A New Local Rule for Convergence of ICLA to a Compatible Point

Hossein Morshedlou and Mohammad Reza Meybodi

***Abstract*—Many problems in the modern world have a decentralized and distributed nature. Irregular cellular learning automata (ICLA) is a powerful mathematical model for decentralized problems and applications. Convergence of ICLA to a compatible point is very important because this convergence can provide efficient solutions for the problems. The local rule of ICLA can play a key role in this convergence. A local rule that simply rewards or punishes learning automata just based on the response of environment and actions of neighbors does not guarantee convergence of ICLA to a compatible point. In this paper, we present a new local rule that guarantees convergence to a compatible point. Formal proofs for the convergence are provided and results of the conducted experiments support our theoretical findings.**

***Index Terms*—Adaptive systems, cellular automata, distributed computing, learning systems.**

I. Introduction

THERE are many problems that have a decentralized and distributed nature. Irregular cellular learning automata (ICLA) which is introduced formally in [12] is a powerful mathematical model for decentralized applications. ICLA is an extension of CLA [7] in which the restriction of regular structure is removed. An ICLA can be defined as a graph in which each vertex represents a cell and there are one or multiple learning automata in each cell [8]. Each edge in this graph represents neighborhood relation between two cells (two learning automata). There is a local rule in ICLA that determines the reinforcement signal to any particular learning automaton (LA). This rule uses response of environment and the actions selected by the neighboring LAs to generate the reinforcement signal. In an iterative process, each LA updates the state of its cell using the received reinforcement signal and ICLA evolves this way until the convergence or desired result is obtained. Concept of compatible point, introduced in [7], is an equilibrium point for CLA and this concept can be extended to ICLA as well. Each LA in ICLA learns to choose an action which maximizes the received reward (positive reinforcement) from the environment. Due to the dependency of the reward to the selected actions of the neighbors, choosing the same action by an LA in different rounds probably does not conduce to the same reward. In such conditions, each LA should reach equilibrium (i.e., an agreement on action selection) with its neighbors to increase its rewards. The equilibrium should be such that no LA in ICLA has any reason to change its selected action. In the other words, unilateral deviation from the equilibrium should not be profitable. The mentioned equilibrium point in ICLA is called compatible point or compatible configuration. Reaching the equilibrium is called convergence of ICLA to a compatible point. Convergence of ICLA to a compatible point is of great importance because it can conduce to efficient solutions for the problems and applications. Examples from such applications are graph coloring [3], clustering the wireless sensor networks [4] or channel assignment in cellular networks [6]. Because there is no any direct interaction between learning automata, each LA just perceives reinforcement signal or response of environment and observes the selected actions of its neighbors. Under these conditions, it is needed to find a way such that ICLA is able to converge to a compatible point. For this purpose, the local rule of ICLA can play a key role. The ordinary local rule which simply maps response (reward or penalty) of an environment to the perceived reinforcement signal by LA does not guarantee convergence of ICLA to a compatible point. In this paper, we aim to present a new local rule that guarantees this convergence. Formal proofs for the convergence are provided and results of the conducted experiments support our theoretical findings.

Manuscript received December 9, 2015; accepted April 30, 2016. This paper was recommended by Associate Editor L. Cao.

The authors are with the Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran 15914, Iran (e-mail: morshedlou@aut.ac.ir; mmeybodi@aut.ac.ir).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2016.2569464

II. Preliminary Concepts

In this section, LA and ICLA are introduced. The contents of these introductions are from [3] and [7]. After that, ICLA game is explained and concept of the compatible point will be presented formally.

A. Learning Automaton

An LA is represented by a triple $<\underline{\beta}, \underline{\alpha}, T>$, where $\underline{\beta}$ is the set of inputs, $\underline{\alpha}$ is the set of actions and T is learning algorithm. Actions of learning automata are inputs to environments. Let $\alpha_i(k) \in \underline{\alpha}$ and $\underline{p(k)}$, respectively, denote the selected action by LA and probability vector defined over the action set at round k. Let a and b indicate the reward and penalty parameters and determine the amount of increases and decreases of

the action probabilities. r is the number of actions that can be taken by an LA. At each round, 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 environment, and it is updated as given in (2), if the selected action is penalized

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

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

If $a = b$, the recurrence (1) and (2) are called linear reward-penalty (L_{R-P}) algorithm. If $a >> b$ the given equations are called linear reward-ε penalty ($L_{R-\varepsilon P}$), and finally if $b = 0$ they are called linear reward-inaction (L_{R-I}). When the learning algorithm is L_{R-I}, all unit probability vectors are absorbing states for LA because when LA enters an absorbing state, it cannot leave it again.

B. ICLA and ICLA Game

An ICLA is defined as an undirected graph in which, each vertex represents a cell which is equipped with one or multiple learning automata. Each edge in this graph represents neighborhood relation between two cells (two learning automata). There is a local rule in ICLA that uses the response of environment and the actions selected by the neighboring LAs to generate the reinforcement signal to any particular LA residing in a cell. The neighboring LAs of any particular LA constitute the local environment of that cell. ICLA game is an extended form of a stochastic game of learning automata [24]. Consider a game with N players. Every iteration the players select an action from their action sets. Then they receive stochastic payoffs from the environment. The payoff each player receives from environment depends on joint actions of all players. These payoffs can be dissimilar for different players. The stationary probability distributions which establish payoffs of joint actions are unknown for all players. Now assume a new game in which payoffs of player i and player j are independent of the selected actions of each others. Assume there exist a sequence of players $i, i_1, \ldots, i_m, j$ for $m \geq 1$ such that each two successive players in the sequence care about each others' strategy. Note that the payoffs of players i and j do not affect the selected action of each other instantly but in the long term, they would be influenced by each other's strategy indirectly and through other players who are in the sequence. Now if in former game and in the latter game we represent each player by an LA, then we have a game of learning automata in the former case and an ICLA game in the latter case. Actions of a player constitute the action set of LA. Hence at any given instant, the probability vector of an LA is equivalent to the mixed strategy of a player.

Definition 1: A configuration of ICLA at iteration t is denoted by $\underline{p}^t = (\underline{p}_1^t, \underline{p}_2^t, \ldots, \underline{p}_n^t)$ where $\underline{p}_i^t$ is the action probability vector of LA A_i.

Definition 2: The set of all probabilistic configurations K in ICLA are

$$K = \left\{ \underline{p} \,|\, \underline{p} = \left(\underline{p}_1, \underline{p}_2, \ldots, \underline{p}_n\right), \underline{p}_i = \left(p_{i1}, \ldots, p_{ir_i}\right), \right.$$
$$\left. \forall y, i : p_{iy} = 0 \text{ or } 1, \forall i : \sum_y p_{iy} = 1 \right\}. \tag{3}$$

Here r_i refers to the number of actions of LA_i.

Definition 3: The evolution of ICLA from a given initial configuration $\underline{p}^0$ is a sequence of configurations $\{\underline{p}^t\}_{t\geq 0}$ where $\underline{p}^{t+1} = G(\underline{p}^t)$. $G : K \to K$ is a mapping that describes global behavior or dynamics of ICLA.

Definition 4: The average reward for action r of LA A_i for configuration $\underline{p}$ is defined as

$$d_{ir}\left(\underline{p}\right) = \sum_{\alpha_2} \ldots \sum_{\alpha_{\tilde{m}}} F^i(r, \alpha_2, \ldots, \alpha_{\tilde{m}_i}) \prod_{\substack{l \in \tilde{N}(i) \\ l \neq i}} p_{l\alpha_l} \tag{4}$$

and the average reward is defined as

$$D_i\left(\underline{p}\right) = \sum_r d_{ir}\left(\underline{p}\right) \times p_{ir}. \tag{5}$$

The above definition implies that if the LA A_j is not a neighboring LA for A_i, then $d_{ir}(\underline{p})$ does not depend on $\underline{p}_j$.

Definition 5: In ICLA game, a configuration $\underline{p}$ is compatible if

$$\sum_r d_{ir}\left(\underline{p}\right) \times p_{ir} \geq \sum_r d_{ir}\left(\underline{p}\right) \times q_{ir} \tag{6}$$

for all configurations $\underline{q} \in K$ and all LA A_i. In other words, a compatible configuration in an ICLA game is equivalent to the Nash equilibrium of a game of learning automata. Therefore, in a compatible configuration, unilateral deviation of an LA is not profitable. In rest of this paper, we use compatible point instead of compatible configuration.

