A Learning Automata-Based Cognitive Radio for Clustered Wireless Ad-Hoc Networks

Javad Akbari Torkestani · Mohammad Reza Meybodi

Abstract In current wireless networks, the radio systems are regulated by a fixed spectrum assignment strategy. This policy partitions the whole radio spectrum into a fixed number of radio ranges, each exclusively assigned to a specific user. Such a spectrum assignment strategy leads to an undesirable condition under which some systems only use a small portion of the allocated spectrum while the others have very serious spectrum insufficiency. The learning automata-based cognitive radio which is proposed in this paper is a highly potential technology to address the spectrum scarcity challenges in wireless ad hoc networks. This paper proposes a learning automata-based dynamic frame length TDMA scheme for slot assignment in clustered wireless ad-hoc networks with unknown traffic parameters, where the intra-cluster communications are scheduled by a TDMA scheme, and a CDMA scheme is overlaid on the TDMA to handle an interference-free inter-cluster communication. In this method, each cluster-head is responsible for a collision-free slot assignment within the cluster and determines the input traffic parameters of its own cluster members. It then takes these traffic parameters into consideration for an optimal channel access scheduling in the cluster. The medium access control layer in each cluster is based on a time division multiple access (TDMA) scheme, in which each host is assigned a fraction of the TDMA frame proportional to its traffic load. The simulation experiments show the superiority of our proposed slot

J. Akbari Torkestani (✉)
Department of Computer Engineering, Islamic Azad University, Arak Branch, Arak, Iran
e-mail: j-akbari@iau-arak.ac.ir

M. R. Meybodi
Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran
e-mail: mmeybodi@aut.ac.ir

M. R. Meybodi
Institute for Studies in Theoretical Physics and Mathematics (IPM), School of Computer Science, Tehran, Iran

assignment algorithm over the existing methods in terms of the channel utilization, control overhead, and throughput, specifically, under bursty traffic conditions.

1 Introduction

Recent advances in wireless communication technology have provided an opportunity to develop a new approach of intelligent or cognitive radios in which the radio frequency spectrum can be adaptively distributed among the users proportional to their needs. That is, the cognitive radio is an emergent paradigm to address the spectrum allocation strategy issues in wireless networks in which the wireless nodes are capable of changing their transmission or reception parameters to communicate efficiently without interference. A cognitive radio system supports a very dynamic MAC layer adaptation based on the active monitoring of available channel bandwidth resource. The following are the main functions of a cognitive radio. (1) Exploring the unused ranges of the radio spectrum and sharing them with the other users avoiding interference and collision. (2) Selecting the best available spectrum to meet the system constraints and user requirements. (3) Supporting the user connection requests even if it exchanges the frequency of operation. (4) Providing the fair collision-free spectrum scheduling method. However, cognitive radio is still in the very early stage of the research and development. This paper aims at discussing the MAC layer challenging issues of cognitive radios and proposing an intelligent channel access scheduling mechanism for cognitive radios [1–3].

Numerous ways have been proposed to organize the MAC layer. The MAC protocols can be divided into fixed assignment, demand-assignment, and contention-access protocols. Fixed-assignment protocols are those for which, as the name implies, channel assignments are fixed, regardless of the transmission requirements. Frequency division multiple access (FDMA) [4], code division multiple access (CDMA) [5], and time division multiple access (TDMA) [6] schemes are some fixed-assignment MAC layer protocols. Among the controlled access MAC protocols, TDMA is the most commonly used in wireless ad-hoc networks. Demand-assignment protocols like polling [7], trunking [8] and reservation [9] methods schedule the channel access based on the demand of the hosts for packet transmission. Both fixed-assignment and demand-assignment protocols are collision-free. These protocols are also referred to as contention-free protocols since the hosts do not compete to seize the channel. In contention-access protocols, the hosts contend for channel access, and the hosts that lose it try again later. Since the collisions are not prohibited by the contention-access protocols, they require a method for detecting and recovering the collisions. ALOHA [10] and carrier sense multiple-access (CSMA) [11] are some well-known contention-access protocols.

In a TDMA scheme, the time is shared and access to the common channel is divided among several hosts by allowing each host to use the channel periodically but only for a small period of time referred to as time slot [12–14]. In other words,

TDMA is a round robin-like technique to access many hosts to the common channel. During the time slot, the entire bandwidth is available to the host which is permitted to access the channel. Kanzaki et al. [15] proposed a dynamic TDMA slot assignment protocol called DTSA to improve the channel utilization in ad-hoc networks. The frame length is assumed as a power of two, and the proposed protocol controls the excessive increase of the unassigned slots by changing the frame length dynamically. This method uses a pre-planned channel assignment technique in which a time slot is pre-assigned to each node. It assigns one of the unassigned slots to the new arrived nodes, or generates unassigned slots by depriving one of the multiple slots assigned to a node, if there are no unassigned slots available. Since the pre-assigned slots are not released when the node has no data transmission, the proposed protocol reduces the channel spatial reusability. Furthermore, the proposed protocol cannot support more slots requested by a node dealing with burst traffic. In [16], a dynamic TDMA frame length expansion and recovery method called dynamic frame length channel assignment (DFLCA) was proposed. The proposed method, taking advantage of the channel spatial reuse concept, efficiently utilizes the channel bandwidth by assigning the unused slots to the new nodes as well as enlarging the frame length when the number of slots in the frame is insufficient to support the nodes. DFLCA controls the expansion and recovery of unassigned time slots by dynamically changing the frame length according to the traffic load and the number of mobile nodes in the contention area. For this purpose, the nodes are allowed to release the unused slots and shrink their channel tables when the frame is inefficient. In [17], Li et al. proposed an evolutionary dynamic TDMA slot assignment protocol for ad-hoc networks. In this slot management method, the frame length and transmission schedule are dynamically updated according to the topology density of network and bandwidth requirement. This protocol allows the transmitter to reserve one or more unscheduled slots from the set of unassigned slots. Mo and Chew [18] proposed a new TDMA scheme in which the time slot duration is not fixed and vary with time for each user. In this method, the time slot duration is independently adjusted for each user proportional to its transmission requirements. In this paper, it is shown that the proposed variable frame length TDMA scheme improves the bandwidth utilization.

