Irregular Cellular Learning Automata

M. Esnaashari and M. R. Meybodi

***Abstract*— Cellular learning automaton (*CLA*) is a recently introduced model that combines cellular automaton (*CA*) and learning automaton (LA). The basic idea of CLA is to use LA to adjust the state transition probability of stochastic CA. This model has been used to solve problems in areas such as channel assignment in cellular networks, call admission control, image processing and VLSI placement. In this paper, an extension of CLA called irregular cellular learning automaton (*ICLA*) is introduced. This extension is obtained by removing the structure regularity assumption in CLA. Irregularity in the structure of ICLA is needed in some applications, such as computer networks, web mining, and grid computing. The concept of expediency has been introduced for ICLA and then, conditions under which an ICLA becomes expedient are analytically found.**

***Index Terms*— Irregular cellular learning automata, Expediency, Steady-state analysis, Markov process.**

I. Introduction

Cellular automaton (*CA*) is a discrete model consists of simple identical components, called cells, organized into a regular grid structure. Each cell can assume a state from a finite set of states. The operation of a CA takes place in discrete steps according to a local rule, which depends on the local environments of the cells. The local environment of a cell is usually taken to be a small number of neighboring cells, which can include the cell itself [1]. The global state of a CA, which represents the states of all its constituting cells together, is referred to as a configuration. The local rule and the initial configuration of a CA specify the evolution of that CA, which tells how each configuration is changed in one step. CA is particularly suitable for modeling natural systems that can be described as massive collections of simple objects interacting locally with each other [2]. CA is called *cellular*, because it is made up of cells like points in the lattice, and called *automata*, because it follows a simple local rule [3].

On the other hand, learning automaton (*LA*) is, by design, a simple agent for making simple and adaptive decisions in unknown random environments. Intuitively, LA could be considered as a learning organism which tries different actions (from its action set) and selects new actions on the basis of the responses of the environment to previous actions. One attractive feature of this model is that it could be regarded as a simple unit from which complex structures could be constructed. These could be designed to handle complicated learning problems [4]. In most applications, local interaction of learning automata (LAs), which can be defined in a form of graph such as tree, mesh, or array, is more suitable. In [5], CA and LA are combined, and a new model, which is called cellular LA (*CLA*), is obtained. This model, which opens a new learning paradigm, is superior to CA because of its ability to learn and is also superior to single LA because it consists of a collection of LAs interacting with each other. CLA has been used in many different applications including: channel assignment in cellular networks [6], call admission control in cellular networks [7], and VLSI placement [8], to mention a few. In [11], a mathematical framework for studying the behavior of the CLA has been introduced. It was shown that, for a class of rules called commutative rules, different models of CLA converge to a globally stable state [7][9][10][11].

In this paper, irregular CLA (*ICLA*) is introduced as an extension to CLA model, in which the structure regularity assumption is removed. We argue that in some applications, such as computer networks, web mining, and grid computing, problems could usually be described by graphs, with irregular structures, and hence, an extension of CLA with irregular structure is needed to model such problems.

For the proposed extended model, the concept of expediency is introduced. Informally, an ICLA is said to be expedient if, in the long run, all of its constituting learning automata perform better than pure-chance automata. A pure-chance automaton is an automaton which chooses any of its actions by pure chance. Expediency is a notion of learning. An automaton which is capable of learning must do at least better than a pure-chance automaton [12]. The steady-state behavior of ICLA is studied and then, conditions under which an ICLA becomes expedient are given.

The rest of this paper is organized as follows. In Section II, we briefly introduce CLA model. Section III and Section IV present ICLA and its steady-state behavior, respectively. A set of numerical examples are given in Section V for illustrating the theoretical results. In Section VI, we present a case study of using ICLA for problem solving in the area of wireless sensor networks. Section VII is the conclusion.

II. Cellular Learning Automata: A Combination of CA and LA Models

Before presenting CLA model in this section, we first give a brief introduction to CA and LA models.

Submitted for review on November, the 9th, 2013.

M. Esnaashari is with the Information Technology Department, Iran Telecommunications Research Center, Tehran, Iran (e-mail: esnaashari@itrc.ac.ir).

M. R. Meybodi is with the Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran. He is also with the School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran (e-mail: mmeybodi@aut.ac.ir).

A. Cellular Automata

A *d*-dimensional CA consists of an infinite *d*-dimensional lattice of identical cells. Each cell can assume a state from a finite set of states. The cells update their states synchronously on discrete steps according to a local rule. The new state of each cell depends on the previous states of a set of cells, including the cell itself, and constitutes its neighborhood. The states of all cells in the lattice are described by a configuration. A configuration can be described as the state of the whole lattice. The local rule and the initial configuration of CA specify the evolution of CA, that is, how the configuration of CA evolves in time.

B. Learning Automata

Learning automata (LA) is an abstract model which randomly selects one action out of its finite set of actions and performs it on a random environment. Environment then evaluates the selected action and responses to the LA with a reinforcement signal. Based on the selected action, and the received signal, LA updates its internal state and selects its next action.

Environment can be defined by the triple $E = \{\alpha, \beta, c\}$ where $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ represents a finite input set, $\beta = \{\beta_1, \beta_2, ..., \beta_r\}$ represents the output set, and $c = \{c_1, c_2, ..., c_r\}$ is a set of penalty probabilities, where each element c_i=E[β | $\alpha = \alpha_i$] of c corresponds to one input α_i. An environment in which β assumes values in the interval [0, 1] is referred to as an S-model environment. LAs are classified into fixed-structure stochastic, and variable-structure stochastic. In the following, we consider only variable-structure LAs.

A variable-structure LA is defined by the quadruple $\{\alpha, \beta, p, T\}$ in which $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ represents the action set of LA, $\beta = \{\beta_1, \beta_2, \cdots, \beta_r\}$ represents the input set, $p = \{p_1, p_2, \cdots, p_r\}$ represents the action probability set, and finally $p(k+1) = T[\alpha(k), \beta(k), p(k)]$ represents the learning algorithm. This LA operates as follows. Based on the action probability set p, LA randomly selects an action $\alpha(k)$, and performs it on the environment. After receiving the environment's reinforcement signal ($\beta(k)$), LA updates its action probability set based on equation (1) or (2) according to the selected action $\alpha(k)$.

$$p_i(k+1) = p_i(k) + \beta(k)\left[\frac{b}{r-1} - b \cdot p_i(k)\right] - \left[1-\beta(k)\right] \cdot a \cdot p_i(k), \quad when\ \alpha(k) \neq \alpha_i. \tag{1}$$

$$p_i(k+1) = p_i(k) - \beta(k) \cdot b \cdot p_i(k) + \left[1-\beta(k)\right] \cdot a \cdot \left(1 - p_i(k)\right), \quad when\ \alpha(k) = \alpha_i. \tag{2}$$

In the above equations, a and b are reward and penalty parameters respectively. For $a = b$, learning algorithm is called L_{RP} [1], for $b << a$, it is called $L_{R\varepsilon P}$ [2], and for $b = 0$, it is called L_{RI} [3].

Since its introduction in 1973 by Tsetlin [21], learning automata have been found a variety of applications in many different topics [22][23][24][25] and still appears in many recent researches such as [26] and [27] to mention a few.

C. Cellular Learning Automata

A CLA is a CA in which a number of LAs is assigned to every cell. Each LA residing in a particular cell determines its action (state) on the basis of its action probability vector. Like CA, there is a local rule that CLA operates under. The local rule of CLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to that LA. The neighboring LAs (cells) of any particular LA (cell) constitute the local environment of that LA (cell). The local environment of an LA (cell) is non-stationary due to the fact that the action probability vectors of the neighboring LAs vary during the evolution of CLA. The operation of a CLA could be described as the following steps: At the first step, the internal state of every cell is determined on the basis of the action probability vector of the LA residing in that cell. In the second step, the local rule of CLA determines the reinforcement signal to the LA residing in that cell. Finally, each LA updates its action probability vector based on the supplied reinforcement signal and the chosen action. This process continues until the desired result is obtained. Formally, a CLA can be defined as follows:

Definition 1. A *d*-dimensional cellular learning automata is a structure $\mathcal{A} = (Z^d, N, \Phi, A, \mathcal{F})$, where

- Z^d is a lattice of *d*-tuples of integer numbers.
- $N = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m\}$ is a finite subset of Z^d called neighborhood vector, where $\bar{x}_i \in Z^d$.
- Φ is a finite set of states. The state of the cell c_i is denoted by φ_i.
- A is the set of learning automata each of which is assigned to one cell of the CLA.
- $F^i: \underline{\underline{\varphi_i}} \rightarrow \underline{\beta}$ is the local rule of the CLA in each cell c_i, where $\underline{\beta}$ is the set of values that the reinforcement signal can take. It computes the reinforcement signal for each learning automaton based on the actions selected by the neighboring learning automata.

III. ICLA: An Extension to CLA Model

Irregular cellular learning automaton (Fig. 1) is a generalization of CLA in which the restriction of regular structure is removed. An ICLA is defined as an undirected graph in which, each vertex represents a cell and is equipped with an LA, and each edge induces an adjacency relation between two cells (two LAs). LA residing in a particular cell determines its state (action) according to its action probability vector. Like CLA, there is a rule that ICLA operates under.