III. Related Work

LA has performed very well in different areas such as pattern recognition [5], certificate verification [14], stochastic graphs [21], Petri nets [28], and process mining [31] to name just a few. Potentials of a single LA to operate in a stochastic environment, made researchers to extend its idea to a team of LAs as a group operating in a game setting. Common payoff [26] and zero-sum [15], [26] games are studied in the literature. An important result is that learning automata reach a game-theoretic equilibrium point as the penalty parameter goes to zero [15], [22]. In [10], a cooperative game-theoretic approach to model a multilevel decision-making process is presented. Players of this game are learning automata which decide about the profitability of group constitution. The reported results illustrate that delayed information, both types of penalties (asymmetric and symmetric) lead to chaos but with different Lyapunov exponents. These results are confirmed by [9] as well. The games of LAs have also been used in many applications such

as multiple access channel selection [31], congestion avoidance in wireless networks [19], channel selection in radio networks [27], clustering in wireless ad hoc networks [2], spectrum allocation in cognitive networks [16], and solving various NP-complete problems [25]. These applications show capabilities of LA when multiple learning automata interact with each other. Another class of games, which uses capabilities of learning automata, is Markov games category. The idea of using LAs in Markov games originates from the ability of LAs to manage and control a Markov chain [29] without knowledge about transition probabilities in an independent and noncentralized way. In [30], it is illustrated that a group of independent LAs is able to converge to the equilibrium of a limiting game, even with limited observations. Furthermore in [17], some learning automata-based methods for finding optimal policies in Markov games are suggested. In all the mentioned works, all LAs affect each other directly but this may not be true for some applications. There are many applications that players do not influence directly each other. The mathematical framework presented in [7] is appropriate for modeling such situations. Also, the introduced concept of a compatible point in [7] can be used as an equivalent equilibrium point to Nash. However, capabilities and applications of CLA are not restricted to stochastic games with transitive influences. CLA has been found to perform well in many application areas such as channel assignment in cellular networks [6], wireless sensor networks [4], mobile networks [13], image processing [18], and solving NP-hard problems [11], to name just a few.

IV. Convergence of ICLA to Compatible Point

In this section, we propose a new local rule but first we show two preliminary lemmas which are needed to prove convergence of ICLA to a compatible point.

A. Preliminary Lemmas

In this section, we prove existence of a compatible point in ICLA and then preliminary lemmas will be presented.

Lemma 1: In ICLA game, there is at least one compatible point.

Proof: For the proof first we define a particular mapping T over K ($T : K \rightarrow K$) and we show that T has at least one fixed point. Then we prove that a point is a fixed point if and only if it is a compatible point. For complete proof see the supplementary materials of this paper. ■

Fig. 1 shows an algorithm for estimation of neighbor's strategy. An epoch t with length L is a time interval in which learning automata choose their actions L times. For example, if in each iteration learning automata choose actions then L iterations constitute an epoch with length L.

Lemma 2: In Algorithm 1 (see Fig. 1), if t approaches infinity then $\hat{\underline{p}}_i^t$ converges to $\underline{p}_i^t$.

Proof: To prove $\hat{\underline{p}}_i^t$ is an accurate estimation of $\underline{p}_i^t$ in infinity, we show when ICLA is converged to a point such as $\underline{p}^*$ [when probability vector of LA_i ($\underline{p}_i^t$) is evolved to $\underline{p}_i^*$], we have $\lim_{t\to\infty}(\hat{\underline{p}}_i^t - \underline{p}_i^*) = 0$. This means $\hat{\underline{p}}_i^t$ is an accurate estimation of $\underline{p}_i^t$ at infinity. For complete proof see the supplementary materials of this paper. ■

$n_i^t(a_j)$ shows how many times, LA_i selected action a_j during epoch t.
L = length of epoch (number of times LA_i chooses an action)
$t = 1$;
$\hat{\underline{p}}_{ij}^1 = \dfrac{n_i^1(a_j)}{L}$;
while(*True*) {
$\hat{\underline{p}}_{ij}^{t+1} = \dfrac{\hat{\underline{p}}_{ij}^t + \dfrac{n_i^t(a_j)}{L}}{2}$;
$t = t + 1$; }

Fig. 1. Pseudocode of Algorithm 1 for estimation of neighbor's strategy.

B. Proposed Local Rule

Let $\underline{p}_i^c$ denotes the current probability vector of LA_i and the best response probability vector of LA_i (with respect to the probability vectors of its neighbors) to be indicated by $\underline{p}_i^{\text{br}}$. Algorithm 2 (see Fig. 2) illustrates our proposed local rule to generate reinforcement signal (β). According to Definition 5 in a compatible point, for every LA unilateral deviation is not profitable. As a result, when ICLA is converged to a compatible point the probability vector of every LA is the best response to the probability vectors of its neighbors. Considering this fact, the concept behind the proposed local rule is as below.

In an iterative process, if each LA selects its action according to the best response probability vector (respect to the observed and estimated probability vectors of its neighbors) then with accurate estimations, a noncompatible point cannot be a stable point. In other words, in a noncompatible point, there is at least one LA that unilateral deviation will be profitable for it. To estimate the best response probability vector for LA i we use the expression in line 5 in the proposed local rule. The proposed local rule tries to reduce difference ($\Delta\underline{R}_i^{\text{RS}}$) between the current probability vector ($\underline{p}_i^c$) and the best response probability vector ($\underline{p}_i^{\text{br}}$). The average reward for LA i (d_i) is determined such that action j to be rewarded proportional to $\Delta\underline{R}_i^{\text{RS}}(j)$ to reduce the difference between $\underline{p}_i^{\text{br}}$ and $\underline{p}_i^c$. For computation of $\underline{p}_i^{\text{br}}$ we need $\hat{\underline{p}}^t$ and Q_t^i. $\hat{\underline{p}}^t$ can be estimated using Algorithm 1, but for Q_t^i, we should maintain a table of Q-values $Q^i(a^1, \ldots, a^{\bar{m}_i})$ for each joint action of $(a^1, \ldots, a^{\bar{m}_i})$. We define the optimal Q-values for LA_i with respect to its neighbors as

$$Q_*^i\left(a^1, \ldots, a^{\bar{m}_i}\right) = \text{Er}_i\left(a^1, \ldots, a^{\bar{m}_i}\right) + \beta.v^i\left(\underline{p}_1^*, \underline{p}_2^*, \ldots, \underline{p}_i^{\text{br}}\left(\underline{p}^*, Q_*^i\right), \ldots, \underline{p}_{\bar{m}_i}^*\right) \quad (7)$$

1 $\underline{p}_i^c$ = current probability vector of LA_i
2 $\underline{p}_i^{br}$ = best response probability vector accuired from (8)
3 d_i = vector of reward probabilities
4 *Begin*
5 $\underline{p}_i^{br}(\underline{\hat{p}}^t, Q_t^i) = \arg\max_{\underline{p}_i} \sum_{a^1} \sum_{a^2} \cdots \sum_{a^{\bar{m}_i}} \underline{p}_i(\underline{\hat{p}}^t, Q_t^i)(a^i) \prod_{\substack{j=1 \\ j\neq i}}^{\bar{m}_i} \underline{\hat{p}}_j^t(a^j).Q_t^i(a^1,...,a^{\bar{m}_i});$
6 $\Delta\underline{p}_i^{RS} = \underline{p}_i^{br} - \underline{p}_i^c;$
7 $j_{\max} = \arg\max_j \Delta\underline{p}_i^{RS}(j);$
8 $j_{\min} = \arg\min_j \Delta\underline{p}_i^{RS}(j);$
9 $d_i = \left(\frac{\Delta\underline{p}_i^{RS}(1) - \Delta\underline{p}_i^{RS}(j_{\min})}{\Delta\underline{p}_i^{RS}(j_{\max}) - \Delta\underline{p}_i^{RS}(j_{\min})}, ..., \frac{\Delta\underline{p}_i^{RS}(\bar{m}_i) - \Delta\underline{p}_i^{RS}(j_{\min})}{\Delta\underline{p}_i^{RS}(j_{\max}) - \Delta\underline{p}_i^{RS}(j_{\min})} \right)$
10 if (selected action == $a^k \in \{a^1,...,a^{\bar{m}_i}\}$) then
11 if ($\Delta\underline{p}_i^{RS}(k) \geq 0$) then
12 set $\beta = 1$ with probability $d_i(k)$ and
set $\beta = 0$ with probability $(1 - d_i(k))$;
13 else if ($\Delta\underline{p}_i^{RS}(k) < 0$) then
14 set $\beta = 0$ with probability $d_i(k)$ and
set $\beta = 1$ with probability $(1 - d_i(k))$;
15 update Q - values using (8);
16 return β;

Fig. 2. Pseudocode of the proposed local rule (Algorithm 2).

where $v^i(\underline{p}_1^*, \underline{p}_2^*, \ldots, \underline{p}_i^{\text{br}}(\underline{p}^*, Q_*^i), \ldots, \underline{p}_{\bar{m}_i}^*)$ is total discounted reward of LA_i over infinite iterations when probability vectors of the neighbors are $(\underline{p}_1^*, \underline{p}_2^*, \ldots, \underline{p}_{i-1}^*, \underline{p}_{i+1}^*, \ldots, \underline{p}_{\bar{m}_i}^*)$ and $\underline{p}_i^{\text{br}}(\underline{p}^*, Q_*^i)$ is the best response probability vector with respect to the probability vectors of the neighbors. Er_i is the expected value of the environment responses that LA i has received until iteration t. The ordinary local rule uses these responses to simply reward or penalize the selected action of LA. Since the environment is stochastic and information-incomplete, we have no idea about Q-values and we should find a way to estimate them. Initially, we propose to assign a fixed value (e.g., zero) to Q-values and then update them using

$$\begin{aligned} Q_{t+1}^i\left(a^1, \ldots, a^{\bar{m}_i}\right) &= (1-\alpha) \times Q_t^i\left(a^1, \ldots, a^{\bar{m}_i}\right) \\ &+ \alpha \times \left[\text{Er}_i^t\left(a^1, \ldots, a^{\bar{m}_i}\right) + Q_t^i\left(a^1, \ldots, a^{\bar{m}_i}\right) \right. \\ &\left. \times \beta \cdot \sum_{a^1} \sum_{a^2} \cdots \sum_{a^{\bar{m}_i}} \underline{p}_i^{\text{br}}\left(\underline{\hat{p}}^t, Q_t^i\right)(a^i) \prod_{\substack{j=1 \\ j\neq i}}^{\bar{m}_i} \underline{\hat{p}}_j^t(a^j) \right] \end{aligned} \quad (8)$$

where $\underline{p}_i^{\text{br}}(\underline{\hat{p}}^t, Q_t^i)$ is the best response probability vector against $(\underline{\hat{p}}_1^t, \ldots, \underline{\hat{p}}_{i-1}^t, \underline{\hat{p}}_{i+1}^t, \ldots, \underline{\hat{p}}_{\bar{m}_i}^t)$. Now we point out a corollary from [23] that is the basis of our proof and then we show that using (8) Q_t^i will converge to Q_*^i when $t \to \infty$.

