# Asynchronous Cellular Learning Automata

Hamid Beigy [a], M. R. Meybodi [b],

[a] *Computer Engineering Department, Sharif University of Technology, Tehran, Iran*
*Institute for Studies in Theoretical Physics and Mathematics (IPM), School of Computer Science, Tehran, Iran*

[b] *Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran*
*Institute for Studies in Theoretical Physics and Mathematics (IPM), School of Computer Science, Tehran, Iran*

**Abstract**

Cellular learning automata is a combination of cellular automata and learning automata. The synchronous version of cellular learning automata in which all learning automata in different cells are activated synchronously, has found many applications. In some applications a type of cellular learning automata in which learning automata in different cells are activated asynchronously (asynchronous cellular learning automata) is needed. In this paper, we introduce asynchronous cellular learning automata and study its steady state behavior. Then an application of this new model to cellular networks has been presented.

*Key words:* Cellular Automata, Learning Automata, Cellular Learning Automata, Reinforcement Learning, Cellular Networks

## 1 Introduction

Cellular automata (CA) consist of a set of simple identical components, called *cells*, each of which occupies a node of a regular, discrete, infinite spatial network. Each cell can assume a state from a finite set, CA evolves in discrete steps, changing the states of all its cells according to a local rule that depends on the environment of the cell. The environment of a cell is usually taken to be a small number of neighboring cells, which can include the cell itself. The evolution (dynamics) of a CA is generated by repeatedly applying the local rule to its cells. The local rule can be applied in a number of different ways that referred to as *updating methods*. In *synchronous updating*, state of all cells are changed at the same time and in *asynchronous updating*, the state of cells are changed one after another [1]. Asynchronous updating requires the specification of an order in which cells are updated.

In other hand, a learning automaton (LA) has finite set of actions and at each stage chooses one of them. The choice of an action depends on the state of LA represented by an action probability vector. For each action chosen by the LA, the environment gives a reinforcement signal with unknown probability distribution. Then upon receiving the reinforcement signal, the LA updates its action probability vector by employing a learning algorithm. LAs can be classified into two main families: *fixed structure learning automata* and *variable structure learning automata* [2]. Variable structure learning automata are represented by a triple $\langle \underline{\beta}, \underline{\alpha}, T \rangle$, where $\underline{\beta} = \{0, 1\}$ is a set of inputs, $\underline{\alpha} = \{\alpha_1, \ldots, \alpha_r\}$ is a set of actions, and $T$ is the learning algorithm. The learning algorithm is a recurrence relation and is used to modify action probability vector ($\underline{p}$) of the automaton. Various learning algorithms have been reported in the literature. Below, a learning algorithm, called *linear reward-inaction* ($L_{R-I}$) is given. Let $\alpha_i$ be the action chosen at stage $k$ as a sample realization from probability distribution $\underline{p}(k)$. In $L_{R-I}$ algorithm, the action probability vector is updated using

$$p_j(k+1) = \begin{cases} p_j(k) + b \times [1 - p_j(k)] & \text{if} \quad i = j \\ p_j(k) - b \times p_j(k) & \text{if} \quad i \neq j \end{cases} \tag{1}$$

when the environment rewards the chosen action of LA and the action probability vector remains unchanged when the environment penalizes the chosen action of LA. Parameters $b \in (0, 1)$ and $r$ represent *learning parameter* and the number of actions for LA, respectively [2]. LA have been used successfully in many applications such as solving NP-Complete problems [3], call admission in cellular mobile networks [4], capacity assignment [5] and neural network engineering [6,7] to mention a few.

