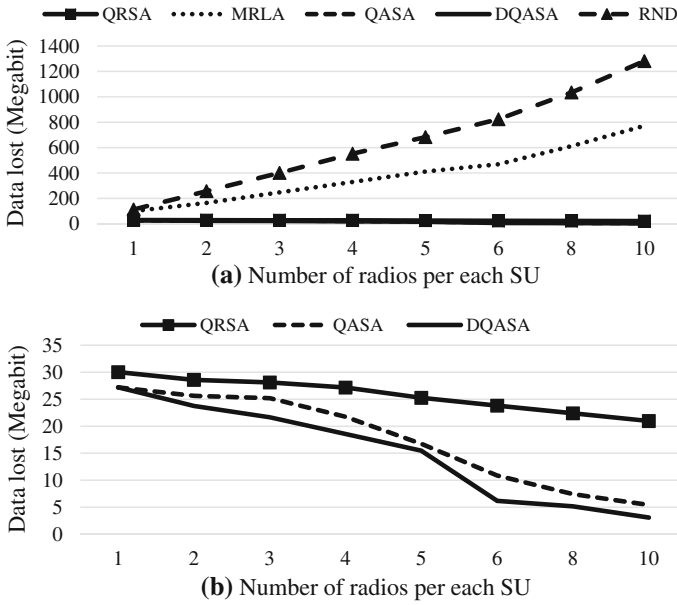


**Fig. 16** Data delivery ratio versus Number of radios in SU **a** all of methods, **b** the first category methods (is shown for more clarity)

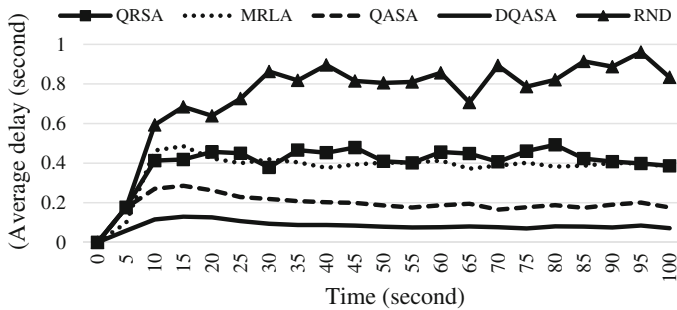
As shown in Fig. 20, by increasing the number of radios and transmission capacity, lower data will be waited in the queue and hence, the queuing delay and as a result average delay of packets will be decreased in all methods. In addition, In DQASA with 6, 8 and 10 radios average delay will not perform any important change and is almost close to 0. This indicates that when SU is equipped with 6 radios (and more), the channel switching and queuing delays will be minimized (almost zero) and only slightly propagation delay will be occurred. By increasing the number of radios the chance of finding a BR in DQASA will be enhanced. Hence, channel switching overhead will decrease significantly because of parallelizing the operation of channel switching and data sending. That is why the distance between the average delay of DQASA and QASA will be increased by increasing the number of radios.



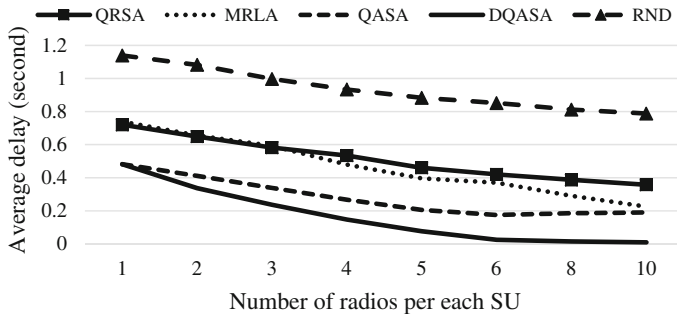
**Fig. 17** Total data lost in the 100 s (R = 5)



**Fig. 18** Data lost versus Number of radios per each SU **a** for all of methods **b** for the first category methods (is shown for more clarity)