In CDMA scheme, a unique code is assigned to each connection and this is a trivial problem if the network size is small. But, when a CDMA scheme is employed in a large multi-hop ad-hoc network, the code assignment becomes an intractable problem. One promising approach to solve the code assignment problem is using the overlaid CDMA/TDMA scheme [19–22]. Many studies have been carried out on CDMA/TDMA scheme in cellular networks [23–26], while in ad-hoc networks it has not received the attention it deserves. To design an overlaid CDMA/TDMA structure, three following issues must be considered. First, grouping the hosts into a number of non-overlapping clusters. Second, supporting the inter-cluster connections by assigning a code to each cluster (code assignment) so that no two neighboring or hidden clusters have the same code, and third, using an efficient TDMA scheme for channel access scheduling within each cluster. Code assignment problem is very similar to the NP-hard [27] vertex coloring problem [28] in graph theory. Besides the above mentioned TDMA schemes, we compare our proposed

model with the TDMA part of the following CDMA/TDMA schemes. In [21], Akaiwa and Andoh proposed a dynamic channel assignment scheme called CS-DCA based on the channel segregation method introduced by Furuya and Akaiwa [19] to improve the spectrum efficiency. In channel segregation scheme, the channels are shared and dynamically assigned to the neighboring cells. In this method, for each cell, a dynamic priority is associated with every channel. When a call arrives, the channel with the highest priority is selected. If the channel is in use by the neighboring cells, the priority of the selected channel decreases and the next highest priority channel is selected. Otherwise, the selected channel is assigned to the call and the priority of the channel increases. In [22], a CDMA/TDMA structure was proposed by Wu for clustered wireless ad-hoc networks. He designed a dynamic channel assignment algorithm called Hybrid-DCA to make the best use of available channels by taking advantage of the spatial reuse concept. In this approach, the TDMA scheme is overlaid on top of the CDMA scheme to divide the bandwidth into smaller chunks. The proposed DCA algorithm forms the channel as a particular time slot of a particular code.

The main problem with the above mentioned channel assignment schemes is that they assume the input traffic is fixed or a stationary process with known parameters. Since in ad hoc networks, the parameters of the input traffic are unknown and possibly time varying, in this paper, we propose an adaptive learning automata-based slot assignment algorithm for clustered wireless ad-hoc networks when the input traffic parameters are unknown. In this method, the wireless hosts are grouped into clusters and each cluster-head takes the responsibility of a collision-free channel access scheduling within the cluster. To do this, each cluster-head is equipped with a learning automaton whose action-set includes an action for each of its cluster members. At each stage, cluster-head randomly chooses one of its actions according to its action probability vector. Then, the cluster member corresponding to the selected action is permitted to transmit its packets during the current time slot. If the selected member has a packet to transmit, the cluster-head rewards the selected action, and penalizes it otherwise. As the proposed algorithm proceeds, the probability of choosing a given host converges to the proportion of time it has a packet to transmit. This probability specifies the fraction of TDMA frame must be assigned to the host. Simulation results show that our proposed method significantly outperforms the existing TDMA-based channel assignment protocols and CDMA/TDMA schemes in terms of the channel utilization, control overhead, and throughput, specifically, under the bursty traffic conditions.

The rest of the paper is organized as follows. The next section introduces the learning automata theory and Sect. 3 proposes an adaptive dynamic frame length TDMA scheme. Section 4 shows the efficiency of the proposed algorithm through the simulation experiments, and Sect. 5 concludes the paper.

2 Learning Automata

A learning automaton [29–33] is an adaptive decision-making unit that improves its performance by learning how to choose the optimal action from a finite set of

allowed actions through repeated interactions with a random environment. The action is chosen at random based on a probability distribution kept over the action-set and at each instant the given action is served as the input to the random environment. The environment responds the taken action in turn with a reinforcement signal. The action probability vector is updated based on the reinforcement feedback from the environment. The objective of a learning automaton is to find the optimal action from the action-set so that the average penalty received from the environment is minimized.

Learning automaton is proved to perform well in complex, dynamic and random environments with a large amount of uncertainties. This probabilistic learning model is believed to be a highly efficient tool in deal with unknown environments where no or incomplete information about the environment exists. A group of learning automata can cooperate to cope with many hard-to-solve problems. To name just a few, learning automata have a wide variety of applications in combinatorial optimization problems [34–36], computer networks [37–41], queuing theory [42], signal processing [43], information retrieval [44], adaptive control [45], and pattern recognition [46].

The environment can be described by a triple $E \equiv \{\alpha, \beta, c\}$, where $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents the finite set of the inputs, $\beta \equiv \{\beta_1, \beta_2, \ldots, \beta_m\}$ denotes the set of the values that can be taken by the reinforcement signal, and $c \equiv \{c_1, c_2, \ldots, c_r\}$ denotes the set of the penalty probabilities, where the element c_i is associated with the given action α_i. If the penalty probabilities are constant, the random environment is said to be a stationary random environment, and if they vary with time, the environment is called a non stationary environment. The environments depending on the nature of the reinforcement signal $\underline{\beta}$ can be classified into P-model, Q-model and S-model. The environments in which the reinforcement signal can only take two binary values 0 and 1 are referred to as P-model environments. Another class of the environment allows a finite number of the values in the interval [0, 1] can be taken by the reinforcement signal. Such an environment is referred to as Q-model environment. In S-model environments, the reinforcement signal lies in the interval [a, b]. Figure 1 shows the relationship between a learning automaton and its random environment. From this figure it can be observed that an action ($\alpha(n)$) is randomly chosen by the learning automaton, it is applied to the environment, random environment evaluates the selected action and emits a response (reinforcement signal $\beta(n)$), and automaton updates its state based on the received response.

Learning automata can be classified into two main families [29]: fixed structure learning automata and variable structure learning automata. Variable structure learning automata are represented by a triple $<\underline{\beta} \underline{\alpha}, T>$, where $\underline{\beta}$ is the set of

Fig. 1 The relationship between the learning automaton and its random environment
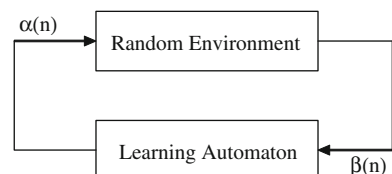
inputs, $\underline{\alpha}$ is the set of actions, and T is learning algorithm. The learning algorithm is a recurrence relation which is used to modify the action probability vector. Let $\alpha_i(k) \in \underline{\alpha}$ and $\underline{p}(k)$ denote the action selected by learning automaton and the probability vector defined over the action set at instant k, respectively. Let a and b denote the reward and penalty parameters and determine the amount of increases and decreases of the action probabilities, respectively. Let r be the number of actions that can be taken by learning automaton. At each instant k, the action probability vector $\underline{p}(k)$ is updated by the linear learning algorithm given in (1), if the selected action $\alpha_i(k)$ is rewarded by the random environment, and it is updated as given in (2) if the taken action is penalized.

$$p_j(k+1) = \begin{cases} p_j(k) + a[1 - p_j(k)] & j = i \\ (1-a)p_j(k) & \forall j \neq i \end{cases} \tag{1}$$

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j = i \\ \left(\frac{b}{r-1}\right) + (1-b)p_j(k) & \forall j \neq i \end{cases} \tag{2}$$

If $a = b$, the recurrence (1) and (2) are called linear reward-penalty (L_{R-P}) algorithm, if $a \gg b$ the given equations are called linear reward-$\in$ penalty ($L_{R-\in P}$), and finally if $b = 0$ they are called linear reward-Inaction (L_{R-I}). In the latter case, the action probability vector remains unchanged when the taken action is penalized by the environment.