[1] Linear Reward-Penalty
[2] Linear Reward epsilon Penalty
[3] Linear Reward Inaction

The rule of ICLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to that LA. The neighboring LAs of any particular LA constitute the local environment of that LA. The local environment of an LA is non-stationary because the action probability vectors of the neighboring LAs vary during the evolution of ICLA.

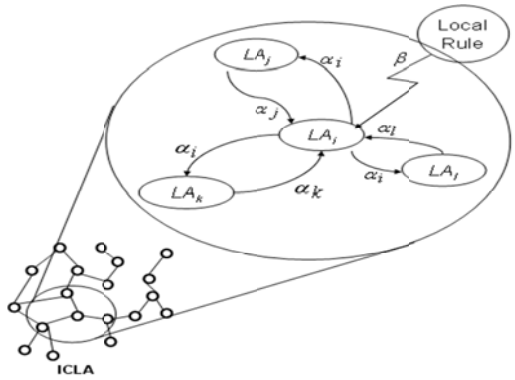


Fig. 1. Irregular cellular learning automaton

The operation of ICLA is similar to the operation of CLA. At the first step, the internal state of each cell is specified on the basis of the action probability vector of the LA residing in that cell. In the second step, the rule of ICLA determines the reinforcement signal to the LA residing in each cell. Finally, each LA updates its action probability vector on the basis of the supplied reinforcement signal and the internal state of the cell. This process continues until the desired result is obtained. Formally, an ICLA is defined as given below.

Definition 2. Irregular cellular learning automaton is a structure $\mathcal{A}=(G<E,V>,\Phi,A,\mathcal{F})$, where

- G is an undirected graph, with V as the set of vertices (cells) and E as the set of edges (adjacency relations).
- Φ is a finite set of states. The state of the cell c_i is denoted by φ_i.
- A is the set of LAs each of which is assigned to one cell of ICLA.
- $\mathcal{F}^i : \underline{\varphi}_i \rightarrow \underline{\beta}$ is the local rule of ICLA in the cell c_i, where $\underline{\varphi}_i = \{\varphi_j \mid \{i,j\} \in E\} \cup \{\varphi_i\}$ is the set of states of all neighbors of c_i and $\underline{\beta}$ is the set of values that the reinforcement signal can assume. Local rule gives the reinforcement signal to each LA from the current actions selected by the neighboring LAs of that LA.

Comparing the above definition with the definition of CLA, the only existing difference is that the lattice Z^d and the neighborhood vector N in CLA are replaced by the undirected graph $G<E, V>$ in ICLA. That is, instead of having a regular lattice structure, ICLA has an irregular graph-based structure. Note that in the definition of ICLA, no explicit definition is given for the neighborhood of each cell. It is implicitly defined in the definition of the graph G.

In what follows, we consider ICLA with n cells. The learning automaton LA_i which has a finite action set $\underline{\alpha}_i$ is associated to cell c_i (for i=1, 2, …, n) of ICLA. Let the cardinality of $\underline{\alpha}_i$ be m_i.

The operation of ICLA takes place as the following iterations. At iteration k, each learning automaton selects an action. Let $\alpha_i \in \underline{\alpha}_i$ be the action selected by LA_i. Then all learning automata receive a reinforcement signal. Let $\beta_i \in \underline{\beta}$ be the reinforcement signal received by LA_i. This reinforcement signal is produced by the application of the local rule $\mathcal{F}^i(\underline{\varphi}_i) \rightarrow \underline{\beta}$. Higher values of β_i mean that the selected action of LA_i will receive higher penalties. Then, each LA_i updates its action probability vector on the basis of the supplied reinforcement signal and its selected action α_i.

Like CLA, ICLA can be either synchronous or asynchronous and an asynchronous ICLA can be either time-driven or step-driven. ICLA have been successfully used for problem solving in the area of wireless sensor networks [16][17][18][19].

A. Evolution of ICLA

Definition 3. A configuration of ICLA at step k is denoted by $\underline{p}(k)=(\underline{p}_1,\underline{p}_2,\ldots,\underline{p}_n)^T$, where $\underline{p}_i$ is the action probability vector of the learning automaton LA_i and T denotes the transpose operator.

Definition 4. A configuration $\underline{p}$ is called deterministic if the action probability vector of each learning automaton is a unit vector, otherwise it is called probabilistic. Hence, the set of all deterministic configurations, $\mathcal{K}^*$, and the set of probabilistic configurations, $\mathcal{K}$, in ICLA are

$$\mathcal{K}^* = \left\{ \underline{p} \;\middle|\; \begin{array}{r} \underline{p}=(\underline{p}_1,\underline{p}_2,\ldots,\underline{p}_n)^T, \underline{p}_i=(p_{i1},\ldots,p_{im_i})^T, \\ \forall y,i: p_{iy} \in \{0,1\}, \forall i: \sum_y p_{iy}=1 \end{array} \right\}. \quad (3)$$

and

$$\mathcal{K} = \left\{ \underline{p} \;\middle|\; \begin{array}{r} \underline{p}=(\underline{p}_1,\underline{p}_2,\ldots,\underline{p}_n)^T, \underline{p}_i=(p_{i1},\ldots,p_{im_i})^T, \\ \forall y,i: 0 \le p_{iy} \le 1, \forall i: \sum_y p_{iy}=1 \end{array} \right\}, \quad (4)$$

respectively.

Lemma 1. $\mathcal{K}$ is the convex hull of $\mathcal{K}^*$.

***Proof*:** Proof of this lemma is given in [11]. ■

The application of the local rule to every cell allows transforming a configuration to a new one.

Definition 5. The global behavior of an ICLA is a mapping $\mathcal{G}: \mathcal{K} \rightarrow \mathcal{K}$ that describes the dynamics of ICLA. The evolution of ICLA from a given initial configuration $\underline{p}(0) \in \mathcal{K}$ is a sequence of configurations $\{\underline{p}(k)\}_{k \ge 0}$, such that $\underline{p}(k+1)=\mathcal{G}(\underline{p}(k))$.

Definition 6. Neighborhood set of any particular LA_i, denoted by $N(i)$, is defined as the set of all learning automata residing

in the adjacent cells of the cell c_i, that is,

$$N(i)=\{LA_j \mid \{i,j\}\in E\}. \quad (5)$$

Let $\mathcal{N}_i$ be the cardinality of $N(i)$.

Definition 7. The average penalty for action r of learning automaton LA_i in configuration $\underline{p}\in\mathcal{K}$ is defined as

$$d_{ir}(\underline{p})=\mathrm{E}[\beta_i \mid \underline{p},\alpha_i=r]= \sum_{y_{j_1},\ldots,y_{j_{\mathcal{N}_i}}} \mathcal{F}^i(y_{j_1},\ldots,y_{j_{\mathcal{N}_i}},r)\prod_{LA_l\in N(i)} p_{ly_{j_l}}, \quad (6)$$

and the average penalty for the learning automaton LA_i is defined as

$$D_i(\underline{p})=\mathrm{E}[\beta_i \mid \underline{p}]=\sum_y d_{iy}(\underline{p})p_{iy}. \quad (7)$$

The above definition implies that if the learning automaton LA_j is not a neighboring learning automaton for LA_i, then $d_{ir}(\underline{p})$ does not depend on $\underline{p}_j$. We assume that $d_{ir}(\underline{p})\neq 0$ for all i, r and $\underline{p}$, that is, in any configuration, any action has a non-zero chance of receiving penalty.

Definition 8. The total average penalty for ICLA at configuration $\underline{p}\in\mathcal{K}$ is the sum of the average penalties for all learning automata in ICLA, that is,

$$D(\underline{p})=\sum_i D_i(\underline{p}). \quad (8)$$

IV. Behavior of ICLA

In this section, we will study the asymptotic behavior of an ICLA, in which all learning automata use the L_{RP} learning algorithm [13], when operating within an S-model environment. We refer to such an ICLA as ICLA with SL_{RP} learning automata hereafter. The process $\{\underline{p}(k)\}_{k\geq 0}$ which evolves according to the L_{RP} learning algorithm can be described by the following difference equation:

$$\underline{p}(k+1)=\underline{p}(k)+\underline{a}\cdot\underline{g}(\underline{p}(k),\underline{\beta}(k)), \quad (9)$$

where $\underline{\beta}(k)$ is composed of components $\beta_{iy}(k)$ (for $1\leq i\leq n$, $1\leq y\leq m_i$, and $\beta_{iy_1}=\beta_{iy_2}$ for every y_1, y_2 such that $1\leq y_1,y_2\leq m_i$), which are dependent on $\underline{p}(k)$. $\underline{a}$ is an $n\times n$ diagonal matrix with $a_{ii}=a_i$ and a_i represents the learning parameter for learning automaton LA_i. $\underline{g}$ represents the learning algorithm, whose components can be obtained using L_{RP} learning algorithm in S-model environment as follows:

$$\underline{g}_{ir}(p_{ir},\beta_{ir})=\begin{cases}(1-p_{ir}-\beta_{ir}); & \alpha_i=r\\ \left(\dfrac{\beta_{ir}}{m_i-1}-p_{ir}\right); & \alpha_i\neq r\end{cases}. \quad (10)$$

From equation (9) it follows that $\{\underline{p}(k)\}_{k\geq 0}$ is a discrete-time Markov process [14] defined on the state space $\mathcal{K}$ (given by equation (4)). Let $(\mathcal{K},d')$ be a metric space, where d' is the metric defined according to equation (11) as given below:

$$d'(\underline{p},\underline{q})=\sum_i \|\underline{p}_i-\underline{q}_i\|, \quad (11)$$

where $\|\underline{X}\|$ stands for the norm of the vector $\underline{X}$.