**Fig. 19** Average delay of any packet



**Fig. 20** Average delay of any packet versus number of radios per each SU

## 7 Conclusion and Future Works

In this paper, the problem of channel allocation in dynamic cognitive networks was investigated, which is known as NP-complete. Most of the previous works only have tried to reduce the overhead of channel tuning by reducing the number of switching through selecting the suitable channels for communication. In this paper, two fully distributed learning automata based channel allocation algorithms were proposed. In the first proposed algorithm, it is tried to satisfy the QoS requirements by prevention and avoidance of communication via bad-quality channels. In the second algorithm, with the aim of reducing the cost of channel switching, the idle radios of SUs have been utilized to parallelize the channel switching and data transmission operations. Simulation results demonstrate that two proposed schemes outperform MRLA [40], RNA and QRSA in terms of QoS parameters. In addition, the second proposed method can significantly overcome the time overhead and interruptions caused by channel switching. Some of the open problems in this research can be listed as follows: It would be interesting to utilize the BR for reducing the overhead of channels switching that are due to the presence of PUs. It is also worthwhile to study about the influence of utilizing the BR on power consumption in the networks. And finally, It is beneficial to extend proposed work for multi hop networks with mobile users.

## References

1. Marcus, M., Burtle, J., Franca, B., Lahjouji, A., & McNeil, N. (2002). Federal communications commission spectrum policy task force. *Report of the unlicensed devices and experimental licenses working group*.
2. Mitola, J., & Maguire, G. Q., Jr. (1999). Cognitive radio: Making software radios more personal. *IEEE Personal Communications*, 6, 13–18.
3. Thomas, R. (2007). *Cognitive networks*. Ph.D. Dissertation, Computer Engineering, Virginia Polytechnic Institute and State University.
4. Karp, R. M. (1972). *Reducibility among combinatorial problems* (pp. 85–103). New York: Springer.
5. Ren, P., Wang, Y., Du, Q., & Xu, J. (2012). A survey on dynamic spectrum access protocols for distributed cognitive wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2012, 60.
6. Plummer, A., & Biswas, S. (2011). Distributed spectrum assignment for cognitive networks with heterogeneous spectrum opportunities. *Wireless Communications and Mobile Computing*, 11, 1239–1253.
7. Hoang, A. T., & Liang, Y.-C. (2008). Downlink channel assignment and power control for cognitive radio networks. *IEEE Transactions on Wireless Communications*, 7, 3106–3117.
8. Su, X., Yuan, C., & Shen, S. (2009). A new mechanism of dynamic spectrum allocation in the cognitive network. In *5th international conference on wireless communications, networking and mobile computing (WiCom'09)* (pp. 1–4).
9. Zhang, T., Wang, B., & Wu, Z. (2009). Spectrum assignment in infrastructure based cognitive radio networks. In *Proceedings of the IEEE national aerospace and electronics conference (NAECON)* (pp. 69–74).
10. Yang, L., Xie, X., & Zheng, Y. (2009). A historical-information-based algorithm in dynamic spectrum allocation. In *International conference on communication software and networks (ICCSN'09)* (pp. 731–736).
11. Bkassiny, M., & Jayaweera, S. K. (2010). Optimal channel and power allocation for secondary users in cooperative cognitive radio networks. In P. Chatzimisios, C. Verikoukis, I. Santamaría, M. Laddomada, O. Hoffmann (Eds.) *Mobile lightweight wireless systems*, vol 45, (pp. 180–191). Berlin, Heidelberg: Springer.
12. Hoang, A. T., Liang, Y.-C., & Islam, M. H. (2007). Maximizing throughput of cognitive radio networks with limited primary users' cooperation. In *ICC'07. IEEE international conference on communications, 2007* (pp. 5177–5182).