2.1 Variable Action-Set Learning Automata

A variable action-set learning automaton is an automaton in which the number of actions available at each instant changes with time. It has been shown in [31, 47] that a learning automaton with a changing number of actions is absolutely expedient and also $\in$-optimal, when the reinforcement scheme is (L_{R-I}). Such an automaton has a finite set of actions, $\underline{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_r\}$. $A = \{A_1, A_2, ..., A_m\}$ denotes the set of action subsets and $A(k) \subseteq \alpha$ is the subset of all the actions can be chosen by the learning automaton, at each instant k. The selection of the particular action subsets is randomly made by an external agency according to the probability distribution $\psi(k) = \{\psi_1(k), \psi_2(k), ..., \psi_m(k)\}$ defined over the possible subsets of the actions, where $\psi_i(k) = \text{prob}[A(k) = A_i | A_i \in A, 1 \leq i \leq 2^n - 1]$.

$\hat{p}_i(k) = \text{prob}[\alpha(k) = \alpha_i | A(k), \alpha_i \in A(k)]$ denotes the probability of choosing action α_i, conditioned on the event that the action subset $A(k)$ has already been selected and $\alpha_i \in A(k)$ too. The scaled probability $\hat{p}_i(k)$ is defined as

$$\hat{p}_i(k) = \frac{p_i(k)}{K(k)} \tag{3}$$

where $K(k) = \sum_{\alpha_i \in A(k)} p_i(k)$ is the sum of the probabilities of the actions in subset $A(k)$, and $p_i(k) = \text{prob}[\alpha(k) = \alpha_i]$.

The procedure of choosing an action and updating the action probabilities in a variable action-set learning automaton can be described as follows. Let $A(k)$ be the action subset selected at instant n. Before choosing an action, the probabilities of all the actions in the selected subset are scaled as defined in (3). The automaton then

randomly selects one of its possible actions according to the scaled action probability vector $\hat{p}(k)$. Depending on the response received from the environment, the learning automaton updates its scaled action probability vector. Note that the probability of the available actions is only updated. Finally, the probability vector of the actions of the chosen subset is rescaled as $p_i(k+1) = \hat{p}_i(k+1) \cdot K(k)$, for all $\alpha_i \in A(k)$. The absolute expediency and ε-optimality of the method described above have been proved in [31].

3 The Proposed Channel Assignment Algorithm

As described earlier, this paper taking advantage of learning automata proposes a dynamic frame length TDMA scheme called LA-TDMA for slot assignment in clustered wireless ad-hoc networks, where the intra-cluster communications are scheduled by a TDMA scheme, and a CDMA scheme is overlaid on the TDMA to handle interference-free inter-cluster communications. This means that our proposed method only aims at optimizing the channel assignment within the clusters, and does not consider the network clustering and inter-cluster communication issues. Indeed, like the previous works reported in the literature [22, 48, 49], our channel access scheduling method is proposed for a clustered ad-hoc network, where using a network clustering algorithm the wireless hosts are grouped into clusters, and an interference-free inter-cluster communication is guaranteed by a CDMA scheme.

In the proposed method, to avoid the inter-cluster interferences, each cluster is assigned a different code. Since the number of available codes is limited, each code can be assigned to one or more clusters except the neighboring and potential hidden clusters. On the other hand, each cluster-head is in charge of a collision-free slot assignment within the cluster. To do so, the cluster-head uses a TDMA scheme, and divides its dedicated code into time slots to form the channels. Indeed, in our proposed method, intra-cluster communications are organized by an adaptive TDMA scheme with a dynamic frame length, and CDMA scheme is overlaid on the TDMA to handle the inter-cluster communications.

Due to the node mobility and node failures, the topology of the wireless mobile ad hoc networks frequently changes. This causes the network information lose its validity very soon. In these networks, the cluster membership is highly dynamic and hard to predict since the hosts can move freely and randomly anywhere [50, 51]. Applying the basic TDMA scheme with a fixed frame length in such dynamic clusters significantly reduces the utilization of the channel which is a scarce resource in ad hoc networks. On the other hand, the input traffic characteristics of the mobile hosts are not necessarily the same and so each host requires a fraction of the TDMA frame (or a set of time slots) proportional to its traffic load. As a result, the channel throughput will considerably decrease, if the same potion of the bandwidth is assigned to all the users. Regarding the above mentioned problems with the basic TDMA, this paper proposes a dynamic frame length TDMA scheme in which the channel utilization and reusability dramatically increases when an

appropriate number of time slots is assigned to each host proportional to its traffic load.

In this method, after the network is clustered, each cluster-head is equipped with a (variable action-set) learning automaton. To form the action-set, each cluster-head assigns an action (of the action-set) to every one of its cluster members. That is, each cluster member is associated with an action, and so "cluster member" might be used instead of "action" or vice versa hereafter in this paper. Let $\underline{\alpha_i} = \{\alpha_i^j | \text{host } h_j \text{ is}$ a member of cluster-head $CH_i\}$ denotes the action-set of cluster-head CH_i, and let $\underline{p_i} = \{p_i^j | \forall \alpha_i^j \in \underline{\alpha_i}\}$ be the action probability vector defined over action set $\underline{\alpha_i}$. p_i^j denotes the choice probability of action α_i^j, and choosing action α_i^j means that host h_j (or cluster member CM_j) is permitted by cluster-head CH_i to access the channel. The proposed algorithm is an iterative algorithm which consists of a number of stages. Let k denotes the stage number. The action probability vector changes over time, and its initial value (where the stage number k is 0) is defined as

$$p_i^j(k) = \frac{1}{|\underline{\alpha_i}(k)|}; \quad \forall \alpha_i^j \in \underline{\alpha_i} \text{ and } k = 0 \tag{4}$$

where $|\underline{\alpha_i}(k)|$ denotes the cardinality of the action-set at stage k which is equal to the number of cluster members for cluster-head CH_i in this stage.

From (4), it can be concluded that all the cluster-members are initially chosen with the same probability. This means that the TDMA frame is initially subdivided into $|\underline{\alpha_i}(k)|$ portions of the equal size and each is assigned to a user. As a result, all the hosts (within a cluster) are initially assigned the same portion of the channel. As the proposed algorithm proceeds, this portion changes proportional to the traffic load for each member as follows: At each stage, the cluster-head randomly chooses one of its possible actions according to its action probability vector. The cluster-member corresponding to the selected action is permitted to transmit its packets for a time slot. Now, if the selected cluster member has a packet to transmit during the next time slot, the cluster-head increases the portion of TDMA frame assigned to this cluster member by rewarding the chosen action using (1) given in Sect. 2. Otherwise, this portion decreases by penalizing the selected action using (2) when the permitted cluster member has no packet to transmit. By this, the choice probability of a cluster member (for packet transmission) increases in future, if it has a packet to send, and decreases otherwise. Then, the cluster-head chooses one of its actions again according to the updated action probability vector and does the same operations as before. As the proposed algorithm progresses through successive iterations, the portion of the TDMA frame assigned to each cluster member converges to the proportion of time it has a packet to transmit. By this method, the channel utilization is maximized when the bandwidth portion taken by each cluster member is proportional to its need. In the beginning of each TDMA frame, an unused time slot is reserved for the new arrived hosts to transmit the control packets for requesting a slot assignment. Thus, no data packets are transmitted in this slot. Figure 2 shows the pseudo code of the proposed channel assignment procedure which is run at each cluster-head CH_i.