Lemma 2 and Lemma 3, given below, state some properties of the Markovian process given by equation (9).

Lemma 2. The Markovian process given by equation (9) is strictly distance diminishing.

***Proof*:** To prove this lemma, we will show that the Markovian process given by equation (9) follows the definition of the strictly distance diminishing processes given in [15] by Norman. The complete proof is given in Appendix A. ■

Corollary 1. Let $\underline{p}^{(h)}$ denotes $\underline{p}(k+h)$ when $\underline{p}(k)=\underline{p}$ and $\underline{q}^{(h)}$ denotes $\underline{p}(k+h)$ when $\underline{p}(k)=\underline{q}$. Then $\underline{p}^{(h)}\to\underline{q}^{(h)}$ as $h\to\infty$ irrespective of the initial configurations $\underline{p}$ and $\underline{q}$.

***Proof*:** The proof is given in Appendix A. ■

Lemma 3. The Markovian process given by equation (9) is ergodic.

Proof: To prove the lemma we can see that the Markovian process given by equation (9) has the following two properties:

- There are no absorbing states for $\{\underline{p}(k)\}$, since there is no $\underline{p}$ that satisfies $\underline{p}(k+1)=\underline{p}(k)$.
- The proposed process is strictly distance diminishing (Lemma 2).

From the above two properties and considering the results given in corollary 1, we can conclude that the Markovian process $\{\underline{p}(k)\}_{k\geq 0}$ is ergodic. ■

Now define

$$\Delta\underline{p}(k)=\mathrm{E}[\underline{p}(k+1)\mid\underline{p}(k)]-\underline{p}(k). \quad (12)$$

Since $\{\underline{p}(k)\}_{k\geq 0}$ is Markovian and $\underline{\beta}(k)$ depends only on $\underline{p}(k)$ and not on k explicitly, then $\Delta\underline{p}(k)$ can be expressed as a function of $\underline{p}(k)$. Hence we can write

$$\Delta\underline{p}=\underline{a}\underline{f}(\underline{p}). \quad (13)$$

The components of $\Delta\underline{p}$ can be obtained as follows:

$$\Delta p_{ir}=a_i p_{ir}\cdot[1-p_{ir}-\mathrm{E}[\beta_{ir}]]- a_i\sum_{y\neq r}p_{iy}\cdot\left[\frac{1}{m_i-1}\mathrm{E}[\beta_{iy}]-p_{ir}\right]$$

$$
\begin{aligned}
&= a_i \cdot \left[\frac{1}{m_i - 1} \sum_{y \neq r} p_{iy} \mathrm{E}\left[\beta_{iy}\right] - p_{ir} \mathrm{E}\left[\beta_{ir}\right] \right] \\
&= a_i \cdot \left[\frac{1}{m_i - 1} \sum_{y \neq r} p_{iy} d_{iy}\left(\underline{p}\right) - p_{ir} d_{ir}\left(\underline{p}\right) \right] \\
&= a_i f_{ir}\left(\underline{p}\right),
\end{aligned} \tag{14}
$$

where

$$
\begin{aligned}
f_{ir}\left(\underline{p}\right) &= \frac{1}{m_i - 1} \sum_{y \neq r} p_{iy} d_{iy}\left(\underline{p}\right) - p_{ir} d_{ir}\left(\underline{p}\right) \\
&= \frac{1}{m_i - 1} \sum_{y} p_{iy} d_{iy}\left(\underline{p}\right) - \left(1 + \frac{1}{m_i - 1}\right) \cdot \\
&\qquad \left[p_{ir} d_{ir}\left(\underline{p}\right) \right] \\
&= \frac{1}{m_i - 1} \cdot \left[D_i\left(\underline{p}\right) - m_i p_{ir} d_{ir}\left(\underline{p}\right) \right].
\end{aligned} \tag{15}
$$

Lemma 4. Function $\underline{f}\left(\underline{p}\right)$ whose components are given by equation (15) is Lipschitz continuous over the compact space $\mathcal{K}$.

***Proof*:** Function $\underline{f}\left(\underline{p}\right)$ has compact support (it is defined over $\mathcal{K}$), is bounded because $-1 \le f_{ir}\left(\underline{p}\right) \le 1$ for all $\underline{p}$, i, r, and is also continuously differentiable with respect to $\underline{p}$ over $\mathcal{K}$. Therefore, its first derivative with respect to $\underline{p}$ is also bounded. Thus, using the Cauchy's mean value theorem, it can be concluded that $\underline{f}\left(\underline{p}\right)$ is Lipschitz continuous over the compact space $\mathcal{K}$ with Lipschitz constant $\mathrm{K} = \sup_{\underline{p}} \left\| \nabla_{\underline{p}} \underline{f}\left(\underline{p}\right) \right\|$. ■

For different values of $\underline{a}$, equation (9) generates different processes and we shall use $\underline{p}^{\underline{a}}\left(k\right)$ to denote this process whenever the value of $\underline{a}$ is to be specified explicitly. To find the approximating ODE for the learning algorithm given by (9), we define a sequence of continuous-time interpolation of (9), denoted by $\tilde{\underline{p}}^{\underline{a}}\left(t\right)$ and called an *interpolated process*, whose components are defined by:

$$
\tilde{\underline{p}}_i^{\,\underline{a}}\left(t\right) = \underline{p}_i\left(k\right), \quad t \in \left[k a_i, \left(k+1\right) a_i\right), \tag{16}
$$

where a_i is the learning parameter of the L_{RP} algorithm for learning automaton LA_i. The interpolated process $\left\{\tilde{\underline{p}}^{\underline{a}}\left(t\right)\right\}_{t \ge 0}$ is a sequence of random variables that takes values from $\mathcal{R}^{m_1 \times \ldots \times m_n}$, where $\mathcal{R}^{m_1 \times \ldots \times m_n}$ is the space of all functions that, at each point, are continuous on the right and have a limit on the left over $\left[0, \infty\right)$ and take values in $\mathcal{K}$, which is a bounded subset of $\mathcal{R}^{m_1 \times \ldots \times m_n}$. Consider the following ordinary differential equation (ODE):

$$
\dot{\underline{p}} = \underline{f}\left(\underline{p}\right), \tag{17}
$$

where $\dot{\underline{p}}$ is composed of the following components:

$$
\frac{dp_{ir}}{dt} = \frac{1}{m_i - 1} \cdot \left[D_i\left(\underline{p}\right) - m_i p_{ir} d_{ir}\left(\underline{p}\right) \right]. \tag{18}
$$

In the following theorem, we will show that equation (9) is the approximation to the ODE (17). This means that if we have the solution to (17), then we can obtain information regarding the behavior of $\underline{p}\left(k\right)$.

Theorem 1. Using the learning algorithm (10) and considering $\max\left\{\underline{a}\right\} \to 0$, $\underline{p}\left(k\right)$ is well approximated by the solution of the ODE (17).

***Proof*:** Following conditions are satisfied by the learning algorithm given by equation (10):

- $\left\{\underline{p}\left(k\right)\right\}_{k \ge 0}$ is a Markovian process.
- Given $\underline{p}\left(k\right)$, $\underline{\alpha}\left(k\right)$ and $\underline{\beta}\left(k\right)$ are independent of $\underline{\alpha}\left(k-1\right)$ and $\underline{\beta}\left(k-1\right)$.
- $\underline{f}\left(\underline{p}\left(k\right)\right)$ is independent of k.
- $\underline{f}\left(\underline{p}\left(k\right)\right)$ is Lipshitz continuous over the compact space $\mathcal{K}$ (Lemma 4).
- $\mathrm{E}\left\| \underline{g}\left(\underline{p}\right) - \underline{f}\left(\underline{p}\right) \right\|^2$ is bounded since $\underline{g}\left(\underline{p}\right) \in \left[-1, 1\right]^{m_1 \times \ldots \times m_n}$.
- Learning parameters a_i, i=1, …, n are sufficiently small since $\max\left\{\underline{a}\right\} \to 0$.