Corollary 1 [23]: Consider the process generated by iteration of (9), where $0 \leq f_t(x) \leq 1$

$$V_{t+1}(x) = (1 - f_t(x)) \times V_t(x) + f_t(x) \times [P_t V_t](x). \quad (9)$$

If the process defined by (10) converges to v^* with probability one

$$U_{t+1}(x) = (1 - f_t(x)) \times U_t(x) + f_t(x) \times \left[P_t v^*\right](x) \quad (10)$$

and the following conditions hold.

1) There exist number $0 < \gamma < 1$ and a sequence $\lambda_t \geq 0$ converging to zero with probability one such that $||P_t V - P_t v^*|| \leq \gamma ||V - v^*|| + \lambda_t$ holds for all $V \in \{V\}$.
2) $0 \leq f_t(x) \leq 1$, $t \geq 0$, and $\sum_{t=1}^n f_t(x)$ converges to infinity uniformly in x as $n \to \infty$.

Then, the iteration defined by (9) will converges to v^* with probability one.

Before presenting the convergence proof for the proposed local rule, we first explain some notations. We put $f_t(x) = \alpha$ and use Q instead of V. Thus, $\{V\}$ is replaced by $\{Q\}$, which is a set of all Q-functions ($Q : A_1 \times \cdots \times A_{\bar{m}} \to R$) and P_t is a mapping from $\{Q\}$ to $\{Q\}$. Also $||Q||$ is equal to $\max_{(a^1,\ldots,a^{\bar{m}})} |Q(a^1, \ldots, a^{\bar{m}})|$ which is a finite value.

Lemma 3: Using (8), the updated Q-values converge to optimal Q-values of (7) with probability one.

Proof: Corollary 1 is the base of this proof. Let Q_*^i to be the optimal Q-values of (7). We define a mapping $P_t : \{Q\} \to \{Q\}$ such that (8) can be rewritten in form of (9). Then we show that using P_t we can write

$$\begin{aligned} &\left\| P_t^i Q_t^i\left(a^1, \ldots, a^{\bar{m}_i}\right) - P_t^i Q_*^i\left(a^1, \ldots, a^{\bar{m}_i}\right) \right\| \\ &\quad \leq \beta \cdot \left\| Q_t^i\left(a^1, \ldots, a^{\bar{m}_i}\right) - Q_*^i\left(a^1, \ldots, a^{\bar{m}_i}\right) \right\| + \lambda_t. \end{aligned} \quad (11)$$

Then using (11) and (12), which are results of Lemma 2 and Corollary 1, we show that all the requirements of Corollary 1 are satisfied. Therefore, updating Q-values using (8) will cause convergence to optimal Q-values of (7) with probability one and proof is completed. For detailed proof see the supplementary materials of this paper

$$\lim_{t\to\infty} P\left(\left| \underline{\hat{p}}_i^t - \underline{p}_i^* \right| > \varepsilon \right) = 0. \quad (12)$$

■

Theorem 1: Using the local rule of Algorithm 2, ICLA converges to a compatible point.

Proof: In a compatible point, every LA has the best response configuration (probability vector) to configurations of the other learning automata. We use Lemmas 2 and 3 to show that $\underline{p}_i^{\text{br}}$ is that best response configuration in infinity [provided that $\underline{\hat{p}}_i^t$ is an accurate estimation of $\underline{p}_i^t$ (Lemma 2) and the Q-values converge to optimal Q-values (Lemma 3)]. We also show Algorithm 2 diminishes the difference between the current probability vector of LA ($\underline{p}_i^c$) and $\underline{p}_i^{\text{br}}$ in each iteration ($\Delta\underline{p}_i^{\text{RS}} = \underline{p}_i^{\text{br}} - \underline{p}_i^c$) and as a consequence ICLA converges to a compatible point. For detailed proof see the supplementary materials of this paper. ■

C. Complexity Analysis of the Proposed Local Rule

As described before, each LA in ICLA needs to maintain Q-values for each combination of joint actions of itself and its neighbors. These Q-values are maintained internally by the LA. The LA i updates $Q^i(a^1, \ldots, a^{\bar{m}_i})$, while $a^j (j = 1, \ldots, \bar{m}_i)$ is the selected action of the jth neighbor (assume the neighbors are labeled by index j). $\bar{m}_i$ is the number of neighbors of LA i in ICLA. Let $|A^i|$ be the size of action space of LA i. If $\bar{m} = \max_i \bar{m}_i$ and $|A| = \max_i |A^i|$ then $n \times |A|^{\bar{m}}$ is an upper bound of total space requirement for Q-values. Here n is the number of learning automata in ICLA. In addition, there is a similar space requirement to maintain $\mathrm{Er}_i(a^1, \ldots, a^{\bar{m}_i})$ values. Therefore, the proposed local rule in terms of space complexity is linear in the number of learning automata (size of ICLA), polynomial in the number of actions of learning automata, but exponential in the number of neighbors of learning automata.

Running time of Algorithm 2 is dominated by the computation of $\underline{p}_i^{\mathrm{br}}(\hat{p}^t, Q_t^i)$. Computational complexity of this computation is $|A|^{\bar{m}}$. It is polynomial in the number of actions of learning automata but exponential in the number of neighbors of learning automata.

Note: If we have multiple learning automata in each cell of ICLA (see [8]) then each LA in a cell will be a neighbor for every LA which is located in that cell or its neighboring cells.

V. Experiments and Results

This section contains results of conducted experiments. First, we present two numerical experiments and then an example for application of ICLA in channel assignment problem is given.

A. Numerical Experiments

The numerical experiments are given to illustrate the superiority and effectiveness of the proposed local rule. The proposed local rule is compared with the ordinary local rule. In the experiments $\mathrm{ICLA}_{m,n}$ refers to an ICLA with m cells and n actions for each LA located in a cell. Because the notation is destitute of neighboring details so a neighboring matrix is also needed to specify a unique ICLA. Learning algorithms of all learning automata are L_{R-I}.

1) Numerical Experiment 1: The aim of this experiment is to compare convergence rate of ICLA using the proposed local rule versus the ordinary local rule. For this purpose, we investigate convergence rate in $\mathrm{ICLA}_{3,2}$, $\mathrm{ICLA}_{4,2}$, and $\mathrm{ICLA}_{5,2}$. The structure of these ICLAs are illustrated in Fig. 3. The edges in the graphs show neighborhood relation. Here response of environment is generated using a response matrix. This matrix contains probability of rewarding the selected actions of learning automata in ICLA. For example, when the selected joint actions by LAs in $\mathrm{ICLA}_{3,2}$ is (low, high, and high) and the corresponding entry in the response matrix is (0.91, 0.88, and 0.91) (see Table I) then the reinforcement signals to LA1, LA2, and LA3 is reward with probability 0.91, 0.88, and 0.91, respectively. Tables I–III show the response matrices for $\mathrm{ICLA}_{3,2}$, $\mathrm{ICLA}_{4,2}$, and $\mathrm{ICLA}_{5,2}$, respectively.

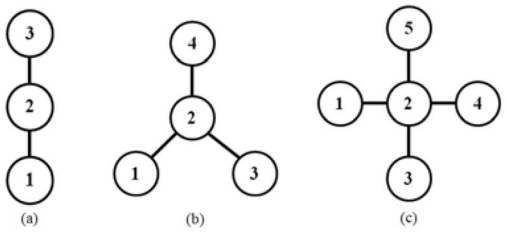

Fig. 3. ICLA structure for (a) $\mathrm{ICLA}_{3,2}$, (b) $\mathrm{ICLA}_{4,2}$, and (c) $\mathrm{ICLA}_{5,2}$.