Procedure Channel Assignment (Cluster-head CH_i)
01:**Begin** procedure
02:   $k \leftarrow 0$
03:   CH_i forms its action-set $\underline{\alpha_i}(k)$
04:   CH_i computes the initial probability vector $\underline{p_i}(k)$
05:   **Loop** for ever
06:       CH_i selects one of its members at random
07:       CH_i assigns the channel to the selected member
08:       $k \leftarrow k + 1$
09:     **If** the selected member has a packet to transmit **then**
10:         CH_i updates $\underline{p_i}(k)$ by increasing the bandwidth portion of the selected member
11:       **Else**
12:         CH_i updates $\underline{p_i}(k)$ by decreasing the bandwidth portion of the selected member
13:     **End if**
14:   **End loop**
15:**End** procedure

Fig. 2 The pseudo code of the channel assignment procedure

When a host joins a cluster, it initially sends a *JREQ* (Join REQuest) message to the cluster-head. Cluster-head adds the sender's ID number to the cluster-member list as a newly joining member. It then updates its action-set and action probability vector as follows: Let us assume that cluster-head CH_i receives a *JREQ* message from host h_j at stage k, and let $\underline{\alpha_i}(k)$ denotes the action-set of cluster-head CH_i at stage k. Upon receiving a *JREQ* message, cluster-head CH_i adds a new action to the action-set $\underline{\alpha_i}(k)$ for the newly joining host (i.e., h_j) and defines its initial choice probability as

$$p_i^j(k) = \frac{1}{\left|\underline{\alpha_i}(k)\right| + 1} \tag{5}$$

From now on, the newly joining host is taken into account by the cluster-head for channel access scheduling. The choice probability of the other actions (i.e., for all $\alpha_i^r \in \underline{\alpha_i}(k)$) is updated as

$$p_i^r(k) = \frac{\left|\underline{\alpha_i}(k)\right|}{\left|\underline{\alpha_i}(k)\right| + 1} \cdot p_i^r(k); \quad \forall \alpha_i^r \in \underline{\alpha_i}(k) \text{ and } r \neq j \tag{6}$$

From (5) and (6), it can be seen that the probability of choosing the newly joining host h_j is subtracted from the choice probabilities of the other members proportional to their values.

When a host (or cluster member) decides to leave the cluster, it sends a *LREQ* (Leave REQuest) message including the sender's ID and receiver's ID (i.e., ID of the cluster-head) to its cluster-head. Upon receiving the *LREQ* message, cluster-head checks the received message to see if its ID matches to the receiver's ID. If so, the cluster-head performs as follows: Suppose that cluster member h_j sends a *LREQ* message to its cluster head (i.e., CH_i). Cluster-head CH_i first updates the choice probability of the other members (i.e., for all $\alpha_i^r \in \underline{\alpha_i}(k)$) as

$$p_i^r(k) = p_i^r(k) \cdot \frac{p_i^j(k)}{1 - p_i^j(k)} + p_i^r(k); \quad \forall \alpha_i^r \in \underline{\alpha_i}(k) \text{ and } r \neq j. \tag{7}$$

Then, cluster-head CH_i sets the choice probability of the leaving host h_j to zero. Finally, it removes the sender's ID from the cluster-member list, and prunes its own action-set by disabling the action associated with the leaving cluster member h_j forever as described in Subsection 2.1 on variable action-set learning automata. From (7), it is obvious that to update the probability vector of the new action-set, the probability of choosing the leaving host must be distributed among the other members proportional to their values. At each stage, the updating schemes given in (6) and (7) guarantee that the choice probability of the cluster-members sum up to one.

Although a *LREQ* message can be used to notify the cluster-head that a host is leaving the cluster, in our proposed method no explicit control message (e.g., *LREQ* message) is required to leave a cluster. This is because when a host leaves the cluster, the cluster-head receives no more packets from the leaving host and as given in Channel Assignment procedure after a short period of time the portion of the TDMA frame assigned to the leaving host approaches zero as the probability of choosing this host (to access the channel) converges to zero.

The main superiority of the proposed method over the others is that it does not need to reserve a large number of unused time slots for the new arrived hosts. In this method, each host is assigned a time slot as soon as it joins the cluster, and its bandwidth portion is adjusted proportional to its need. The assigned time slots are withdrawn and given to the other members, when a member leaves the cluster. Another advantage of the learning automata-based proposed method is the adaptation to the changing traffic conditions (environment). Therefore, unlike the other approaches reported in the literature [15, 16, 21, 22], we assume that the traffic load varies with time. Under such an assumption, the random environment, wherein the learning automata operate, is a non stationary environment in which the penalty probabilities are directly proportional to the traffic load and vary with time. When the traffic load of a given host changes, the bandwidth portion assigned to the host may not be appropriate anymore, and needs to be tuned again. In this case, the proposed method adjusts the penalty probability associated with this host proportional to its traffic load changes. New penalty probability leads the cluster-head to a new channel scheduling strategy by which a proper portion of the TDMA frame (bandwidth) is assigned to each cluster member. This results in adaptation to the network changing traffic conditions.

In learning automata theory, choosing the (proper) learning rate is the most challenging issue. In learning automata-based algorithms, the complexity of algorithm (e.g., computational and communicational complexity) is inversely proportional and the response error rate is directly proportional to the learning rate. In fact, the solution optimality increases as the learning rate decreases. This is because the learning automaton is capable of exploring almost all possible solutions and choosing the optimal one, if the learning rate is chosen small enough. On the other side, in ad hoc networks where the hosts suffer from the strict resource limitations, the communication and processing overheads should be kept as low as possible and so the learning rate cannot be selected very small. As a result, the learning rate must be selected so that a trade-off between the costs of algorithm and its optimality can be achieved. In this paper, we tried different learning rates and

found that a good trade-off can be made when the learning rate is set to 0.1. In the proposed channel assignment algorithm, the traffic load placed on each host (arrival rate) is assumed to be a random variable with unknown parameters (i.e., the network condition is non-stationary). In stationary conditions, where the network parameters (e.g., traffic load on the hosts) are assumed to be fixed, the convergence to the unique optimal solution (i.e., channel assignment strategy) is feasible. However in non-stationary conditions, where the optimal channel allocation strategy changes over time, the algorithm cannot converge to the optimal strategy of a snapshot of the network for ever, since it is not any longer valid after changing the network conditions. In this case, as the network conditions change the penalty probabilities also change and force the learning automaton to update the action probability vector toward the new optimum configuration. Under such circumstances, the learning automaton tries to find new optimal strategies upon changing the conditions. Therefore, to minimize the negative impacts of the variable traffic load on the channel utilization, the proposed algorithm attempts to find the channel allocation strategy under which the TDMA frame portion which is assigned to each host converges to the mean of distribution of its random load as the learning process proceeds.