Learning automata are, by design, "simple agents for

*Email addresses:* `beigy@ce.sharif.edu` (Hamid Beigy), `mmeybodi@aut.ac.ir` (M. R. Meybodi).

doing simple things". The full potential of a LA is realized when multiple LA interact with each other. Interaction may assume different forms such as tree, mesh, array and etc. Depending on the problem that needs to be solved, one of these structures for interaction may be chosen. In most applications, full interaction between all LAs is not necessary and is not natural. Local interaction of LAs, which can be defined in a from of graph such as tree, mesh, or array, is natural in many applications. In the other hand, CA are mathematical models for systems consisting of large numbers of simple identical components with local interactions. In [8], CA and LA is combined to obtain a new model called cellular learning automata (CLA). This model is superior to CA because of its ability to learn and also is superior to single LA because it is a collection of LAs which can interact with each other. The basic idea of CLA, which is a subclass of stochastic CA, is to use LAs to adjust the state transition probability of stochastic CA.

CLA is a mathematical model for dynamical complex systems that consists of large number of simple components [8]. The simple components, which have learning capability, act together to produce complicated behavioral patterns. A CLA is a CA in which a learning automaton will be assigned to its every cell. The learning automaton residing in each cell determines the state of the cell on the basis of its action probability vector. Like CA, there is a rule that CLA operate under it. The rule of CLA and the actions selected by the neighboring LAs of any cell determine the reinforcement signal to the LA residing in that cell. In CLA, the neighboring LAs of any cell constitute its local environment. This environment is nonstationary because of the fact that it changes as action probability vectors of neighboring LAs change.

The operation of CLA could be described as follows: at the first step, the internal state of every cell is specified. The state of every cell is determined on the basis of action probability vector of the LA residing in that cell. The initial value of this state may be chosen on the basis of past experience or at random. In the second step, the rule of CLA determines the reinforcement signal to each LA residing in that cell. Finally, each LA updates its action probability vector on the basis of supplied reinforcement signal and the chosen action. This process continues until the desired result is obtained. A number of applications for synchronous CLA have been developed recently such as image processing [9], and channel assignment in cellular networks [10]. In [11], a mathematical framework to study the behavior of the synchronous CLA is given and its steady state properties has been investigated. It is shown that the synchronous CLA converges to a globally stable state for a class of rules called commutative rules.

In some applications such as call admission control and dynamic channel assignment in cellular networks, a type of CLA in which the LA in different cells are activated asynchronously is needed. We call such a CLA as *asynchronous CLA* (ACLA). In this paper, we introduce ACLA and study its steady state behavior. It is shown that for commutative rules, the ACLA converges to a globally stable state. Then an application of this new model to cellular mobile networks has been presented.

The rest of this paper is organized as follows. In sections 2 and 3, we present the ACLA and its steady state behavior, respectively. In section 4, the behavior of the ACLA when the commutative rules are used is studied. Sections 5 presents the numerical example and section 6 concludes the paper.

## 2 Asynchronous Cellular Learning Automata

A CLA is called asynchronous if at a given time only some LAs are activated independently from each other, rather than all together in parallel. The LAs may be activated in either *time-driven* or *step-driven* manner. In time-driven ACLA, each cell is assumed to have an internal clock which wakes up the LA associated to that cell while in step-driven ACLA, a cell is selected in fixed or random sequence. The ACLA in which cells are selected randomly is of more interest to us because of its applications. Formally a $d-$dimensional step–driven ACLA is given below.

**Definition 1** *A $d-$dimensional step-driven ACLA with $n$ cells is a structure $\langle Z^d, \Phi, A, N, F, \underline{\rho}\rangle$, where*

*(1) $Z^d$ is a lattice of $d-$tuples of integer numbers.*
*(2) $\underline{\Phi}$ is a finite set of states.*
*(3) $A$ is the set of LAs each of which is assigned to each cell of the CA.*
*(4) $N = \{\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_{\bar{m}}\}$ is neighborhood vector.*
*(5) $F : \underline{\Phi}^{\bar{m}} \rightarrow \underline{\beta}$ is the local rule of the ACLA.*
*(6) $\underline{\rho}$ is an $n$-dimensional vector called activation probability vector, where $\rho_i$ is the probability that the LA in cell $i$ (for $i = 1, \ldots, n$) to be activated in each stage.*

Suppose LA $A_i$ with finite action set $\underline{\alpha}_i$ is associated to cell $i$ (for $i = 1, \ldots, n$) of ACLA. Let cardinality of $\underline{\alpha}_i$ be $m_i$ and the state of ACLA represented by $\underline{p} = (\underline{p}_1^T, \underline{p}_2^T, \ldots, \underline{p}_n^T)^T$, where $\underline{p}_i = (p_{i1}, \ldots, p_{im_i})^T$ is the action probability vector of LA $A_i$.