TABLE I
Response Matrix for $\mathrm{ICLA}_{3,2}$

a_1	a_3	a_2 = ***Low***
Low	***Low***	(0.54, 0.59, 0.72)
	High	(0.54, 0.82, 0.66)
High	***Low***	(0.48, 0.75 , 0.72)
	High	(0.48, 0.59, 0.66)
		a_2 = ***High***
Low	*Low*	(0.91, 0.68, 0.81)
	High	**(0.91,0.88, 0.91)**
High	*Low*	(0.84, 0.70, 0.81)
	High	(0.84, 0.77, 0.91)

TABLE II
Response Matrix for $\mathrm{ICLA}_{4,2}$

		a_2 = ***Low***	
a_1	a_3	a_4 = ***Low***	a_4 = ***High***
Low	***Low***	(0.54, 0.59, 0.72, 0.62)	(0.54, 0.71, 0.72, 0.48)
	High	(0.54, 0.82, 0.66, 0.62)	(0.54, 0.69, 0.66, 0.48)
High	***Low***	(0.48, 0.75, 0.72, 0.62)	(0.48, 0.61 , 0.72, 0.48)
	High	(0.48, 0.59, 0.66, 0.62)	(0.48, 0.79, 0.66, 0.48)
		a_2 = ***High***	
a_1	a_3	a_4 = ***Low***	a_4 = ***High***
Low	***Low***	(0.91, 0.68, 0.81, 0.95)	(0.91, 0.68, 0.81, 0.78)
	High	**(0.91,0.88, 0.91, 0.95)**	(0.91,0.76, 0.91, 0.78)
High	***Low***	(0.84, 0.70, 0.81, 0.95)	(0.84, 0.70, 0.81, 0.78)
	High	(0.84, 0.77, 0.91, 0.95)	(0.84, 0.77, 0.91, 0.78)

Table IV shows convergence rates with different amounts of reward parameter a [see (1)] using the proposed local rule and the ordinary local rule. These results are obtained from 1000 times running of the learning process by starting from the initial configuration of (0.5, 0.5) for each LA in the ICLAs.

As illustrated in Table IV, using the bigger reward parameters we have lower convergence rate by the both local rules. Problem of catching in absorbing states (see Section II-A) is a major reason of lower convergence rate here. Except for the big reward parameter sizes in $\mathrm{ICLA}_{3,2}$, the proposed local rule always reaches better convergence rate than the ordinary local rule. The difference in convergence rate is due to the local rules. The ordinary local rule just uses the response of environment in current iteration to generate reinforcement signal, while the proposed local rule uses a history of the responses (using Q-values). When size of the reward parameter increases, the convergence rate drops. Despite the better convergence rate using the proposed local rule, but the drop of convergence rate

TABLE III
RESPONSE MATRIX FOR $ICLA_{5,2}$

			a_2 = *Low*	
a_1	a_3	a_5	a_4 = *Low*	a_4 = *High*
Low	*Low*	*Low*	(0.54, 0.59, 0.72, 0.62, 0.57)	(0.54, 0.71, 0.72, 0.48, 0.57)
		High	(0.54, 0.67, 0.72, 0.62, 0.68)	(0.54, 0.81, 0.72, 0.48, 0.68)
	High	*Low*	(0.54, 0.82, 0.66, 0.62, 0.57)	(0.54, 0.69, 0.66, 0.48, 0.57)
		High	(0.54, 0.65, 0.66, 0.62, 0.68)	(0.54, 0.77, 0.66, 0.48, 0.68)
High	*Low*	*Low*	(0.48, 0.75, 0.72, 0.62, 0.57)	(0.48, 0.61 , 0.72, 0.48, 0.57)
		High	(0.48, 0.83, 0.72, 0.62, 0.68)	(0.48, 0.70 , 0.72, 0.48, 0.68)
	High	*Low*	(0.48, 0.59, 0.66, 0.62, 0.57)	(0.48, 0.79, 0.66, 0.48, 0.57)
		High	(0.48, 0.68, 0.66, 0.62, 0.68)	(0.48, 0.81, 0.66, 0.48, 0.68)
			a_2 = *High*	
a_1	a_3	a_5	a_4 = *Low*	a_4 = *High*
Low	*Low*	*Low*	(0.91, 0.68, 0.81, 0.95, 0.88)	(0.91, 0.68, 0.81, 0.78, 0.88)
		High	(0.91, 0.77, 0.81, 0.95, 0.75)	(0.91, 0.78, 0.81, 0.78, 0.75)
	High	*Low*	**(0.91,0.88, 0.91, 0.95, 0.88)**	(0.91,0.76, 0.91, 0.78, 0.88)
		High	(0.91,0.72, 0.91, 0.95, 0.75)	(0.91,0.69, 0.91, 0.78, 0.75)
High	*Low*	*Low*	(0.84, 0.70, 0.81, 0.95, 0.88)	(0.84, 0.70, 0.81, 0.78, 0.88)
		High	(0.84, 0.69, 0.81, 0.95, 0.75)	(0.84, 0.80, 0.81, 0.78, 0.75)
	High	*Low*	(0.84, 0.77, 0.91, 0.95, 0.88)	(0.84, 0.77, 0.91, 0.78, 0.88)
		High	(0.84, 0.35, 0.91, 0.95, 0.75)	(0.84, 0.75, 0.91, 0.78, 0.75)

TABLE IV
CONVERGENCE RATE USING DIFFERENT REWARD PARAMETERS

Reward Parameter		E-05	E-04	E-03	E-02	E-01	2.E-1
$ICLA_{3,2}$	**Proposed**	%100	%99	%92	%81	%54	%32
	Ordinary	%99	%97	%88	%76	%64	%42
$ICLA_{4,2}$	**Proposed**	%100	%98	%88	%71	%49	%24
	Ordinary	%81	%73	%68	%53	%39	%26
$ICLA_{5,2}$	**Proposed**	%100	%95	%85	%70	%44	%16
	Ordinary	%66	%64	%56	%44	%28	%10

using the proposed local rule is more sensible than the drop using the ordinary local rule. It is due to significant changes of ICLA configuration in two successive iterations using big reward parameters. This causes a drop in convergence rate because Algorithm 2 assumes the best response in iteration t (see $\underline{p}_i^{\text{br}}$ in Algorithm 2) is too close to the best response in iteration $t+1$.

Using Table I as a response matrix, $ICLA_{3,2}$ converges to (low, high, and high) which is a compatible point. Fig. 4 shows the required iterations for convergence of ICLA to the compatible point for $ICLA_{3,2}$, $ICLA_{4,2}$, and $ICLA_{5,2}$ using the proposed and ordinary local rules. As illustrated in this figure, using the proposed local rule ICLA need further iterations to converge. Now let $L_{nm} = (I_{nm}(\text{Proposed})/I_{nm}(\text{Ordinary}))$ where $I_{nm}(R)$ denotes the number of iterations $ICLA_{n,m}$ needs for convergence using the local rule R. According to the results illustrated in Fig. 4, we have $L_{32} = 3.6$, $L_{42} = 2.4$, and $L_{52} = 1.7$. For $n \geq 9$, L_{n2} is less than 1. So by increasing the number of cells in $ICLA_{n,2}$, it is expected that fewer iterations to be needed for convergence by the proposed local rule in comparison to the ordinary local rule. Fig. 4 also demonstrates that using the ordinary local rule the necessary iterations for convergence of $ICLA_{3,2}$, $ICLA_{4,2}$, and $ICLA_{5,2}$ are 23%, 29%,

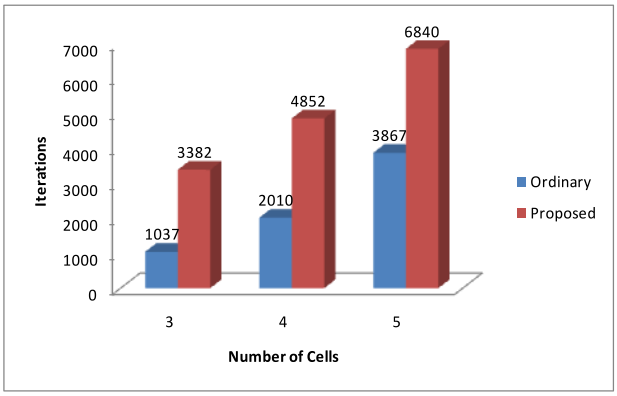

Fig. 4. Comparing the number of necessary iterations for convergence of ICLA using the proposed and ordinary local rules.

and 36% of $(I_{n2}(\text{Proposed}) + I_{n2}(\text{Ordinary}))$ for n = 3, 4, and 5, respectively.

2) Numerical Experiment 2: The aim of this numerical experiment is to study the convergence of ICLA when it starts learning from different initial configurations. For this experiment we use an $ICLA_{3,3}$ with L_{R-I} learning algorithm. Structure of $ICLA_{3,3}$ is similar to $ICLA_{3,2}$ in Fig. 3(a). Responses of the environment are generated using the response matrix of Table V. Table VI contains the initial configurations for this ICLA. For example, configuration 3 shows that the initial probability vector of the LA in cell 2 is [0.2, 0.7, 0.1]. Employing Table V to generate environment responses, [(0, 1, 0), (0, 0, 1), (0, 0, 1)] is a compatible configuration for $ICLA_{3,3}$. Table VII shows the convergence rate of ICLA using different amounts of reward parameter a. As illustrated in Table VII, the proposed local rule still has a better convergence rate. Moreover, by starting from different initial configurations, using the proposed local rule differences between convergence rates are smaller. In other words, by the proposed local rule, the convergence rate is less dependent on initial configuration. Fig. 5 compares this dependency. The best and worst convergence rates are obtained by starting from configurations 2 and 3, respectively. Configuration 2 spatially is nearer to the compatible point than other configurations and configuration 3 is the farthest configuration among the others.