4 Experimental Results

To investigate the performance of the proposed channel access scheduling scheme, we have conducted several simulation experiments in two sets. The first set of the simulation experiments compares the results of the proposed scheme with those of CS-DCA [21] (which is a channel segregation-based channel assignment method proposed by Akaiwa and Andoh) and Hybrid-DCA [22] (which is a dynamic channel assignment strategy proposed for clustered wireless ad-hoc networks by Wu). In this set of experiments, the proposed algorithm is compared with CS-DCA and Hybrid-DCA in terms of channel utilization, blocking rate, slot utilization and throughput. In the second set of the experimental simulations, the efficiency of the proposed algorithm is compared with DTSA [15] (which is a dynamic TDMA slot assignment protocol proposed by Kanzaki et al.) and DFLCA [16] (which is a dynamic frame length TDMA scheme based on expansion and recovery methods proposed by Wu) in terms of channel utilization, control overhead, and slot utilization. Since in CDMA/TDMA schemes the TDMA improvements have not received the attention they merit, the aim of the second set of the simulation experiments is to show the performance of the proposed channel assignment scheme in contrast with the pure TDMA schemes [15, 16] in which the slot assignment issues are only considered.

In our simulation scenarios, an ad-hoc network consisting of mobile hosts is modeled in which the hosts are randomly and uniformly distributed within a two-dimensional simulation area of size 1,000(m) × 1,000(m). Each mobile host moves according to a random waypoint mobility model with zero pause time and host speed (fixed at) 5(m/s). The number of hosts ranges from 10 to 100 with increment

step of 10. Each host is modeled as an infinite-buffer, store-and-forward queuing station. The radius of the transmission range of all hosts is set to be the same which is 250(m) throughout the simulation process. The channel bit rate is fixed at 2 Mbps, and the TDMA frame is subdivided into 64 transmission time slots of the equal length 4096 Byte. The first time slot of each TDMA frame is reserved for the control packets and the remaining slots are used to send data packets. Each simulation experiment is executed for 1,000 s. For each host, the arrival of the new connections is Poisson distributed with arrival rates (mean) $\lambda = 5, 10, 15,$ and 20 connections per minute. The duration of the connections is assumed to be exponentially distributed with mean 0.2. In each simulation experiment, an arrival rate λ is randomly drawn from set {5, 10, 15, 20} according to a uniform distribution and assigned to each host for generation of connection requests. That is, the traffic load placed on each host during a single simulation is a random variable having a Poisson distribution with a randomly selected parameter λ. The proposed method does not aim at optimizing the number of clusters or the number of codes assigned to the clusters. Therefore in simulation scenarios conducted for our proposed algorithm, the cluster formation and cluster maintenance mechanisms are supported by the clustering algorithm proposed in [52]. We also assume that a unique code is drawn from an infinite pool of free codes and assigned to each cluster. Our proposed algorithm is then implemented on such a network set up. The reward and penalty parameters of our algorithm are set to 0.1 and 0, respectively. The simulation results are averaged over 50 runs. In conducted experiments, the efficiency of the above mentioned channel assignment schemes is measured in terms of the following metrics of interest.

- *Channel utilization* This metric is defined as the ratio of the number of time slots assigned to the host to the total number of time slots in TDMA frame (or frame length). The channel utilization for the whole network is calculated as the average channel utilization of the hosts [53]. Our proposed channel assignment algorithm aims at assigning a portion of bandwidth to each host proportional to its need, and so it is expected to improve the channel utilization. This metric represents how much effective an algorithm assigns the channel.
- *Control overhead* This metric is defined as the average number of control packets related to the channel access scheduling process. In a CDMA/TDMA network, it is divided into three parts, namely clustering overhead, code assignment overhead, and slot assignment overhead. In this paper, since the proposed method only considers the slot assignment problem, we compute the control overhead regardless of the clustering and code assignment overhead. Such a control overhead is defined as the average number of (non-data) control packets related to slot assignment process generated per unit time. Due to the scarce bandwidth in ad-hoc networks, this metric must be reduced as much as possible.
- *Blocking rate* Blocking rate is defined as the ratio of the average number of the blocked connections due to the lack of free slots to the total number of connection requests.

- *Slot utilization* This metric is defined as the percentage of the used slots. Since in our proposed method the slots are assigned on demand and proportional to the traffic load, we expect that it increases the slot utilization rate.
- *Throughput* This metric is defined as the ratio of the average number of packets transmitted per slot to the total number of packets that can be transmitted per slot. The traffic load of the various hosts is different in realistic scenarios, so this metric can be optimized by our proposed algorithm in which the bandwidth portion (number of slots) assigned to each host is proportional to its traffic load.

Figure 7 shows the channel utilization of CS-DCA [21], Hybrid-DCA [22], and our proposed algorithm (LA-TDMA) as a function of the number of hosts. From the results shown in Fig. 3, it is clear that LA-TDMA significantly outperforms CS-DCA and Hybrid-DCA in terms of channel utilization. This is due to the fact that in LA-TDMA, the cluster-head reserves no free slots and assigns a time slot to each host as soon as it joins the cluster. Comparing the curves of CS-DCA and Hybrid-DCA depicted in Fig. 3, we observe that the rate of channel utilization in Hybrid-DCA is considerably higher than that of CS-DCA. This is because Hybrid-DCA has the global knowledge of the code assignment in the hidden cluster, and so provides a higher channel spatial reuse. From the results, it can be also seen that for all algorithms the channel utilization slightly decreases as the number of hosts increase.

The average number of time slots used (slot utilization) in CS-DCA [21], Hybrid-DCA [22], and LA-TDMA versus the network size is shown in Fig. 4. The obtained results show that Hybrid-DCA considerably outperforms CS-DCA in terms of slot utilization. This is because CS-DCA uses a larger set of codes to assign the clusters. This increases the number of unused time slots of each code in CS-DCA compared to Hybrid-DCA. For LA-TDMA, since the codes are chosen from a large set (an infinite pool) of codes, the same conclusion holds true. But on the other side in LA-TDMA due to an adaptive slot assignment method with a dynamic frame length, the number of unused slots will be minimized and so the rate of slot utilization in Hybrid-DCA is very close to (only slightly better than) LA-TDMA.
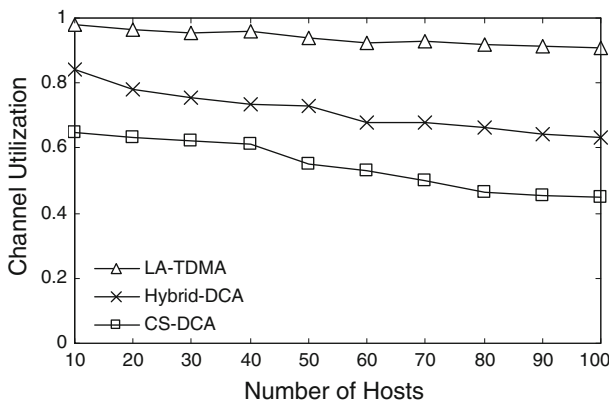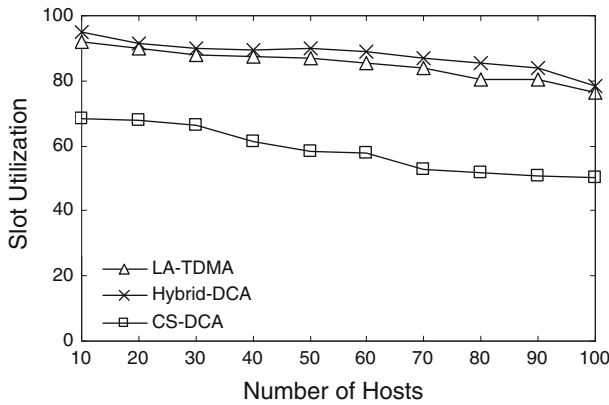