Therefore, using theorem (A.1) in [4], we can conclude the theorem. ■

Equation (9) is the so called Euler approximation to the ODE (17). Specifically, if $\underline{p}\left(k\right)$ is a solution to (9) and $\tilde{\underline{p}}\left(t\right)$ is a solution to (17), then for any $T > 0$, we have

$$
\lim_{a_i \to 0} \sup_{0 \le k \le T/a_i} \left\| \underline{p}_{ir}\left(k\right) - \tilde{\underline{p}}\left(k a_i\right) \right\| = 0. \tag{19}
$$

What Theorem 1 says is that $\underline{p}\left(k\right)$, given by equation (9), will closely follow the solution of the ODE (17), that is, $\underline{p}\left(k\right)$ can be made to closely approximate the solution of its approximating ODE by taking $\max\left\{\underline{a}\right\}$ sufficiently small. Thus, if the ODE (17) has a globally asymptotically stable equilibrium point, then we can conclude that (by taking $\max\left\{\underline{a}\right\}$ sufficiently small), $\underline{p}\left(k\right)$, for large k, would be close to this equilibrium point irrespective of its initial

configuration $\underline{p}(0)$. Therefore, the analysis of the process $\{\underline{p}(k)\}_{k\geq 0}$ is done in two stages. In the first stage, we solve ODE (17) and in the second stage, we characterize the solution of this ODE.

In the following subsections, we first find the equilibrium points of ODE (17), then study the stability property of these equilibrium points, and finally state some theorems about the convergence of ICLA.

A. Equilibrium Points

To find the equilibrium points of ODE (17), we first show that this ODE has at least one equilibrium point and then specify a set of conditions which must be satisfied by a configuration $\underline{p}$ to be an equilibrium point of the ODE (17).

Lemma 5. ODE (17) has at least one equilibrium point.

***Proof*:** To prove this lemma, we first propose a continuous mapping $\zeta(\underline{p})$ from $\mathcal{K}$ to $\mathcal{K}$. Then, using the Brouwer's fixed point theorem, we will show that any continuous mapping from $\mathcal{K}$ to $\mathcal{K}$ has at least one fixed point. Finally, we will show that the fixed point of $\zeta(\underline{p})$ is the equilibrium point of the ODE (17). This indicates that ODE (17) has at least one equilibrium point. The complete proof is given in Appendix B. ■

Theorem 2. The equilibrium points of ODE (17) are the set of configurations $\underline{p}^*$ which satisfy the set of conditions $p^*_{ir} = \dfrac{\prod_{y\neq r} d_{iy}(\underline{p}^*)}{\sum_{y_1}\left(\prod_{y_2\neq y_1} d_{iy_2}(\underline{p}^*)\right)}$ for all i, r.

***Proof*:** To find the equilibrium points of ODE (17), we have to solve equations of the form

$$\frac{dp^*_{ir}(t)}{dt} = 0 \quad for\ all\ i, r. \qquad (20)$$

Using equation (18), (20) can be rewritten as

$$\frac{1}{m_i - 1}\cdot\left[D_i(\underline{p}^*(t)) - m_i p^*_{ir}(t) d_{ir}(\underline{p}^*(t))\right] = 0, \quad for\ all\ i, r. \qquad (21)$$

These equations have solutions of the form:

$$p^*_{ir} = \frac{D_i(\underline{p}^*)}{m_i d_{ir}(\underline{p}^*)}, \quad for\ all\ i, r, \qquad (22)$$

which after some algebraic manipulations, can be rewritten as:

$$p^*_{ir} = \frac{\prod_{y\neq r} d_{iy}(\underline{p}^*)}{\sum_{y_1}\left(\prod_{y_2\neq y_1} d_{iy_2}(\underline{p}^*)\right)}, \quad for\ all\ i, r, \qquad (23)$$

and hence the theorem. ■

It follows from Theorem 2 that the difference equation given by (13) has equilibrium points $\underline{p}^*$ that satisfy the set of conditions $p^*_{ir}(k) = \dfrac{\prod_{y\neq r} d_{iy}(\underline{p}^*(k))}{\sum_{y_1}\left(\prod_{y_2\neq y_1} d_{iy_2}(\underline{p}^*(k))\right)}$ for all i, r.

B. The Stability Property

In this subsection, we characterize the stability of equilibrium configurations of ICLA, that is, the equilibrium points of ODE (17). To do this, the origin is first transferred to an equilibrium point $\underline{p}^*$, and then a candidate for a Lyapunov function is introduced for studying the stability of this equilibrium point. Consider the following transformation:

$$\hat{p}_{ir} = p_{ir} - \frac{D_i(\underline{p}^*)}{m_i d_{ir}(\underline{p}^*)} \quad for\ all\ i, r. \qquad (24)$$

Using this transformation, the origin is transferred to $\underline{p}^*$.

Lemma 6. Derivative of $\underline{\hat{p}}$ with respect to time has components of the following form:

$$\frac{d\hat{p}_{ir}}{dt} = -d_{ir}(\underline{p})\hat{p}_{ir} \quad for\ all\ i, r. \qquad (25)$$

***Proof*:** The proof is given in Appendix C. ■

Corollary 2. $\underline{\hat{p}}_{ir}$ and its time derivative $\dfrac{d\hat{p}_{ir}}{dt}$ have different signs.

***Proof*:** Considering the fact that $d_{ir}(\underline{p}) \geq 0$ for all configurations $\underline{p}$ and for all i and r, the proof is an immediate result of equation (25). ■

Theorem 3. A configuration $\underline{p}^*$, whose components satisfy set of equations $p^*_{ir} = \dfrac{\prod_{y\neq r} d_{iy}(\underline{p}^*)}{\sum_{y_1}\left(\prod_{y_2\neq y_1} d_{iy_2}(\underline{p}^*)\right)}$ for all i, r, is an asymptotically stable equilibrium point of ODE (17) over $\mathcal{K}$.

***Proof*:** To prove this theorem, we first apply transformation (24) to transfer the origin to $\underline{p}^*$. Then we propose a positive definite function $V(\underline{p})$ and show that the time derivative of $V(\underline{p})$ is globally negative definite over $\mathcal{K}$. This indicates that $\underline{p}^*$ is an asymptotically stable equilibrium point of ODE (17) over $\mathcal{K}$. The complete proof is given in Appendix D. ■

Corollary 3. The equilibrium point of ODE (17) is unique

over $\mathcal{K}$.

***Proof*:** Let $\underline{p}^*, \underline{q}^*$ be two equilibrium points of ODE (17). Theorem 3 proves that any equilibrium point of ODE (17), including $\underline{p}^*$, is asymptotically stable over $\mathcal{K}$. This means that all initial configurations within the state space $\mathcal{K}$ converge to $\underline{p}^*$. Using a similar approach for $\underline{q}^*$, one can conclude that all initial configurations within the state space $\mathcal{K}$ converge to $\underline{q}^*$. This implies that $\underline{p}^* = \underline{q}^*$, and thus, the equilibrium point of ODE (17) is unique over $\mathcal{K}$. ■

C. Convergence Results

In this section, we summarize the main results specified in the above lemmas and theorems in a main theorem (Theorem 4 given below).

Theorem 4. An ICLA with SL_{RP} learning automata, regardless of its initial configuration, converges in distribution to a random configuration, in which the mean value of the action probability of any action of any LA is inversely proportional to the average penalty received by that action.

***Proof*:** The evolution of an ICLA with SL_{RP} learning automata is described by equation (9). From this equation, it follows that $\{\underline{p}(k)\}_{k\geq 0}$ is a discrete-time Markov process. Lemma 3 states that this Markovian process is ergodic, and hence, it converges in distribution to a random configuration $\underline{p}^*$, irrespective of its initial configuration. Lemma 5 shows that such a configuration exists for ICLA, Corollary 3 states that it is unique, and Theorem 3 proves that it is asymptotically stable. Theorem 2 specifies the properties of the configuration $\underline{p}^*$. It shows that $\underline{p}^*$ satisfies the set of conditions given by (22). According to equation (22), in configuration $\underline{p}^*$, the action probability of any action of any LA is inversely proportional to the average penalty received by that action. ■

D. Expediency of ICLA

In this section, we introduce the concept of expediency for ICLA and specify the set of conditions under which an ICLA becomes expedient.

Definition 9. A pure-chance automaton is an automaton that chooses each of its actions with equal probability i.e., by pure chance, that is, an m-action automaton is pure-chance if $p_i = \frac{1}{m}, i = 1,2,\ldots,m$.

Definition 10. A pure-chance ICLA is an ICLA, for which every cell contains a pure-chance automaton rather than a learning automaton. The configuration of a pure-chance ICLA is denoted by $\underline{p}^{pc}$.

Definition 11. An ICLA is said to be expedient with respect to the cell c_i if $\lim_{k\to\infty} \underline{p}(k) = \underline{p}^*$ exists and the following inequality holds:

$$\lim_{k\to\infty} \mathrm{E}\left[D_i\left(\underline{p}(k)\right)\right] < \frac{1}{m_i}\sum_y d_{iy}\left(\underline{p}^*\right). \qquad (26)$$

In other words, an ICLA is expedient with respect to the cell c_i if, in the long run, the ith learning automaton performs better (receives less penalty) than a pure-chance automaton.

Definition 12. An ICLA is said to be expedient if it is expedient with respect to every cell in ICLA.

Theorem 5. An ICLA with SL_{RP} learning automata, regardless of the local rule being used, is expedient.