Fig. 6(a)–(d) plots the evolution of the probability vector of LA in cell 2 of $ICLA_{3,3}$ by starting from (a) configuration 1, (b) configuration 2, (c) configuration 3, and (d) configuration 4. Fig. 7(a)–(d) plots the evolution of the action probability for the selected actions of learning automata using the initial configurations of Table VI. As it can be seen from Fig. 7, regardless of the initial configuration, ICLA converges to the same configuration. Fig. 7(a), (c), and (d) shows that the configuration, ICLA is converged to it, is [(0, 1, 0), (0, 0, 1), (0, 0, 1)] which is the compatible point of Table V.

B. Channel Assignment Problem

In this section, we consider the application of the proposed local rule in ICLA for channel assignment problem. First, a brief introduction to this problem is provided and then mapping the problem to ICLA is explained. Finally, results of the

TABLE V
RESPONSE MATRIX FOR $ICLA_{3,3}$

Reward Probability for (a_1, a_2 , a_3)		*a_2=Low*	*a_2= Normal*	*a_2= High*
a_1=Low	*a_3= Low*	(0.51, 0.81, 0.38)	(0.36, 0.25, 0.84)	(0.53, 0.44, 0.32)
	a_3= Normal	(0.51, 0.76, 0.52)	(0.36, 0.47, 0.39)	(0.53, 0.25, 0.57)
	a_3= High	(0.51, 0.45, 0.77)	(0.36, 0.66, 0.71)	(0.53, 0.36, 0.90)
a_1=Normal	*a_3= Low*	(0.87, 0.32, 0.38)	(0.65, 0.90, 0.84)	(0.92, 0.51, 0.32)
	a_3= Normal	(0.87, 0.61, 0.52)	(0.65, 0.54, 0.39)	(0.92, 0.75, 0.57)
	a_3= High	(0.87, 0.71, 0.77)	(0.65, 0.82, 0.71)	**(0.92, 0.95, 0.90)**
a_1=High	*a_3= Low*	(0.44, 0.37, 0.38)	(0.73, 0.61, 0.84)	(0.71, 0.65, 0.32)
	a_3= Normal	(0.44, 0.49, 0.52)	(0.73, 0.37, 0.39)	(0.71, 0.51, 0.57)
	a_3= High	(0.44, 0.56, 0.77)	(0.73, 0.80, 0.71)	(0.71, 0.73, 0.90)

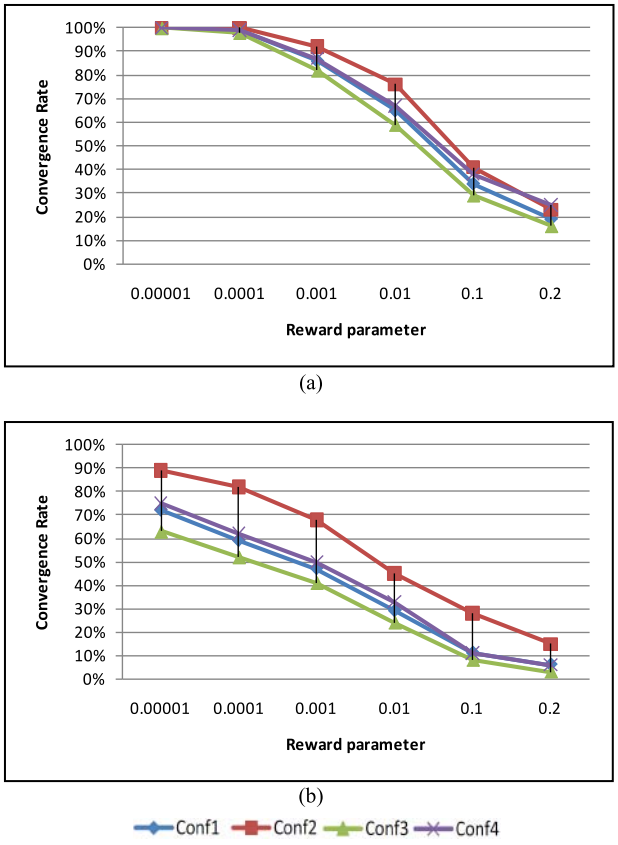

Fig. 5. Comparing convergence rate by starting from different initial configurations using the (a) proposed local rule and (b) ordinary local rule.

TABLE VI
INITIAL CONFIGURATIONS

(Low, Normal, High)	**Initial Configuration 1**
Probability Vector of LA 1	(0.33, 0.33, 0.34)
Probability Vector of LA 2	(0.33, 0.33, 0.34)
Probability Vector of LA 3	(0.33, 0.33, 0.34)
(Low, Normal, High)	**Initial Configuration 2**
Probability Vector of LA 1	(0.1, 0.7, 0.2)
Probability Vector of LA 2	(0.2, 0.2, 0.6)
Probability Vector of LA 3	(0.05, 0.05, 0.9)
(Low, Normal, High)	**Initial Configuration 3**
Probability Vector of LA 1	(0.6, 0.2, 0.1)
Probability Vector of LA 2	(0.2, 0.7, 0.1)
Probability Vector of LA 3	(0.35, 0.5, 0.15)
(Low, Normal, High)	**Initial Configuration 4**
Probability Vector of LA 1	(0.3, 0.5, 0.2)
Probability Vector of LA 2	(0.6, 0.2, 0.2)
Probability Vector of LA 3	(0.2, 0.1, 0.7)

TABLE VII
CONVERGENCE RATE (DIFFERENT INITIAL CONFIGURATIONS)

Reward Parameter		**E-05**	**E-04**	**E-03**	**E-02**	**E-01**	**2.E-01**
Conf 1	**Proposed**	%100	%99	%86	%65	%34	%19
	Ordinary	%72	%59	%47	%29	%11	%6
Conf 2	**Proposed**	%100	%100	%92	%76	%41	%23
	Ordinary	%89	%82	%68	%45	%28	%15
Conf 3	**Proposed**	%100	%98	%82	%59	%29	%16
	Ordinary	%63	%52	%41	%24	%8	%3
Conf 4	**Proposed**	%100	%99	%87	%67	%38	%25
	Ordinary	%75	%62	%50	%33	%11	%6

conducted experiment are given to show the superiority of the proposed local rule.

1) Cellular Networks: A cellular network [1] is constituted from multiple small regions called cells. These cells form a covered area by the network. There is a base station (BS) located in center of each cell which serves the area covered by the cell. There is a switching center which connects the BSs to each other. This switching center also operates as a gateway that connects the network to the wired networks. A node uses a wireless connection to communicate with other nodes of the network through the BS of its cell. We assume that channel access method used here to be frequency division multiple access (FDMA). FDMA gives users an individual allocation of a frequency band (channel). A dedicated channel or frequency band is needed for each connection. If a channel is used by two or more connections at the same time in the same cell or in the neighboring cells, the signals of connections will interfere with each others. These interferences are called co-channel interference. However, two nonadjacent cells that their signals do not interfere with each other can use same channel at the same time.

Assignment of channels to connections is called channel assignment problem. A constraint matrix C is used to illustrate the required gap between assigned channels of cells to have interference-less assignment. For example, element $c(u, v)$ shows the minimum required gap between assigned channels of cells u and v. Co-channel reuse distance is the minimum distance at which a channel can be reused without interference. A cluster is set of all neighboring cells that are in co-channel interference range of each other. To have an interference-less assignment, at most one connection in a cluster can use