Fig. 3 Channel utilization versus the number of hosts

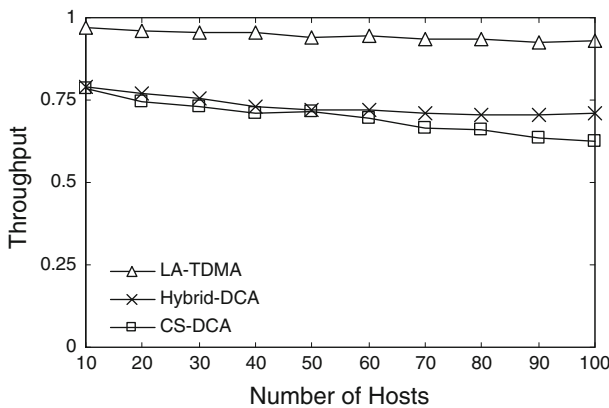**Fig. 4** Slot utilization versus the number of hosts

Fig. 5 Throughput versus the number of hosts

Figure 5 shows the average throughput of LA-TDMA, CS-DCA [21], and Hybrid-DCA [22] as a function of the number of hosts. As expected, it is observed from the results given in Fig. 5 that LA-TDMA has a much higher throughput as compared with CS-DCA and Hybrid-DCA. This is because LA-TDMA assigns a portion of TDMA frame to each host proportional to its need (traffic load). Furthermore, in LA-TDMA the slots assigned to the leaving hosts are immediately distributed among the remaining members. From the obtained results, it can be also seen that the throughput of Hybrid-DCA is very close to that of CS-DCA and only for the number of hosts larger than 60 Hybrid-DCA slightly outperforms CS-DCA.

Figure 6 shows the number of blocked connections due to the lack of free slots in the proposed method, CS-DCA [21], and Hybrid-DCA [22]. Comparing the curves of different schemes depicted in Fig. 6, we observe that LA-TDMA has a lower blocking rate in comparison with CS-DCA and Hybrid-DCA. In LA-TDMA, no slot is reserved for the new hosts, the probability of choosing the slots assigned to the
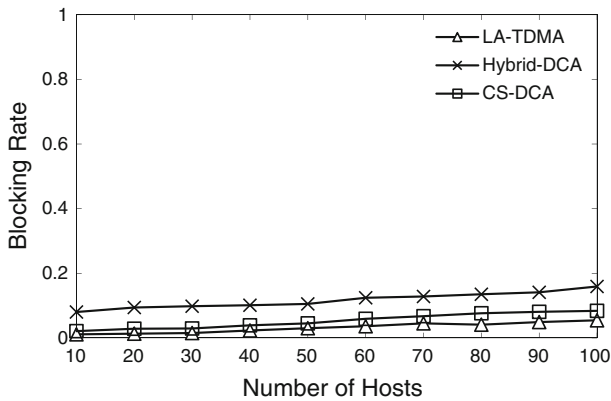
Fig. 6 Blocking rate versus the number of hosts

leaving hosts are immediately distributed over the other cluster-members, and the number of slots assigned to each host is proportional to its traffic load. Due to the above mentioned reasons, LA-TDMA shows better results as compared to CS-DCA and Hybrid-DCA. Comparing the results shown in Fig. 6, it is observed that CS-DCA outperforms Hybrid-DCA in terms of the blocking rate. As mentioned earlier, this is due to the fact that CS-DCA uses a larger number of codes than Hybrid-DCA.

The second group of our experiments investigates the performance of LA-TDMA in contrast with DTSA [15], and DFLCA [16] in terms of channel utilization, control overhead, and slot utilization. Figure 7 shows a channel utilization comparison between our proposed algorithm, DTSA, and DFLCA. From the results shown in this figure, we observe that the rate of channel utilization in DTSA is lower than that in DFLCA. This is because DTSA does not support any (frame length) recovery mechanism for reorganization of the unassigned time slots. This deficiency drastically reduces the percentage of the assigned time slots and so degrades the channel utilization. In LA-TDMA, each host is assigned a portion of the TDMA frame proportional to its need (traffic load), and unassigned time slots are immediately recovered and distributed among the other cluster members. So, LA-TDMA is expected to perform better as compared to DTSA and DFLCA. The results shown in Fig. 7 confirm the anticipation. These results also reveal that the channel utilization decreases as the number of host increases. Figure 8 shows the average number of time slots used in LA-TDMA, DTSA [15], and DFLCA [16] as a function of the network size. The results show that our proposed algorithm significantly outperforms the others, DFLCA lags far behind it, and DTSA has the worst results.

Figure 9 shows the total number of control messages (per second) generated by LA-TDMA, DTSA [15], and DFLCA [16] as a function of the number of hosts. From the results shown in this figure, it is clear that LA-TDMA is superior to DTSA and DFLCA in a control overhead point of view. This is due to the fact that LA-TDMA needs no explicit control message for adjusting the length of the TDMA frame (or updating the action probability vector). LA-TDMA exploits the rate of the
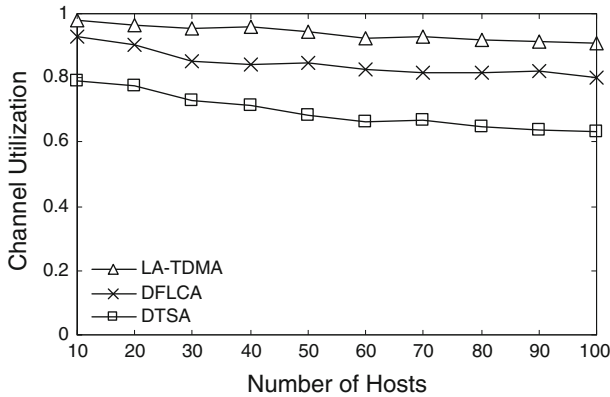
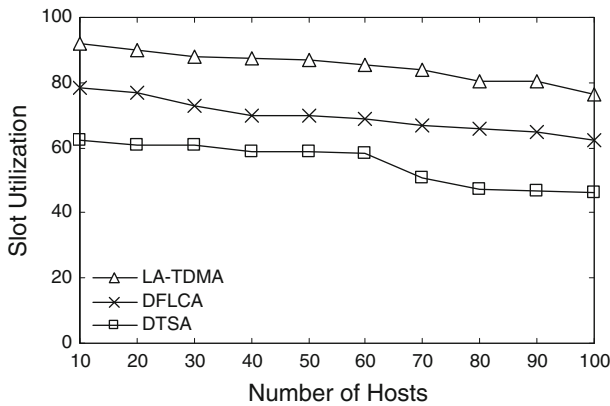**Fig. 7** Channel utilization versus the number of hosts

Fig. 8 Slot utilization versus the number of hosts

packets transmitted by each host to adjust the bandwidth portion needs to be assigned to the host.