***Proof*:** To prove this theorem, we show that an ICLA with SL_{RP} learning automata is expedient with respect to every cell in ICLA. The proof is given in Appendix E. ■

V. Numerical Examples

In this section, we will give a number of numerical examples for illustrating the analytical results specified in previous sections. The first two sets of examples are used to illustrate the analytical results given in Theorem 4, and the next set of examples is used to study the behavior of ICLA in terms of expediency.

A. Numerical Example 1

This set of examples are given to illustrate that the action probability of any action of any learning automaton in an ICLA with SL_{RP} learning automata converges in distribution to a random variable, whose mean is inversely proportional to the average penalty received by that action. We consider three different ICLAs with different number of cells and different number of cell states. TABLE I gives the specifications used for this set of simulations. Here, $ICLA_{(n,m)}$ refers to an ICLA with n cells and m states for each cell. TABLE II compares the action probabilities of the actions of the learning automata in each ICLA at the end of the simulation time ($k > 3 \times 10^6$) with their theoretical values obtained from equation (22). As it can be seen from these tables, the action probabilities of all actions and their theoretical values approach each other. This is in coincidence with the results of the theoretical analysis given in Theorem 4, that is, an ICLA with SL_{RP} learning automata converges in distribution to a random configuration, in which the mean value of the action probability of any action of any LA is inversely proportional to the average penalty received by that action. Fig. 2 also shows the approach of these two values to each other over the simulation time for randomly selected actions of three randomly selected learning automata from $ICLA_{2,3}$ and $ICLA_{5,5}$.

TABLE I
SPECIFICATIONS USED FOR NUMERICAL EXAMPLE 1

ICLA	$ICLA_{2,3}$	$ICLA_{3,2}$	$ICLA_{5,5}$
Neighborhood	All cells are neighbors		
Learning Parameter	$a = 5\times10^{-5}$		
Initial Configuration	$p_{i1} = 1, p_{ir} = 0, \forall i,r, r \neq 1$		
Environment Response	$\mathcal{F}^i$ (α_1, …, α_n) is selected uniformly at random from the range [0, 1] at the beginning of the simulation		

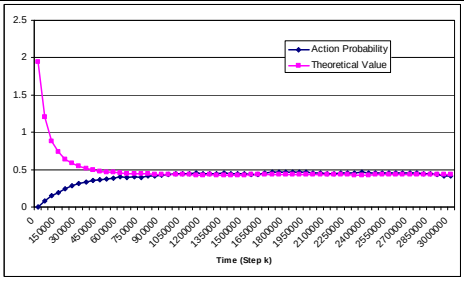
B. Numerical Example 2

The goal of conducting this set of numerical examples is to

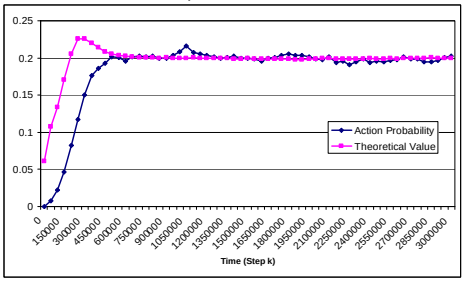
TABLE II

RESULTS OF THE NUMERICAL EXAMPLE 1 FOR $ICLA_{2,3}$, $ICLA_{3,2}$, $ICLA_{5,5}$ (P_{ij} IS THE ACTION PROBABILITY OF THE jTH ACTION OF THE iTH LEARNING AUTOMATON OBTAINED FROM THE SIMULATION STUDY AND p_{ij}^{*} IS THE CORRESPONDING THEORETICAL VALUE)

$ICLA_{2,3}$	LA_1	α_1	$p_{11}=.352$ $p_{11}^{*}=.361$	LA_2	α_1	$p_{21}=.289$ $p_{21}^{*}=.288$	$ICLA_{3,2}$	LA_1	α_1	$p_{11}=.746$ $p_{11}^{*}=.75$	LA_2	α_1	$p_{21}=.797$ $p_{21}^{*}=.8$	LA_3	α_1	$p_{31}=.663$ $p_{31}^{*}=.659$
		α_2	$p_{12}=.454$ $p_{12}^{*}=.441$		α_2	$p_{22}=.268$ $p_{22}^{*}=.268$			α_2	$p_{12}=.254$ $p_{12}^{*}=.25$		α_2	$p_{22}=.203$ $p_{22}^{*}=.2$		α_2	$p_{32}=.337$ $p_{32}^{*}=.341$
		α_3	$p_{13}=.194$ $p_{13}^{*}=.198$		α_3	$p_{23}=.443$ $p_{23}^{*}=.444$										
$ICLA_{5,5}$	LA_1	α_1	$p_{11}=.21$ $p_{11}^{*}=.211$	LA_2	α_1	$p_{21}=.208$ $p_{21}^{*}=.211$		LA_3	α_1	$p_{31}=.189$ $p_{31}^{*}=.192$	LA_4	α_1	$p_{41}=.176$ $p_{41}^{*}=.173$	LA_5	α_1	$p_{51}=.214$ $p_{51}^{*}=.212$
		α_2	$p_{12}=.212$ $p_{12}^{*}=.213$		α_2	$p_{22}=.215$ $p_{22}^{*}=.212$			α_2	$p_{32}=.222$ $p_{32}^{*}=.215$		α_2	$p_{42}=.208$ $p_{42}^{*}=.207$		α_2	$p_{52}=.183$ $p_{52}^{*}=.188$
		α_3	$p_{13}=.2$ $p_{13}^{*}=.199$		α_3	$p_{23}=.201$ $p_{23}^{*}=.2$			α_3	$p_{33}=.174$ $p_{33}^{*}=.177$		α_3	$p_{43}=.189$ $p_{43}^{*}=.192$		α_3	$p_{53}=.212$ $p_{53}^{*}=.214$
		α_4	$p_{14}=.213$ $p_{14}^{*}=.212$		α_4	$p_{24}=.178$ $p_{24}^{*}=.179$			α_4	$p_{34}=.182$ $p_{34}^{*}=.181$		α_4	$p_{44}=.197$ $p_{44}^{*}=.196$		α_4	$p_{54}=.174$ $p_{54}^{*}=.175$
		α_5	$p_{15}=.165$ $p_{15}^{*}=.165$		α_5	$p_{25}=.198$ $p_{25}^{*}=.198$			α_5	$p_{35}=.233$ $p_{35}^{*}=.235$		α_5	$p_{45}=.23$ $p_{45}^{*}=.232$		α_5	$p_{55}=.217$ $p_{55}^{*}=.211$



***ICLA*$_{2,3}$: *LA*$_1$: Action 2**



***ICLA*$_{5,5}$: *LA*$_4$: Action 4**

Fig. 2. Results of the numerical example 1 for randomly selected actions of 2 randomly selected learning automata from $ICLA_{2,3}$, $ICLA_{5,5}$

study the convergence behavior of ICLA when it starts to evolve within the environment from different initial configurations. For this study, we use an $ICLA_{5,5}$ with SL_{RP} learning automata. TABLE III gives the initial configurations of this ICLA. In this table, for each configuration, the initial action probability vectors for learning automata are given from left to right. For example, in configuration 3, the initial action probability vector of the learning automaton LA_4 is [.5, .1, .1, .2, .1]^T. Fig. 3 plots the evolution of the action probabilities of two randomly selected actions from two of the LA in ICLA for different initial configurations. As it can be seen from this figure, no matter what the initial configuration of ICLA is, it converges to its equilibrium configuration. Thus, the results of this set of examples coincide with the results given in Theorem 4 in Section 3, that is, the convergence of ICLA to its equilibrium configuration is independent of its initial configuration.

TABLE III

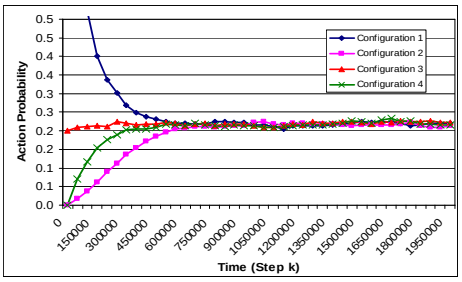
INITIAL CONFIGURATIONS FOR $ICLA_{5,5}$ USED IN NUMERICAL EXAMPLE 2

Configuration 1					**Configuration 2**				
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
1	1	1	1	1	0	0	0	0	1
Configuration 3					**Configuration 4**				
.2	.4	.3	.5	.6	.1	.8	.9	.2	.4
.2	0	.1	.1	.3	.2	.05	.05	.6	.1
.2	.4	.3	.1	0	.3	0	0	.1	.3
.2	0	.1	.2	0	.4	.05	0	.1	0
.2	.2	.2	.1	.1	0	.1	.05	0	.2

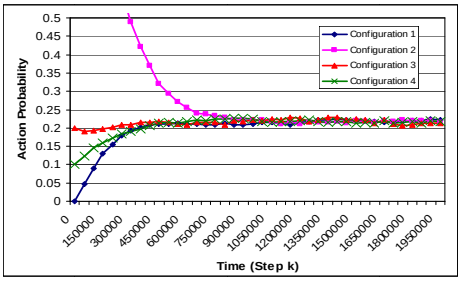
C. *Numerical Example 3*

This set of numerical examples is conducted to study the behavior of ICLA in terms of the expediency. For this study, we use the specifications of the first numerical example, given

in TABLE I. TABLE IV compares $\mathrm{E}\left[D_i\left(\underline{p}(k)\right)\right]$ and $\frac{1}{m_i}\sum_y d_{iy}\left(\underline{p}(k)\right)$ at the end of the simulation time (k> 2.85 × 10^6) for all learning automata of all ICLAs given in TABLE I. As it can be seen from this table, the average penalty received by any learning automaton is less than that of a pure chance automaton. This is in coincidence with the theoretical results given in Theorem 5, i.e., an ICLA with SL_{RP} learning automata, regardless of the local rule being used, is expedient.