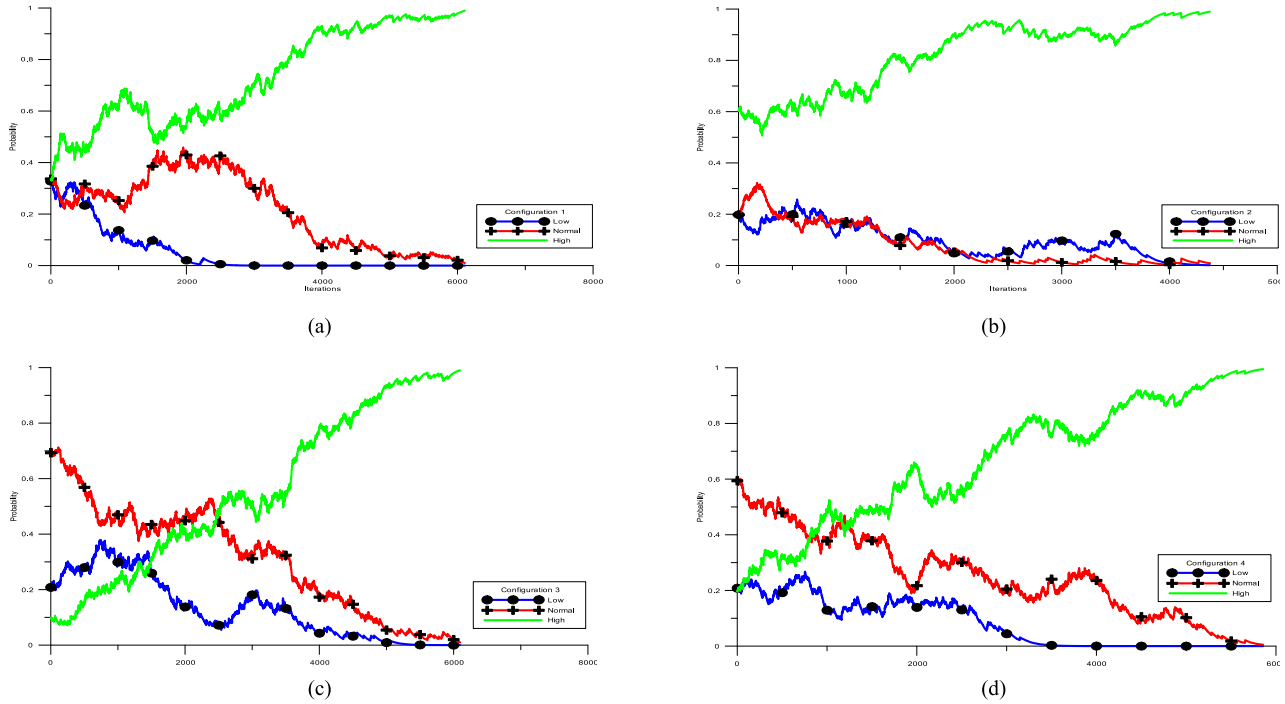

Fig. 6. Evolution of the probability vector of LA2 in $\text{ICLA}_{3,3}$ by starting from (a) conf 1, (b) conf 2, (c) conf 3, and (d) conf 4.

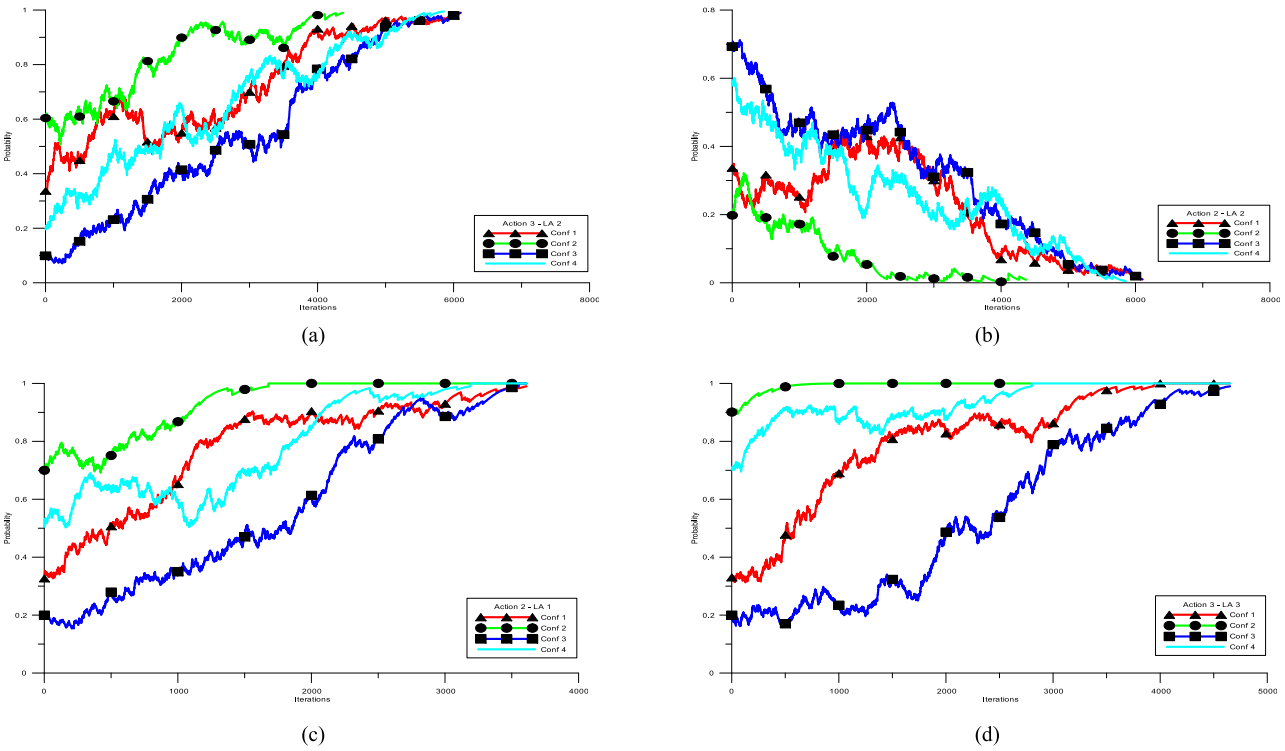

Fig. 7. Evolution of probability of (a) action 3 of LA2, (b) action 2 of LA 2, (c) action 2 of LA 1, and (d) action 3 of LA 3 by starting from different initial configurations when (normal, high, high), which is equivalent to configuration $[(0, 1, 0), (0, 0, 1), (0, 0, 1)]$, is a compatible point.

a channel. There are three common approaches for channel assignment problem: 1) fixed channel assignment (FCA); 2) dynamic channel assignment (DCA); and 3) hybrid channel assignment (HCA). ICLA can be employed to offer FCA, DCA or HCA solutions [8] but in this paper for simplicity purposes, we consider just an FCA solution.

2) Mapping Channel Assignment Problem to ICLA: FCA allocates a set of channels to each cell permanently. Note that these channels can be used in another cell with enough distance again. When a request arrives at any cell, the BS of the cell allocates a free channel to the request. If all channels of this cell are busy, then the BS blocks the request. To offer

```
1   Initialize the ICLA.
2   While interference for assignment is not found do
3     for every cell in the ICLA concurrently do
4       for every LA i in cell u concurrently do
5           choose an action.
            Let j be chosen action of this LA.
6           if channel j doesn't interfere with channels
              used in the neighboring cells then
7               reward the action j of LA i in cell u
8         end if
9       end for
10    end for
11  end while
```

Fig. 8. FCA algorithm.

an FCA approach, first, it should be determined how many channels per cell are needed to support incoming requests. This can be done by calculation of the expected traffic load of the network. Then the required channels must be assigned to each cell such that it avoids interference among the neighboring cells. Considering the features of the channel assignment problem, ICLA is a good candidate for solving this problem in cellular networks. In our proposed approach, ICLA evolves until it reaches a compatible point which is an FCA solution for the channel assignment problem. It is assumed that estimation for demand is given *a priori*. Let $\hat{d}_v$ denote the expected number of required channels for cell v. Now assume a cellular network with n cells and total of m available channels. This network can be modeled as an ICLA with n cells, where there are $\hat{d}_v$ learning automata in cell v. Each LA has m actions to choose (one action per available channel) and the learning algorithm is L_{R-I}. All the cells in interference area of a cell are considered to be neighboring cells of that cell in ICLA. Let to put a label with superscript j ($j = 1, \ldots, \overline{m}_{vi}$) on all neighbors of the i th LA located in cell v (LA_{vi}). $\overline{m}_{vi}$ is total number of LAs which are located in cell v or neighboring cells to v. Furthermore let $(a^1, \ldots, a^j, \ldots, a^{\overline{m}_{vi}})$ denotes joint actions of LA_{vi} and its neighbors. To reward or punish the selected action of LA_{vi} according to the proposed local rule we need value of $\mathrm{Er}^t_{vi}(a^1, \ldots, a^{\overline{m}_{vi}})$ [see (8)]. To calculate $r^t_{vi}(a^1, \ldots, a^{\overline{m}_{vi}})$ we have used the constraint matrix C as illustrated by (13). $C_{\max}$ is maximum value among the elements of the constraint matrix C

$$r^t_{vi}\left(a^1, \ldots, a^{\overline{m}_{vi}}\right) = \sum_{k=1}^{\overline{m}_{vi}-1} \sum_{j=k+1}^{\overline{m}_{vi}} \left(C_{\max} - C\left(a^k, a^j\right)\right). \quad (13)$$

In the following section, we repeat the experiments of [8] for FCA approach and compare results of the proposed local rule with the results of the proposed local rule in FCA algorithm in [8]. In following the local rule of this algorithm is called the ordinary local rule.

3) Simulation Results: To illustrate the superiority of the proposed local rule, we give its results for a simplified version of Philadelphia problem. The Philadelphia problem is

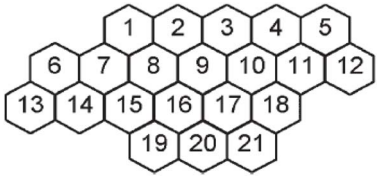

Fig. 9. Regular grid of Philadelphia problem.