As shown in this figure, the control message overhead of DFLCA is considerably higher than that of DTSA. This is because DFLCA uses a frame recovery mechanism in which a large number of request and confirmation packets should be sent. Furthermore in DFLCA, the increase in the channel spatial reuse is achieved by exchanging extra control information.

5 Conclusion

In this paper, we proposed an adaptive learning automata-based slot assignment algorithm for multi-hop clustered wireless ad-hoc networks when the input traffic parameters are unknown. This method is recommended for clustered CDMA/ TDMA networks in which the cluster-head is responsible for a collision-free slot
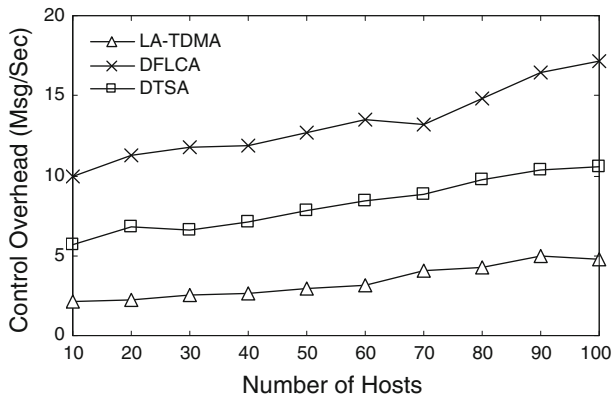
Fig. 9 Control overhead versus the number of hosts

assignment within the cluster. The aim of this paper is to show the capability of learning automaton as a random probabilistic learning technique to recognize the parameters of an unknown traffic distribution and to find the optimal slot allocation strategy. Based on this information, cluster-head assigns a portion of the TDMA frame to each member proportional to its traffic load. To show the superiority of the proposed channel assignment scheme over the existing methods, we conducted two sets of simulation experiments. In the first experiment set, we compared our proposed method with two well-known CDMA/TDMA schemes called CS-DCA and Hybrid-DCA, and in the latter set we compared it with two dynamic frame length slot assignment schemes called DTSA and DFLCA. The results showed that our proposed method outperforms the others in most cases.

References

1. Ghozzi, M., Dohler, M., Marx, F., Palicot, J.: Cognitive radio: Methods for the detection of free bands. Comptes. Rendus. Physique. **7**, 794–804 (2006)
2. Raychaudhuri, D., Jing, X., Seskar, I., Le, K., Evans, J.B.: Cognitive radio technology: from distributed spectrum coordination to adaptive network collaboration. Pervasive Mob. Comput. **4**, 278–302 (2008)
3. Akyildiz, I.F., Lee, W.-Y., Chowdhury, K.R.: CRAHNs: cognitive radio ad hoc networks. Ad Hoc. Netw. **7**, 810–836 (2009)
4. Saleh, A.A.M.: Inter-modulation analysis of FDMA satellite systems employing compensated and uncompensated TWTs. IEEE Trans. Commun. **30**(5), 1233–1242 (1982)
5. Lee, W.C.Y.: Overview of cellular CDMA. IEEE Trans. Veh. Technol. **40**(2), 291–302 (1991)
6. Sekimoto, T., Puente, J.G.: A satellite time-division multiple-access experiment. IEEE Trans. Commun. **16**(4), 581–588 (1968)
7. Capone, A., Gerla, M., Kapoor, R.: Efficient polling schemes for bluetooth picocells. In: Proceeding of the IEEE International Conference on Communications (ICC2001), vol. 7, pp. 1990–1994. Finland (2001)
8. Chrapkowski, A., Grube, G.: Mobile trunked radio system design and simulation. In: Proceedings of the IEEE Vehicular Technology Conference, pp. 245–250 (1991)

9. Goodman, D.J., Valenzuela, R.A., Gayliard, K.T., Ramamurthi, B.: Packet reservation multiple-access for local wireless communications. IEEE Trans. Commun. **37**, 885–890 (1989)
10. Abramson, N.: The ALOHA system-another alternative for computer communications. In: Proceedings of the AFIPS Fall Joint Computer Conference, vol. 37, pp. 281–285 (1970)
11. Kleinrock, L., Tobagi, F.A.: Packet switching in radio channels: part I-carrier sense multiple-access modes and their throughput-delay characteristics. IEEE Trans. Commun. **23**(12), 1400–1416 (1975)
12. Papadimitriou, G.I., Pomportsis, A.S.: Self-adaptive TDMA protocols: a learning-automata-based approach. In: Proceedings of International Conference on Networks, pp. 85–90 (1999)
13. Kim, S., Kim, J.I.: An adaptive time slot assignment algorithm for variable bandwidth switching systems. Comput. Oper. Res. **27**, 423–435 (2000)
14. Xin, L., Qionghia, D., Qiu-feng, W.: Time allocation scheme in IEEE 802.15.3 TDMA mechanism. J. Zhejiang Univ. Sci. A. **7**, 159–164 (2006)
15. Kanzaki, A., Uemukai, T., Hara, T., Nishio, S.: Dynamic TDMA slot assignment in ad-hoc networks. In: Proceedings of 17th IEEE International Conference on Advanced Information Networking and Applications (AINA'03), pp. 330–335 (2003)
16. Wu, C.M.: Dynamic frame length channel assignment in wireless multihop ad-hoc networks. J. Comput. Commun. **30**, 3832–3840 (2007)
17. Li, W., Wei, J.B., Wang, S.: An evolutionary-dynamic TDMA slot assignment protocol for ad-hoc networks. Wirel. Commun. Netw. Conf. 138–142 (2007)
18. Mo, R., Chew, Y.H.: System throughput analysis of rate adaptive TDMA system supporting two class services. Wirel. Netw. **11**, 687–695 (2005)
19. Furuya, Y., Akaiwa, Y.: Channel segregation, a distributed adaptive channel allocation scheme for mobile communication systems. IEICE Trans. **E74**, 1531–1537 (1991)
20. Gerla, M., Tsai, J.: Multicluster, mobile, multimedia radio network. ACM/Baltzer J. Wirel. Netw. **1**(3), 255–265 (1995)
21. Akaiwa, Y., Andoh, H.: Channel segregation—a self-organized dynamic channel allocation method: application to TDMA/FDMA microcellular system. IEEE J. Sel. Areas Commun. **11**(6), 949–954 (1993)
22. Wu, C.-M.: Hybrid dynamic channel assignment in clustered wireless multihop CDMA/TDMA ad-hoc networks. Wirel. Pers. Commun. **42**, 85–105 (2007)
23. Perez-Romero, J., Sallent, O., Agusti, R.: On the optimum traffic allocation in heterogeneous CDMA/TDMA networks. IEEE Trans. Wirel. Commun. **6**(9), 3170–3174 (2007)
24. Navaie, K., Yanikomeroglu, H.: Optimal downlink resource allocation for non-real time traffic in cellular CDMA/TDMA networks. IEEE Commun. Lett. **10**(4), 278–280 (2006)
25. Navaie, K., Yanikomeroglu, H.: Downlink joint base-station assignment and packet scheduling algorithm for cellular CDMA/TDMA networks. IEEE Int. Conf. Commun. 4339–4344 (2006)
26. Vannithamby, R., Sousa, E.S.: An optimum rate/power allocation scheme for downlink in hybrid CDMA/TDMA cellular system. In: 52nd IEEE Conference on Vehicular Technology, pp. 1734–1738 (2000)
27. Karp, R.M.: Reducibility among combinatorial problems, pp. 85–103. Complexity of Computer Computations, Plenum Press, USA (1972)
28. Akbari Torkestani, J., Meybodi, M.R.: Graph coloring problem based on learning automata. In: Proceedings of International Conference on Information Management and Engineering (ICIME 2009), pp. 718–772. Malaysia (2009)
29. Narendra, K.S., Thathachar, K.S.: Learning automata: an introduction. Printice-Hall, New York (1989)
30. Thathachar, M.A.L., Sastry, P.S.: A hierarchical system of learning automata that can learn the globally optimal path. Inf. Sci. **42**, 743–766 (1997)
31. Thathachar, M.A.L., Harita, B.R.: Learning automata with changing number of actions. IEEE Trans. Syst. Man Cybern. **SMG17**, 1095–1100 (1987)
32. Lakshmivarahan, S., Thathachar, M.A.L.: Bounds on the convergence probabilities of learning automata. IEEE Trans. Syst. Man Cybern. **SMC-6**, 756–763 (1995)
33. Narendra, K.S., Thathachar, M.A.L.: On the behavior of a learning automaton in a changing environment with application to telephone traffic routing. IEEE Trans. Syst. Man Cybern. **SMC-l0**(5), 262–269 (1980)
34. Akbari Torkestani, J., Meybodi, M.R.: A learning automata-based heuristic algorithm for solving the minimum spanning tree problem in stochastic graphs. J. Supercomput., in press (2010)