***LA*$_1$: Action 5**



***LA*$_4$: Action 4**

Fig. 3. Results of the numerical example 2 for randomly selected actions of two randomly selected LA

TABLE IV
RESULTS OF THE NUMERICAL EXAMPLE 3

		$\mathrm{E}\left[D_i\left(\underline{p}(k)\right)\right]$	$\frac{1}{m_i}\sum_y d_{iy}\left(\underline{p}(k)\right)$
$ICLA_{2,3}$	LA_1	3.61×10^{-2}	5.05×10^{-2}
	LA_2	3.61×10^{-2}	4.48×10^{-2}
$ICLA_{3,2}$	LA_1	1.02×10^{-1}	1.04×10^{-1}
	LA_2	1.016×10^{-1}	1.018×10^{-1}
	LA_3	1.02×10^{-1}	1.03×10^{-1}
$ICLA_{5,5}$	LA_1	1.368×10^{-1}	1.372×10^{-1}
	LA_2	1.368×10^{-1}	1.371×10^{-1}
	LA_3	1.368×10^{-1}	1.374×10^{-1}
	LA_4	1.368×10^{-1}	1.373×10^{-1}
	LA_5	1.368×10^{-1}	1.374×10^{-1}

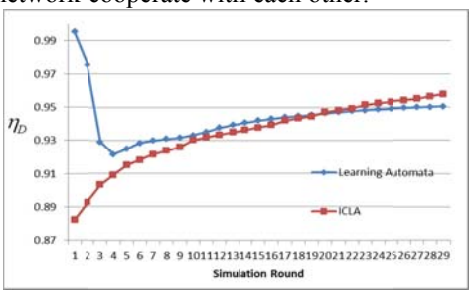
VI. CASE STUDY

To demonstrate the superiority of ICLA over learning automata, in this section we compare the results of applying these two models for solving the dynamic point coverage problem [20] in the area of wireless sensor networks. In this problem, an unknown number of targets are moving throughout the sensor field and the aim is to detect and track these moving targets using as few sensor nodes as possible.

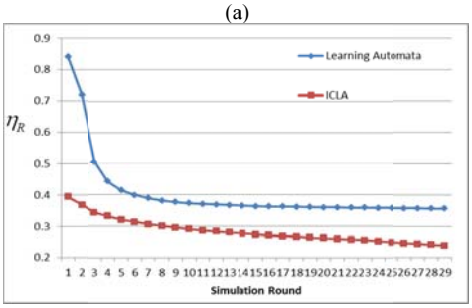
We have proposed two different approaches for solving this problem; 1) by using a number of non-cooperating learning automata, one at each sensor node of the network [20] and 2) by using an ICLA, in which there again exists one learning automaton at each sensor node, but these learning automata cooperate with each other [18]. Fig. 4. compares the results of these two approaches in terms of the following two criteria:

1. Network detection rate (η_D): This criterion determines the accuracy of detecting and tracking moving targets.
2. Network redundant active rate (η_R): This criterion determines the ratio of the time that a target is redundantly detected by more than one sensor node.

As it can be seen from this figure, using ICLA, results in higher η_D and lower η_R than using learning automata. In other words, the aim of the network, which is to detect moving targets (η_D) using as few sensor nodes as possible (η_R), is better fulfilled when learning automata residing in different nodes of the network cooperate with each other.



(a)



(b)

Fig. 4. Comparison of LA and ICLA for solving dynamic point coverage problem in terms of (a) network detection rate (η_D) and (b) network redundant active rate (η_R) criteria

VII. CONCLUSION

In this paper, we proposed irregular cellular learning automaton as an extension to cellular learning automaton (*CLA*) model. In contrast to CLA, the proposed model have irregular structure which is needed for modeling problems in some areas such as computer networks, web mining, and grid computing. The steady-state behavior of the proposed model was analytically studied and the results of this study were illustrated through some numerical examples. The concept of expediency was introduced for the proposed learning model. An ICLA is expedient with respect to each of its cells if, in the long run, the learning automaton resides in that cell performs better (receives less penalty) than a pure-chance automaton. An ICLA is expedient, if it is expedient with respect to all of

its constituting cells. Expediency is a notion of learning. Any model that is said to learn must then do at least better than its equivalent pure-chance model. The intended analytical studies showed that the proposed model, using SL_{RP} learning algorithm, is expedient. The proposed analytical results are valid only if the learning rate is sufficiently small.

APPENDIX A.

***Proof of Lemma 2*:** Let $ES = \left\{ \underline{\alpha} \middle| \underline{\alpha} = \left(\underline{\alpha}^T{}_1, \underline{\alpha}^T{}_1, \ldots, \underline{\alpha}^T{}_n \right)^T \right\}$ be the event set which causes the evolution of the state $\underline{p}(k)$. The evolution of the state $\underline{p}(k)$ is dependent on the occurrence of an associated event $ES(k)$. Thus

$$\underline{p}(k+1) = f_{ES(k)}\left(\underline{p}(k)\right). \quad (27)$$

where $f_{ES(k)}$ is defined according to equation (9). Let $\Pr\left[ES(k) = e \middle| \underline{p}(k) = \underline{p}\right] = \phi_e\left(\underline{p}\right)$ where $\phi_e\left(\underline{p}\right)$ is a real-valued function on $ES \times \mathcal{K}$. Now define $m\left(\phi_e\right)$ and $\mu\left(f_e\right)$ as:

$$m\left(\phi_e\right) = \sup_{\underline{p} \neq \underline{p}'} \frac{\left|\phi_e\left(\underline{p}\right) - \phi_e\left(\underline{p}'\right)\right|}{d\left(\underline{p}, \underline{p}'\right)}, \quad (28)$$

$$\mu\left(f_e\right) = \sup_{\underline{p} \neq \underline{p}'} \frac{d\left(f_e\left(\underline{p}\right), f_e\left(\underline{p}'\right)\right)}{d\left(\underline{p}, \underline{p}'\right)}, \quad (29)$$

and

$$\mu\left(f_e\right) = \sup_{\underline{p} \neq \underline{p}'} \frac{d\left(f_e\left(\underline{p}\right), f_e\left(\underline{p}'\right)\right)}{d\left(\underline{p}, \underline{p}'\right)}, \quad (30)$$

whether or not these are finite. Following propositions are held:

- ES is a finite set,
- $\left(\mathcal{K}, d\right)$ is a metric space and is compact,
- $m\left(\phi_e\right) < \infty$ for all $e \in ES$,
- $\mu\left(f_e\right) < 1$ for all $e \in ES$. To see this, consider $\underline{p}$ and $\underline{q}$ as two states of the process $\left\{\underline{p}(k)\right\}$. From equations (10), (11) we have:

$$\mu\left(f_e\right) = \frac{d\left(f_e\left(\underline{p}\right), f_e\left(\underline{q}\right)\right)}{d\left(\underline{p}, \underline{q}\right)} = \frac{\sum_i \left\| f_{e_i}\left(\underline{p}_i\right) - f_{e_i}\left(\underline{q}_i\right) \right\|}{\sum_i \left\| \underline{p}_i - \underline{q}_i \right\|} = \frac{\sum_i (1 - a_i) \cdot \left\| \underline{p}_i - \underline{q}_i \right\|}{\sum_i \left\| \underline{p}_i - \underline{q}_i \right\|}. \quad (31)$$

Since $0 < a_i < 1, \forall i$ it follows that $\mu\left(f_e\right) < 1$.

Therefore, and according to the definition of the distance diminishing processes given in [15] by Norman, Markovian process given by equation (9) is strictly distance diminishing. ■

***Proof of Corollary 1*:** From Lemma 2 It follows that:

$$d\left(\underline{p}^{(h)}, \underline{q}^{(h)}\right) = \sum_i (1 - a_i)^h \cdot \left\| \underline{p}_i - \underline{q}_i \right\|. \quad (32)$$

Right hand side of equation (32) tends to zero as $h \to \infty$. Hence $\underline{p}^{(h)} \to \underline{q}^{(h)}$ irrespective of the initial configurations $\underline{p}$ and $\underline{q}$. ■

APPENDIX B.