13. Chen, S., Huang, Y., & Namuduri, K. (2011). A factor graph based dynamic spectrum allocation approach for cognitive network. In *Wireless communications and networking conference (WCNC), 2011 IEEE* (pp. 850–855).
14. Xu, Y., Lui, J. C., & Chiu, D.-M. (2010). On oligopoly spectrum allocation game in cognitive radio networks with capacity constraints. *Computer Networks*, *54*, 925–943.
15. Chen, L., Iellamo, S., Coupechoux, M., Godlewski, P. (2010). An auction framework for spectrum allocation with interference constraint in cognitive radio networks. In *INFOCOM, 2010 proceedings IEEE* (pp. 1–9).
16. Teng, Y., Yu, F. R., Han, K., Wei, Y., & Zhang, Y. (2013). Reinforcement-learning-based double auction design for dynamic spectrum access in cognitive radio networks. *Wireless Personal Communications*, *69*, 771–791.
17. Saad, W., Han, Z., Zheng, R., Hjørungnes, A., Başar, T., & Poor, H. V. (2012). Coalitional games in partition form for joint spectrum sensing and access in cognitive radio networks. *IEEE Journal of Selected Topics in Signal Processing*, *6*, 195–209.
18. Salameh, B., & Ahmad, H. (2011). Throughput-oriented channel assignment for opportunistic spectrum access networks. *Mathematical and Computer Modelling*, *53*, 2108–2118.
19. Salameh, H. B. (2010). Rate-maximization channel assignment scheme for cognitive radio networks. In *Global telecommunications conference (GLOBECOM 2010), 2010 IEEE* (pp. 1–5).
20. Tragos, E. Z., Zeadally, S., Fragkiadakis, A. G., & Siris, V. A. (2013). Spectrum assignment in cognitive radio networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials*, *15*, 1108–1135.
21. Lee, D. H., Jeon, W. S., & Jeong, D. G. (2010). Joint channel assignment and routing in cognitive radio-based wireless mesh networks. In *2010 IEEE 71st vehicular technology conference (VTC)* (pp 1–5).
22. Wang, W., Kasiri, B., Cai, J., & Alfa, A. S. (2011). Channel assignment of cooperative spectrum sensing in multi-channel cognitive radio networks. In *2011 IEEE international conference on communications (ICC)* (pp. 1–5).
23. Salameh, H. B., Krunz, M., & Younis, O. (2008). Distance-and traffic-aware channel assignment in cognitive radio networks. In *5th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks, 2008. SECON'08* (pp. 10–18).
24. Kaur, P., Uddin, M., & Khosla, A. (2010). Adaptive bandwidth allocation scheme for cognitive radios. *International Journal of Advancements in Computing Technology*, *2*, 35–41.
25. Wenzhu, Z., & Xuchen, L. (2011). Centralized dynamic spectrum allocation in cognitive radio networks based on fuzzy logic and Q-learning. *China Communications*, *8*, 46–54.
26. Ye, F., Yang, R., & Li, Y. (2011). Genetic spectrum assignment model with constraints in cognitive radio networks. *International Journal of Computer Network and Information Security (IJCNIS)*, *3*, 39–45.
27. Zhao, Z., Peng, Z., Zheng, S., & Shang, J. (2009). Cognitive radio spectrum allocation using evolutionary algorithms. *IEEE Transactions on Wireless Communications*, *8*, 4421–4425.
28. Ye, F., Yang, R., & Li, Y. (2010). Genetic algorithm based spectrum assignment model in cognitive radio networks. In *2nd international conference on information engineering and computer science (ICIECS)* (pp. 1–4).
29. Del Ser, J., Matinmikko, M., Gil-López, S., & Mustonen, M. (2012). Centralized and distributed spectrum channel assignment in cognitive wireless networks: A harmony search approach. *Applied Soft Computing*, *12*, 921–930.
30. Cheng, X., & Jiang, M. (2011). Cognitive radio spectrum assignment based on artificial bee colony algorithm. In *13th IEEE international conference on communication technology (ICCT)* (pp. 161–164).
31. Ghasemi, A., Masnadi-shirazi, M., Biguesh, M., Jahromi, A. F., & Ghasemi, F. (2012). Spectrum allocation based on artificial bee colony in cognitive radio networks. In *2012 sixth international symposium on telecommunications (IST)* (pp. 182–187).
32. Jiang, T., Grace, D., & Mitchell, P. D. (2011). Efficient exploration in reinforcement learning-based cognitive radio spectrum sharing. *IET Communications*, *5*, 1309–1317.
33. Faganello, L. R., Kunst, R., Both, C. B., Granville, L. Z., & Rochol, J. (2013). Improving reinforcement learning algorithms for dynamic spectrum allocation in cognitive sensor networks. In *IEEE wireless communications and networking conference (WCNC)* (pp. 35–40).
34. Vucevic, N., Akyildiz, I. F., & Pérez-Romero, J. (2010). Cooperation reliability based on reinforcement learning for cognitive radio networks. In *Fifth IEEE workshop on networking technologies for software defined radio (SDR) networks* (pp. 1–6).
35. Jiang, T., Grace, D., & Liu, Y. (2011). Two-stage reinforcement-learning-based cognitive radio with exploration control. *IET Communications*, *5*, 644–651.
36. Song, Y., Fang, Y., & Zhang, Y. (2007). Stochastic channel selection in cognitive radio networks. In *IEEE global telecommunications conference (GLOBECOM'07)* (pp. 4878–4882).