35. Akbari Torkestani, J., Meybodi, M.R.: Learning automata-based algorithms for finding minimum weakly connected dominating set in stochastic graphs. Int. J. Uncertain. Fuzziness Knowl. Based Syst., in press (2010)
36. Akbari Torkestani, J., Meybodi, M.R.: A new vertex coloring algorithm based on variable action-set learning automata. J. Comput. Inform. **29**(3), 1001–1020 (2010)
37. Akbari Torkestani, J., Meybodi, M.R.: Mobility-based multicast routing algorithm in wireless mobile ad hoc networks: a learning automata approach. J. Comput. Commun. **33**, 721–735 (2010)
38. Akbari Torkestani, J., Meybodi, M.R.: "Weighted Steiner Connected Dominating Set and its Application to Multicast Routing in Wireless MANETs," Wireless Personal Communications. Springer, Netherlands (2010). doi:10.1007/s11277-010-9936-4
39. Akbari Torkestani, J., Meybodi, M.R.: An efficient cluster-based CDMA/TDMA scheme for wireless mobile ad-hoc networks: a learning automata approach. J. Netw. Comput. Appl. **33**, 477–490 (2010)
40. Akbari Torkestani, J., Meybodi, M.R.: Clustering the wireless ad-hoc networks: a distributed learning automata approach. J. Parallel Distrib. Comput. **70**, 394–405 (2010)
41. Akbari Torkestani, J., Meybodi, M.R.: An intelligent backbone formation algorithm in wireless ad hoc networks based on distributed learning automata. J. Comput. Netw. **54**, 826–843 (2010)
42. 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 (1983)
43. Hashim, A.A., Amir, S., Mars, P.: Application of learning automata to data compression. In: Narendra, K.S. (ed.) Adaptive and Learning Systems, pp. 229–234. Plenum Press, New York (1986)
44. Oommen, B.J., Hansen, E.R.: List organizing strategies using stochastic move-to-front and stochastic move-to-rear operations. SIAM J. Comput. **16**, 705–716 (1987)
45. Unsal, C., Kachroo, P., Bay, J.S.: Multiple stochastic learning automata for vehicle path control in an automated highway system. IEEE Trans. Syst. Man Cybern. Part A **29**, 120–128 (1999)
46. Barto, A.G., Anandan, P.: Pattern-recognizing stochastic learning automata. IEEE Trans. Syst. Man Cybern. **SMC-15**, 360–375 (1985)
47. Akbari Torkestani, J., Meybodi, M.R.: Approximating the minimum connected dominating set in stochastic graphs based on learning automata. In: Proceedings of International Conference on Information Management and Engineering (ICIME 2009), pp. 672–676. Malaysia (2009)
48. Richard Lin, C.R., Gerla, M.: Adaptive clustering for mobile wireless networks. IEEE J. Sel. Areas Commun. **15**(7), 1265–1275 (1997)
49. Hod, T.C., Wu, C.M., Chan, M.C.: Dynamic channel assignment in clustered multihop CDMA/TDMA ad-hoc networks. In: Proceedings of 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 145–149 (2004)
50. Chen, Y.P., Listman, A.L.: Maintaining weakly-connected dominating sets for clustering ad-hoc networks. Ad-hoc Netw. **3**, 629–642 (2005)
51. Esnaashari, M., Meybodi, M.R.: A cellular learning automata based clustering algorithm for wireless sensor networks. Sens. Lett. **6**, 723–735 (2008)
52. Hou, T.C., Tsai, T.J.: On the cluster based dynamic channel assignment for multihop ad hoc networks. J. Commun. Netw. **4**(1), 40–47 (2002)
53. Yeo, J., Lee, H., Kim, S.: An efficient broadcast scheduling algorithm for TDMA ad-hoc networks. Comput. Oper. Res. **29**, 1793–1806 (2002)

Author Biographies

Javad Akbari Torkestani received the B.S. and M.S. degrees in Computer Engineering in Iran, in 2001 and 2004, respectively. He also received the Ph.D. degree in Computer Engineering from Science and Research University, Iran, in 2009. Currently, he is an assistant professor in Computer Engineering Department at Arak Azad University, Arak, Iran. Prior to the current position, he joined the faculty of the Computer Engineering Department at Arak Azad University as a lecturer. His research interests include wireless networks, mobile ad hoc networks, fault tolerant systems, learning systems, parallel algorithms, and soft computing.

Mohammad Reza Meybodi received the B.S. and M.S. degrees in Economics from Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M.S. and Ph.D. degree from

Oklahoma University, USA, in 1980 and 1983, respectively in Computer Science. Currently, he is a full professor in Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. Prior to current position, he worked from 1983 to 1985 as an assistant professor at Western Michigan University, and from 1985 to 1991 as an associate professor at Ohio University, USA. His research interests include wireless networks, fault tolerant systems, learning systems, parallel algorithms, soft computing and software development.