***Proof of Lemma 5*:** Let $\zeta\left(\underline{p}\right) = \underline{a} f\left(\underline{p}\right) + \underline{p}$ where matrix $\underline{a}$ is equivalent to the one given in equation (9). Components of $\zeta\left(\underline{p}\right)$ can be obtained as follows:

$$\zeta_{ir}\left(\underline{p}\right) = \frac{a_i}{m_i - 1} \cdot \left[D_i\left(\underline{p}\right) - m_i p_{ir} d_{ir}\left(\underline{p}\right) \right] + p_{ir}. \quad (33)$$

It is easy to verify that $0 \le \zeta_{ir}\left(\underline{p}\right) < 1$ for all i and r. Thus, $\zeta\left(\underline{p}\right)$ is a continuous mapping from $\mathcal{K}$ to $\mathcal{K}$. Since $\mathcal{K}$ is closed, bounded, and convex (Lemma 1), we can use the Brouwer's fixed point theorem to show that $\zeta\left(\underline{p}\right)$ has at least one fixed point. Let $\underline{p}^*$ be a fixed point of $\zeta\left(\underline{p}\right)$, thus we have

$$\zeta\left(\underline{p}^*\right) = \underline{p}^*, \quad (34)$$

or equivalently

$$\underline{a} f\left(\underline{p}^*\right) + \underline{p}^* = \underline{p}^*. \quad (35)$$

Since $\underline{a}$ is a diagonal matrix with no zero elements on its main diagonal, it can be concluded from (35) that

$$f\left(\underline{p}^*\right) = \underline{0}. \quad (36)$$

Since every point $\underline{p}^*$, that satisfies $f\left(\underline{p}^*\right) = \underline{0}$, is an equilibrium point of ODE (17), we can conclude that ODE (17) has at least one equilibrium point. ■

APPENDIX C.

***Proof of Lemma 6*:** Let

$$\delta^*{}_{ir} = \frac{D_i\left(\underline{p}^*\right)}{m_i d_{ir}\left(\underline{p}^*\right)}. \quad (37)$$

Using equations (24) and (37), components of the derivative of $\hat{\underline{p}}$ with respect to time can be given as

$$\frac{d\hat{p}_{ir}}{dt} = \frac{dp_{ir}}{dt} + \frac{d\,\delta^*{}_{ir}}{dt} = \frac{dp_{ir}}{dt} \quad for\ all\ i, r. \quad (38)$$

Using equations (18) and (37), (38) can be rewritten as

$$\frac{d\hat{p}_{ir}}{dt} = \frac{1}{m_i - 1} \cdot \left[D_i\left(\underline{p}\right) - m_i \cdot \left(\hat{p}_{ir} + \delta^*{}_{ir}\right) \cdot d_{ir}\left(\underline{p}\right) \right] \quad (39)$$

Equation (39) is valid for all configurations $\underline{\hat{p}}$ including $\underline{\hat{q}}$ in which $\hat{q}_{ir} = 0$ for a particular action r of the ith learning automaton. For this configuration, it is easy to verify that $\frac{d\underline{\hat{q}}_{ir}}{dt} = 0$.

Next, use equation (7) to rewrite equation (39) as follows:

$$\frac{d\hat{p}_{ir}}{dt} = \frac{1}{m_i - 1} \cdot \left[\begin{array}{l} d_{ir}\left(\underline{p}\right) \cdot \left(1 - m_i\right) \cdot \left(\hat{p}_{ir} + \delta^*_{ir}\right) + \\ \sum_{y \neq r} \left(\hat{p}_{iy} + \delta^*_{iy}\right) d_{iy}\left(\underline{p}\right) \end{array} \right] \quad \text{for all } i, r \tag{40}$$

Equation (40) is also valid for all configurations $\underline{\hat{p}}$ including $\underline{\hat{q}}$. Thus, we have

$$\frac{d\hat{q}_{ir}}{dt} = \frac{1}{m_i - 1} \cdot \left[\begin{array}{l} d_{ir}\left(\underline{q}\right) \cdot \left(1 - m_i\right) \cdot \delta^*_{ir} + \\ \sum_{y \neq r} \left(\hat{q}_{iy} + \delta^*_{iy}\right) d_{iy}\left(\underline{q}\right) \end{array} \right] = 0, \tag{41}$$

from which it immediately follows that

$$d_{ir}\left(\underline{q}\right) \cdot \left(1 - m_i\right) \cdot \delta^*_{ir} + \sum_{y \neq r} \left(\hat{q}_{iy} + \delta^*_{iy}\right) d_{iy}\left(\underline{q}\right) = 0. \tag{42}$$

Since none of the terms in equation (42) depend on the value of $\hat{q}_{ir}$, this equation is valid for all configurations $\underline{\hat{p}}$, that is,

$$d_{ir}\left(\underline{p}\right) \cdot \left(1 - m_i\right) \cdot \delta^*_{ir} + \sum_{y \neq r} \left(\hat{p}_{iy} + \delta^*_{iy}\right) d_{iy}\left(\underline{p}\right) = 0, \quad \text{for all } \hat{p}, i, r \tag{43}$$

Using equation (43) in equation (40) we get

$$\frac{d\hat{p}_{ir}}{dt} = -d_{ir}\left(\underline{p}\right) \hat{p}_{ir} \quad \text{for all } i, r, \tag{44}$$

and hence the lemma. ■

APPENDIX D.

***Proof of Theorem 3*:** Apply transformation (24) to transfer the origin to $\underline{p}^*$. Now consider the following positive definite Lyapunov function:

$$V\left(\underline{\hat{p}}\right) = -\sum_i \sum_y \hat{p}_{iy} \cdot \ln\left(1 - \hat{p}_{iy}\right). \tag{45}$$

$V\left(\underline{\hat{p}}\right) \geq 0$ for all configurations $\underline{\hat{p}}$ and is zero only when $\hat{p}_{ir} = 0$ for all i and r. Time derivative of $V(.)$ can be expressed as

$$\dot{V}\left(\underline{\hat{p}}\right) = -\sum_i \sum_y \frac{d\hat{p}_{iy}}{dt} \cdot \upsilon\left(\hat{p}_{iy}\right), \tag{46}$$

where

$$\upsilon\left(\hat{p}_{iy}\right) = \left[\frac{\ln\left(1 - \hat{p}_{iy}\right) \cdot \left(1 - \hat{p}_{iy}\right) - \hat{p}_{iy}}{1 - \hat{p}_{iy}} \right]. \tag{47}$$

Considering the value of $\underline{\hat{p}}_{iy}$, following three cases may arise for each term of this derivative:

- $0 < \hat{p}_{iy} \leq 1$: In this case, $\upsilon\left(\hat{p}_{iy}\right) < 0$ and $\frac{d\hat{p}_{iy}}{dt} < 0$ (Corollary 2). Therefore, $\frac{d\hat{p}_{iy}}{dt} \cdot \upsilon\left(\hat{p}_{iy}\right) > 0$.
- $-1 \leq \hat{p}_{iy} < 0$: In this case $\upsilon\left(\hat{p}_{iy}\right) > 0$ and $\frac{d\hat{p}_{iy}}{dt} > 0$ (Corollary 2). Therefore, $\frac{d\hat{p}_{iy}}{dt} \cdot \upsilon\left(\hat{p}_{iy}\right) > 0$.
- $\hat{p}_{iy} = 0$: In this case $\frac{d\hat{p}_{iy}}{dt} \cdot \upsilon\left(\hat{p}_{iy}\right) = 0$.

Thus, $\dot{V}\left(\underline{\hat{p}}\right) \leq 0$, for all configurations $\underline{\hat{p}}$ and is zero only when $\hat{p}_{ir} = 0$ for all i and r. Therefore, using the Lyapunov theorems for autonomous systems, it can be proved that $\underline{p}^*$ is an asymptotically stable equilibrium point of ODE (17) over $\mathcal{K}$. ■

APPENDIX E.

***Proof of Theorem 5*:** We have to show that an ICLA with SL_{RP} learning automata is expedient with respect to all of its cells; that is, $\lim_{k \to \infty} \underline{p}\left(k\right) = \underline{p}^*$ exists and the following inequality holds:

$$\lim_{k \to \infty} \mathrm{E}\left[D_i\left(\underline{p}\left(k\right)\right)\right] < \frac{1}{m_i} \sum_y d_{iy}\left(\underline{p}^*\right), \quad \text{for every } i \tag{48}$$

Using equation (7), the left hand side of this inequality can be rewritten as:

$$\begin{aligned} \lim_{k \to \infty} \mathrm{E}\left[D_i\left(\underline{p}\left(k\right)\right)\right] &= \lim_{k \to \infty} \mathrm{E}\left[\sum_y d_{iy}\left(\underline{p}\left(k\right)\right) p_{iy}\left(k\right)\right] \\ &= \sum_y \lim_{k \to \infty} \mathrm{E}\left[d_{iy}\left(\underline{p}\left(k\right)\right) p_{iy}\left(k\right)\right]. \end{aligned} \tag{49}$$

Since $d_{iy}\left(\underline{p}\left(k\right)\right)$ and $p_{iy}\left(k\right)$ are independent, (49) can be simplified to:

$$\lim_{k \to \infty} \mathrm{E}\left[D_i\left(\underline{p}\left(k\right)\right)\right] = \sum_y \left(\lim_{k \to \infty} \mathrm{E}\left[d_{iy}\left(\underline{p}\left(k\right)\right)\right] \cdot \lim_{k \to \infty} \mathrm{E}\left[p_{iy}\left(k\right)\right] \right). \tag{50}$$