37. 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 (ICIS)* (pp. 294–298).
38. Tuan, T. A., Tong, L. C., & Premkumar, A. (2010). An adaptive learning automata algorithm for channel selection in cognitive radio network. In *International conference on communications and mobile computing (CMC)* (pp. 159–163).
39. Liu, L., Hu, G., Xu, M., & Peng, Y. (2010). Learning automata based spectrum allocation in cognitive networks. In *IEEE international conference on wireless communications, networking and information security (WCNIS)* (pp. 503–508).
40. Bizhani, H., & Ghasemi, A. (2013). Joint admission control and channel selection based on multi response learning automata (MRLA) in cognitive radio networks. *Wireless Personal Communications, 71*, 629–649.
41. Narendra, K. S., & Thathachar, M. A. (2012). *Learning automata: An introduction*. North Chelmsford: Courier Corporation.
42. Torkestani, J. A., & Meybodi, M. R. (2011). A learning automata-based cognitive radio for clustered wireless ad-hoc networks. *Journal of Network and Systems Management, 19*, 278–297.
43. Narendra, K. S., & Thathachar, M. (1974). Learning automata—A survey. *IEEE Transactions on Systems, Man and Cybernetics, 4*, 323–334.
44. Maulik, S., Roy, R., De, A., & Bhattacharya, A. (2012). Online dynamic resource allocation in interference temperature constrained cognitive radio network using reinforcement learning. In *International conference on signal processing and communications (SPCOM)* (pp. 1–5).
45. Zhang, W., & Yeo, C. K. (2014). Sequential sensing based spectrum handoff in cognitive radio networks with multiple users. *Computer Networks, 58*, 87–98.
46. <http://www.omnetpp.org>.



**Setare Rasaneh** received the B.S. degree in from Islamic Azad University of South Tehran Branch and her M.Sc. degree from Islamic Azad University of Central Tehran Branch, in 2009 and 2014 respectively both in Computer Software Engineering. Her research interests include reinforcement learning, routing and channel allocation in cognitive and sensor networks.



**Mohsen Jahanshahi** completed his B.S. and M.S. studies in Computer Engineering in Iran, dated 2002 and 2005 respectively. He also achieved a Ph.D. degree in Computer Engineering from Islamic Azad University (Tehran Science and Research Branch) in 2011. He joined the faculty of Computer Engineering Department at Islamic Azad University (Central Tehran Branch) in 2005. His research interests include performance evaluation, multistage interconnection networks, wireless mesh networks, wireless sensor networks, cognitive networks, learning systems, mathematical optimization, and soft computing.