TABLE VIII
DEMAND VECTORS FOR PHILADELPHIA PROBLEM

Cell	1	2	3	4	5	6	7
Modified Vector 1	1	1	1	1	1	1	1
Modified Vector 2	1	1	1	1	1	2	2
Cell	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Modified Vector 1	1	1	1	1	1	1	1
Modified Vector 2	2	2	2	2	2	2	2
Cell	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Modified Vector 1	1	1	1	1	1	1	1
Modified Vector 2	1	1	2	2	2	2	2

a channel assignment problem based on a realistic cellular mobile network covering this city. We use an FCA algorithm which is proposed in [8]. Fig. 8 shows this algorithm. To apply the proposed local rule to this algorithm we must use the reward (step 7) to calculate Er_i in (7). Then reward or punishment for the action j is obtained using Algorithm 2. In following, we compare results of the proposed local rule with results of the ordinary local rule in this algorithm (see lines 6 and 7).

In simplified version of Philadelphia problem, the interference $c(i, j)$ is defined as follows:

$$c(i,j) = \begin{cases} 2, & \text{if } i = j \\ 1, & \text{if } d(i,j) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where $c(i, j) = 0$ means there is no interference between two cells i and j. Covering network in Philadelphia problem is similar to a regular grid with 21 cells as shown in Fig. 9. If we map this grid to an ICLA then each cell and its six neighboring cells, which constitute a cluster, define neighboring cells in ICLA. Moreover, when there are multiple learning automata in each cell, all those are neighbors of each other as well. Similar to [8] we consider two demand vectors which is given in Table VIII.

Now we aim to check number of interfering channels for demand vector 1 using the proposed local rule versus ordinary local rule. Due to randomness of the demands for connections, conducting the described experiment in [8], every time may lead to different number of interfering channels. Therefore for the purpose of better evaluation, we repeated the mentioned experiments for demand vectors 1 and 2 thousand times and compared the average results. Fig. 10 show average number of interfering channels for demand vector 1. As illustrated in Fig. 10(a), by the existence of 4 channels, the average number of interfering channels using the proposed local rule is less than what it is using the ordinary local rule. Fig. 10(b)–(d) supports the idea of superiority of the proposed

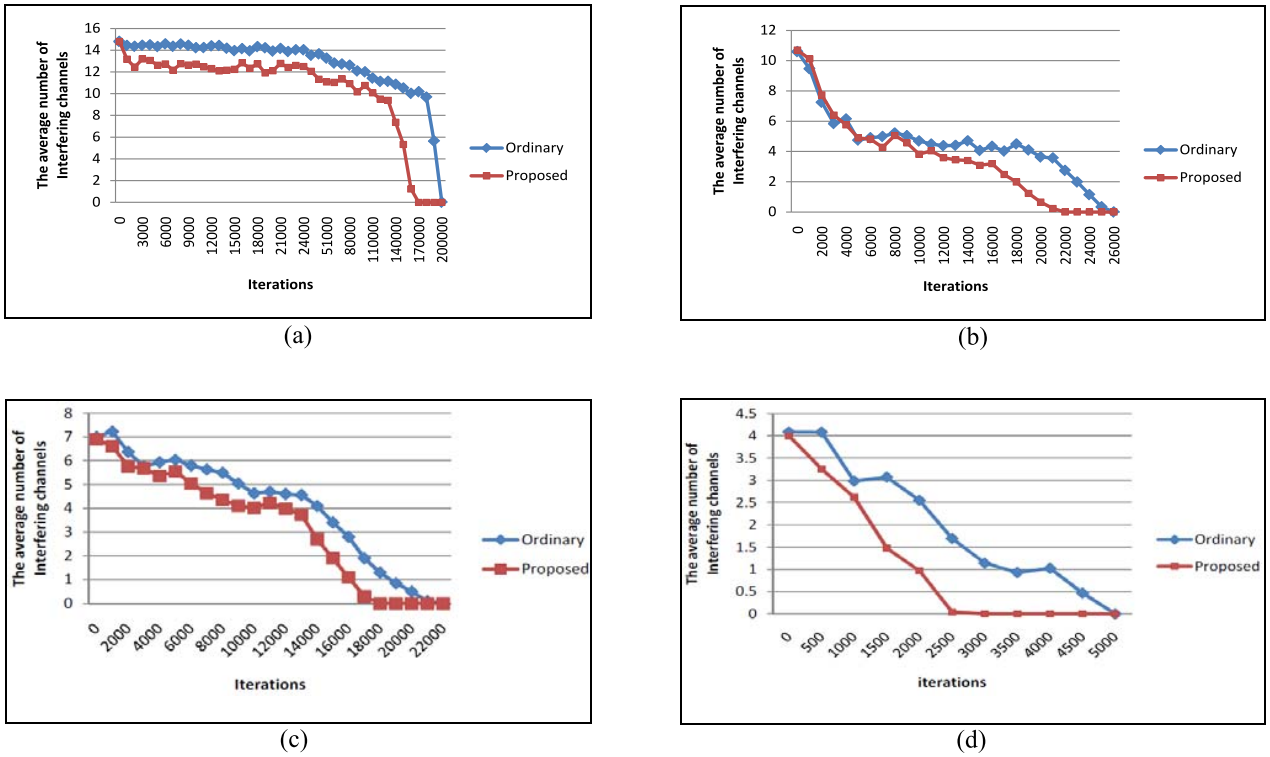

Fig. 10. Comparing the average number of interferences over 1000 times for demand vector 1 with (a) 4 channels, (b) 5 channels, (c) 6 channels, and (d) 7 channels using the proposed local rule and the ordinary local rule.

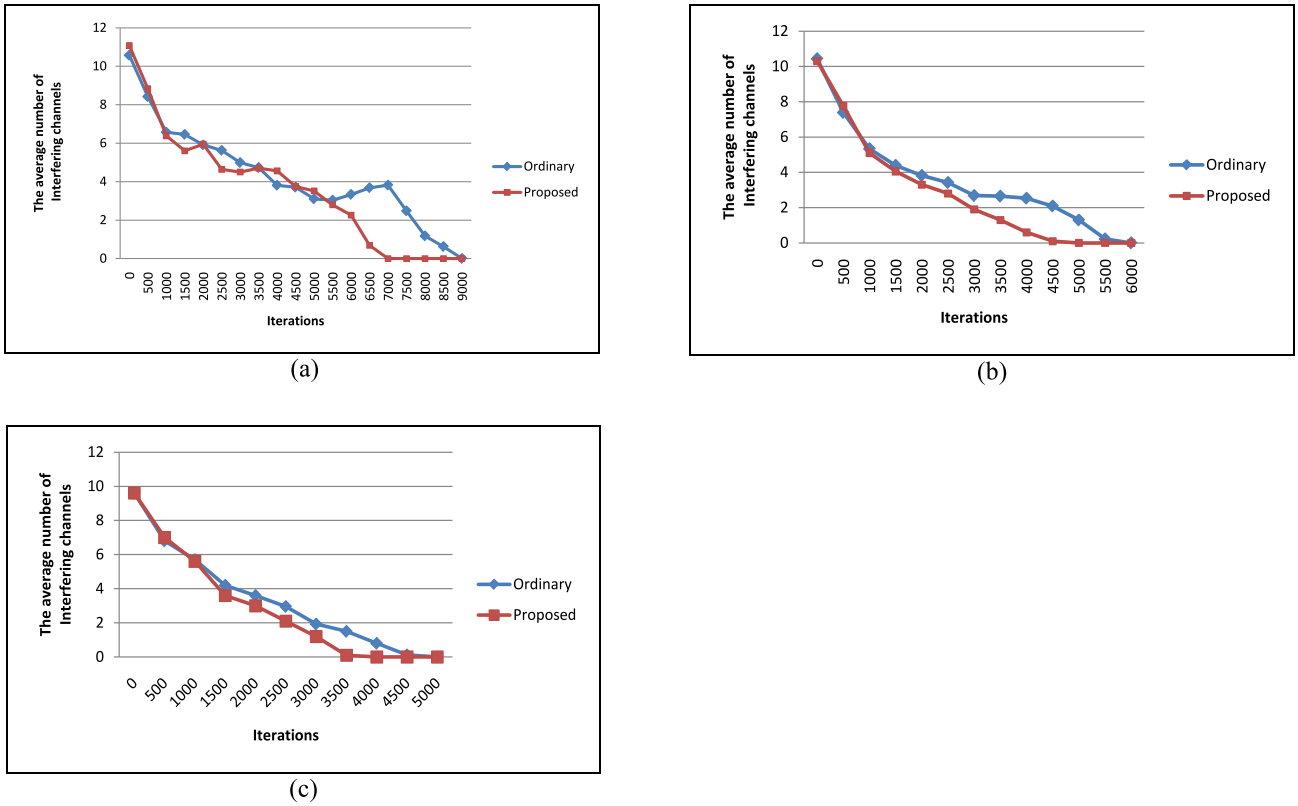

Fig. 11. Comparing the average number of interferences over 1000 times for demand vector 2 with (a) 16 channels, (b) 17 channels, and (c) 18 channels using the proposed local rule and the ordinary local rule.

local rule as well. Using the proposed local rule, ICLA converges faster. For example in Fig. 10(a), the convergence is reached before iteration 170 000 using the proposed local rule, while it takes 200 000 iterations using the ordinary local rule. Faster convergence of ICLA using the proposed local rule is illustrated by Fig. 10(b)–(d) as well. Comparing the four

diagrams of Fig. 10 shows that convergence of ICLA in parts (b)–(d) is before iteration 26 000 which is too faster than what illustrated in Fig. 10(a). Having fewer channels, longer time is needed for convergence. In Fig. 10(a), there exist four channels which is near to the minimum number of the required channels (to have an interfere-less assignment in this problem, at least 3 channels are needed). Fig. 11(a)–(c) shows the experiment results for demand vector 2. The diagrams show that using the proposed local rule, faster convergence, and smaller average number of interfering channels is obtained. These results support the idea of superiority of the proposed local rule.