From Theorem 4, we have $\lim_{k \to \infty} \mathrm{E}\left[\underline{p}\left(k\right)\right] = \underline{p}^*$ and hence $\lim_{k \to \infty} \mathrm{E}\left[p_{ir}\left(k\right)\right] = p^*_{ir}$ for all i and r. Using equation (6), $\lim_{k \to \infty} \mathrm{E}\left[d_{ir}\left(\underline{p}\left(k\right)\right)\right]$ can be computed as follows:

$$\lim_{k \to \infty} \mathrm{E}\left[d_{ir}\left(\underline{p}\left(k\right)\right)\right] = \lim_{k \to \infty} \tag{51}$$

$$\mathrm{E}\left[\sum_{y_{j_1},\ldots,y_{j_{N_i}}} \mathcal{F}^i\left(y_{j_1}, y_{j_2},\ldots, y_{j_{N_i}}, r\right) \prod_{LA_l \in N(i)} p_{ly_{j_l}}(k)\right]$$

$$= \sum_{y_{j_1},\ldots,y_{j_{N_i}}} \mathcal{F}^i\left(y_{j_1}, y_{j_2},\ldots, y_{j_{N_i}}, r\right) \prod_{LA_l \in N(i)} \lim_{k\to\infty} \mathrm{E}\left[p_{ly_{j_l}}(k)\right]$$

$$= d_{ir}\left(\underline{p}^*(k)\right).$$

Thus, we get:

$$\lim_{k\to\infty} \mathrm{E}\left[D_i\left(\underline{p}(k)\right)\right] = \sum_y \left(d_{iy}\left(\underline{p}^*\right) p^*_{iy}\right). \quad (52)$$

Now, we have to show that:

$$\sum_y d_{iy}\left(\underline{p}^*\right) p^*_{iy} < \frac{1}{m_i} \sum_y d_{iy}\left(\underline{p}^*\right) \quad for\ every\ i. \quad (53)$$

Each side of this inequality is a convex combination of $d_{iy}\left(\underline{p}^*\right), y = 1,\ldots,m_i$. In the convex combination given on the right hand side of (53), weights of all $d_{ir}\left(\underline{p}^*\right)$ are equal to $\frac{1}{m_i}$, whereas in the convex combination given on the left hand side, weight of each $d_{ir}\left(\underline{p}^*\right)$ is inversely proportional to its value, that is, the larger $d_{ir}\left(\underline{p}^*\right)$, the smaller its weight is (considering equation (22)). Therefore, the convex combination given on the left hand side of inequality (53) is smaller than the one given on the right hand side, and hence the theorem. ■

REFERENCES

[1] J. Kari, "Reversibility of 2D Cellular Automata is Undecidable," *Physica D*, Vol. 45, No. 1–3, 1990, pp. 379–385.

[2] N. H. Pakard, S. Wolfram, "Two-dimensional Cellular Automata," *Journal of State Physics*, Vol. 38, No. 5/6, pp. 901-946, 1985.

[3] E. Fredkin, "Digital Machine: An Informational Process Based on Reversible Cellular Automata," Physica D, Vol. 45, No. 1–3, 1990, pp. 254–270.

[4] M. A. L. Thathachar, P. S. Sastry, "Networks of Learning Automata," *Kluwer Academic Publishers*, USA, 2004.

[5] M. R.Meybodi, H. Beigy, and M. Taherkhani, "Cellular Learning Automata and its Applications," *Sharif Journal of Science and Technol*ogy, Vol. 19, No. 25, 2003, pp. 54–77.

[6] H. Beigy, M. R. Meybodi, "A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach," in *Springer-Verlag Lecture Notes in Computer Science*, Vol. 2690. New York: Springer-Verlag, 2003, pp. 119–126.

[7] H. Beigy, M. R. Meybodi, "Asynchronous Cellular Learning Automata," *Automatica*, Vol. 44, No. 5, pp. 1350–1357, 2008.

[8] M. R. Meybodi, F. Mehdipour, "Application of Cellular Learning Automata with Input to VLSI Placement," *Journal of Modarres,* University of Tarbiat Modarres, Vol. 16, 2004, pp. 81-95.

[9] H. Beigy, M. R. Meybodi, "Open Synchronous Cellular Learning Automata," *Advances in Complex Systems*, Vol. 10, No. 4, pp. 527–556, 2007.

[10] H. Beigy, M. R. Meybodi, "Cellular Learning Automata with Multiple Learning Automata in Each Cell and Its Applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* Vol. 40, No. 1, 2010, pp. 54-66.

[11] H. Beigy, M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata," *Journal of Advances in Complex Sys*tems, Vol. 7, No. 3, pp. 295–320, 2004.

[12] M. L. Tsetlin, "On the Behavior of Finite Automata in Random Media," *Automation and Remote Control*, Vol. 22, pp. 1210-1219, 1962.

[13] K. S. Narendra, M. A. L. Thathachar, "Learning Automata: An Introduction," *Prentice-Hall Inc*, New Jercy, 1989.

[14] D. L. Isaacson, R. W. Madsen, "Markov Chains: Theory and Applications," *John Wiley & Sons*, New York, 1976.

[15] M. F. Norman, "Some Convergence Theorems for Stochastic Learning Models with Distance Diminishing Operators," *Journal of Mathematical Psychology*, vol. 5, 1968, pp. 61-101.

[16] M. Esnaashari and M. R. Meybodi, "Irregular Cellular Learning Automata and Its Application to Clustering in Sensor Networks," *in Proc. of 15th Conf. on Electrical Engineering,* Volume on Communication, Telecommunication Research Center, Tehran, Iran, 2007.

[17] M. Esnaashari and M. R. Meybodi, "A Cellular Learning Automata based Clustering Algorithm for Wireless Sensor Networks," *Sensor Letters,* Vol. 6, No. 5, 2008, pp. 723-735.

[18] M. Esnaashari and M. R. Meybodi, "Dynamic Point Coverage Problem in Wireless Sensor Networks: A Cellular Learning Automata Approach," *Journal of Ad hoc and Sensors Wireless Networks,* Vol. 10, No. 2-3, 2010, pp. 193-234.

[19] M. Ahmadinia, M. R. Meybodi, M. Esnaashari, and H. Alinejad Rokney, "Energy Efficient and Multi Clustering Algorithm in Wireless Networks Using Cellular Learning Automata," *IETE Journal of Research*, Vol. 59, No. 6, 2013, pp. 774-782.

[20] M. Esnaashari and M. R. Meybodi, "A Learning Automata based Scheduling Solution to the Dynamic Point Coverage Problem in Wireless Sensor Networks," *Computer Networks,* Vol. 54, 2010, pp. 2410-2438.

[21] M. L. Tsetlin, "Automaton Theory and Modeling of Biological Systems," *Academia Press*, New York, 1973.

[22] A. G. Barto and P. Anandan, "Pattern-Recognizing Stochastic Learning Automata," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 15, No. 3, 1985, pp. 360-375.

[23] K. S. Narendra and K. Parthasarathy, "Learning Automata Approach to Hierarchical Multi-objective Analysis," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 21, No. 1, 1991, pp. 263-272.

[24] P. S. Sastry, V. V. Phansalkar, and M. Thathachar, "Decentralized Learning of Nash Equilibria in Multi-Person Stochastic Games with Incomplete Information," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 24, No. 5, 1994, pp. 769-777.

[25] O-C. Granmo, B. J. Oommen, S. A. Myrer, and M. G. Olsen, "Learning Automata-Based Solutions to the Nonlinear Fractional Knapsack Problem With Applications to Optimal Resource Allocation," *IEEE Transactions on Systems, Man and Cybernetics*, *Part B: Cybernetics,* Vol. 37, No. 2B, 2007, pp. 166-175.

[26] B. J. Oommen and K. Hashem, "Modeling the 'Learning Process' of the Teacher in a Tutorial-like System Using Learning Automata," *IEEE Transactions on Systems, Man and Cybernetics*, *Part B: Cybernetics,* Vol. 43, 2013, pp. 2020-2031.

[27] A. Yazidi, O-C Granmo, and B. J. Oommen, "Learning Automaton Based On-line Discovery and Tracking of Spatio-Temporal Event Patterns," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics,* Vol. 43, 2013, pp. 1118-1130.

Mehdi Esnaashari received the B.S., M.S., and Ph.D. degrees in Computer Engineering all from Amirkabir University of Technology in Iran, in 2002, 2005, and 2011 respectively. Currently, he is an assistant professor at Iran Telecommunications Research Center, Tehran, Iran. His research interests include computer networks, learning systems, soft computing, and information retrieval.

M.R. Meybodi received the B.S. and M.S. degrees in Economics from the Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M.S. and Ph.D. degrees from the 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 the Western Michigan University, and from1985 to 1991 as an Associate Professor at the Ohio University, USA. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.