VI. Conclusion

ICLA is a powerful mathematical model for decentralized applications. Convergence of ICLA to a compatible point has great importance in studying behavior of ICLA. With a simple local rule which rewards or punishes LAs just based on response of environment and the selected actions of neighbors, convergence of ICLA to a compatible point is not guaranteed. In this paper, we proposed a new local rule which guarantees convergence of ICLA to a compatible point. Formal proofs for existence of a compatible point and convergence are provided. We provided different experiments to show usefulness and superiority of the proposed local rule. The obtained results from experiments support our idea about superiority of the proposed local rule against the ordinary local rule. For future works, we aim to study and analysis behavior of ICLA using the proposed local rule under different conditions such as existence of multiple compatible points. Having multiple compatible points, existence of an optimal compatible point and convergence of ICLA to it is an appealing subject. A compatible point is optimal if the obtained reward by each LA in ICLA is the maximum reward that can be obtained among all the other compatible points. Unlike compatible points, existence of an optimal compatible point is not guaranteed always and its existence is application-dependent.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which improved this paper.

References

[1] D. Agrawal and Q. Zeng, *Introduction to Wireless and Mobile Systems*. Toronto, ON, Canada: Thomson Nelson, 2006.

[2] J. A. Torkestani and M. R. Meybodi, "Clustering the wireless ad hoc networks: A distributed learning automata approach," *J. Parallel Distrib. Comput.*, vol. 70, no. 4, pp. 394–405, 2010.

[3] J. A. Torkestani and M. R. Meybodi, "A cellular learning automata-based algorithm for solving the vertex coloring problem," *Expert Syst. Applicat.*, vol. 38, no. 8, pp. 9237–9247, 2011.

[4] M. Esnaashari and M. R. Meybodi, "Irregular cellular learning automata and its application to clustering in sensor networks," in *Proc. 15th Conf. Elect. Eng. (ICEE)*, Tehran, Iran, 2007, pp. 15–17.

[5] A. Barto and P. Anandan, "Pattern-recognizing stochastic learning automata," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 3, pp. 360–375, May/Jun. 1985.

[6] H. Beigy and M. R. Meybodi, "Cellular learning automata based dynamic channel assignment algorithms," *Int. J. Comp. Intel. Appl.*, vol. 8, no. 3, pp. 287–314, 2009.

[7] H. Beigy and M. R. Meybodi, "A mathematical framework for cellular learning automata," *Adv. Complex Syst.*, vol. 7, nos. 3–4, pp. 295–319, 2004.

[8] H. Beigy and M. R. Meybodi, "Cellular learning automata with multiple learning automata in each cell and its applications," *IEEE Trans. Syst., Man, Cybern. B*, vol. 40, no. 1, pp. 54–65, Feb. 2010.

[9] E. A. Billard, "Chaotic behavior of learning automata in multi-level games under delayed information," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Orlando, FL, USA, 1997, pp. 1412–1417.

[10] E. A. Billard, "Asymmetry in learning automata playing multi-level games," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, San Diego, CA, USA, 1998, pp. 2202–2206.

[11] A. E. Eraghi, J. A. Torkestani, and M. R. Meybodi, "Cellular learning automata-based graph coloring problem," in *Proc. Int. Conf. Mach. Learn. Comput. (ICMLC)*, Singapore, 2011, pp. 163–167.

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

[13] M. Esnaashari and M. R. Meybodi, "Deployment of a mobile wireless sensor network with *k*-coverage constraint: A cellular learning automata approach," *Wireless Netw.*, vol. 19, no. 5, pp. 945–968, 2012.

[14] P. Krishna, S. Misra, D. Joshi, A. Gupta, and M. Obaidat, "Secure socket layer certificate verification: A learning automata approach," *Secur. Commun. Netw.*, vol. 7, no. 11, pp. 1712–1718, 2013.

[15] S. Lakshmivarahan and K. Narendra, "Learning algorithms for two-person zero-sum stochastic games with incomplete information," *Math. Oper. Res.*, vol. 6, no. 3, pp. 379–386, 1981.

[16] L. Liu, G. Hu, M. Xu, and Y. Peng, "Learning automata based spectrum allocation in cognitive networks," in *Proc. IEEE Int. Conf. Wireless Commun., Netw. Inf. Secur. (WCNIS)*, Beijing, China, 2010, pp. 503–508.

[17] B. Masoumi and M. Meybodi, "Learning automata based multi-agent system algorithms for finding optimal policies in Markov games," *Asian J. Control*, vol. 14, no. 1, pp. 137–152, 2010.

[18] M. R. Meybodi and M. R. Kharazmi, "Application of cellular learning automata to image processing," *J. Amirkabir*, vol. 14, no. 56A, pp. 1101–1126, 2004.

[19] S. Misra, V. Tiwari, and M. Obaidat, "Lacas: Learning automata-based congestion avoidance scheme for healthcare wireless sensor networks," *IEEE J. Select. Areas Commun.*, vol. 27, no. 4, pp. 466–479, May 2009.

[20] N. Nisan, M. Schapira, G. Valiant, and A. Zohar, "Best-response mechanisms," in *Proc. ICS*, Beijing, China, 2011, pp. 155–165.

[21] A. Rezvanian and M. R. Meybodi, "Finding minimum vertex covering in stochastic graphs: A learning automata approach," *Cybern. Syst.*, vol. 46, no. 8, pp. 698–727, 2015.

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

[23] C. Szepesvári and M. Littman, "A unified analysis of value-function-based reinforcement-learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 2017–2060, 1999.

[24] M. Thathachar and P. Sastry, *Networks of Learning Automata*. Boston, MA, USA: Kluwer Academic, 2004.

[25] D. Thierens, "Adaptive pursuit strategy for allocating operator probabilities," in *Proc. Conf. Genetic Evol. Comput.*, Washington, DC, USA, 2005, pp. 1539–1546.

[26] O. Tilak, R. Martin, and S. Mukhopadhyay, "Decentralized indirect methods for learning automata games," *IEEE Trans. Syst., Man, Cybern. B*, vol. 41, no. 5, pp. 1213–1223, Oct. 2011.

[27] T. A. Tuan, L. C. Tong, and A. B. Premkumar, "An adaptive learning automata algorithm for channel selection in cognitive radio network," in *Proc. IEEE Int. Conf. Commun. Mobile Comput.*, Shenzhen, China, 2010, pp. 159–163.

[28] S. M. Vahidipour, M. R. Meybodi, and M. Esnaashari, "Learning automata-based adaptive Petri net and its application to priority assignment in queuing systems with unknown parameters," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 10, pp. 1373–1384, Oct. 2015.

[29] P. Vrancx, K. Verbeeck, and A. Nowe, "Decentralized learning in markov games," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, no. 4, pp. 976–981, Aug. 2008.

[30] P. Vrancx, K. Verbeeck, and A. Nowé, "Networks of learning automata and limiting games," *Lecture Notes in Computer Science*, vol. 4865, no. 2008, pp. 224–238, 2008.
[31] P. Weber, B. Bordbar, and P. Tino, "A framework for the analysis of process mining algorithms," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 2, pp. 303–317, Mar. 2013.
[32] W. Zhong, Y. Xu, and M. Tao, "Precoding strategy selection for cognitive MIMO multiple access channels using learning automata," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Cape Town, South Africa, 2010, pp. 1–5.

Hossein Morshedlou received the B.Sc. degree in computer engineering from Ferdowsi University, Mashhad, Iran, and the M.Sc. degree in computer engineering from the AmirKabir University of Technology, Tehran, Iran, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree in computer engineering with the AmirKabir University of Technology.

His research interests include distributed artificial intelligence, learning automata, reinforcement learning, parallel algorithms, and soft computing.

Mohammad Reza Meybodi received the B.Sc. and M.Sc. degrees in economics from Shahid Beheshti University, Tehran, Iran, in 1973 and 1977, respectively, the M.Sc. and Ph.D. degrees in computer science from Oklahoma University, Norman, OK, USA, in 1980 and 1983, respectively.

He was an Assistant Professor with Western Michigan University, Kalamazoo, MI, USA, from 1983 to 1985, and an Associate Professor with Ohio University, Athens, OH, USA, from 1985 to 1991. He is currently a Full Professor with the Computer Engineering Department, Amirkabir University of Technology, Tehran. His research interests include wireless networks, fault tolerant systems, learning systems, parallel algorithms, soft computing, and software development.