The operation of ACLA takes place as the following iterations. At stage $k$, each LA $A_i$ is activated with probability $\rho_i$ and the activated LAs choose one of their actions. Then rule calculates the reinforcement signal. The actions of neighboring cells of an activated cell are their most recently selected actions. Let $\alpha_i \in \underline{\alpha}_i$ and $\beta_i \in \underline{\beta}$ be the action chosen by the activated LA $A_i$ and the reinforcement signal received by LA $A_i$, respectively. The reinforcement signal is produced by the application of local rule $F^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2} \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) \to \underline{\beta}$, where $F^i$ is the local rule for cell $i$. The higher value of $\underline{\beta}_i$ means that the chosen action of $A_i$ is more rewarded. Finally, activated LAs update their action probability vectors and the process repeats.

Since each set $\underline{\alpha}_i$ is finite, the local rule can be represented by a hyper matrix of dimensions $m_1 \times m_2 \times \cdots \times m_{\bar{m}}$. These $n$ hyper matrices together constitutes what we call the local rule of ACLA. When all hyper matrices are equal, the rule is uniform; otherwise the rule is nonuniform. For the sake of simplicity in our presentation, the rule $F^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2} \ldots, \alpha_{i+\bar{x}_{\bar{m}}})$ is denoted by $F^i(\alpha_1, \alpha_2 \ldots, \alpha_{\bar{m}})$.

### *2.1 Definitions and Notations*

In what follows, we give some definitions and then derive some preliminary results regarding ACLA which will be used later in this paper to study the behavior of ACLA.

**Definition 2** *Configuration of ACLA is a mapping $K: Z^d \to \underline{p}$ that associates an action probability vector with every cell. We will denote the set of all configurations of ACLA by $K$.*

The application of the local rule to every cell allows transforming a configuration to a new one. ACLA may be described as a network of LA assigned to the nodes of a graph (usually a finite and regular lattice). Each node is connected to a set of nodes (its neighbors), and each LA updates its action probability vector at discrete time instants, using a learning algorithm and a rule which depends on its own action and that of its neighbors.

**Definition 3** *Let $\underline{p}$ be a configuration where $\underline{p} = (\underline{p}_1^T, \underline{p}_2^T, \ldots, \underline{p}_n^T)^T$ and $\underline{p}_i = (p_{i1}, \ldots, p_{im_i})^T$. Configuration $\underline{p}$ is called deterministic if action probability vector of each LA is a unit vector; otherwise it is called probabilistic. Hence, the set of all deterministic configurations, $K^*$, and the set of all probabilistic configurations, $K$, in ACLA are $K^* = \{\underline{p} | p_{iy} \in \{0,1\} \quad \forall y, i\}$, and $K = \{\underline{p} | p_{iy} \in [0,1] \quad \forall y, i\}$, respectively, where $\sum_y p_{iy} = 1$ for all $i$. Every configuration $\underline{p} \in K^*$ is called a corner of $K$. Note that, $K$ is the convex hull of $K^*$ [11].*

**Definition 4** *For configuration $\underline{p}$, the average reward for action $r$ of LA $A_i$ is defined as*

$$d_{ir}(\underline{p}) = \sum_{y_2} \cdots \sum_{y_{\bar{m}}} F^i(r, y_2, \ldots, y_{\bar{m}}) \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{l y_l}, \tag{2}$$

*and the average reward for LA $A_i$ is equal to*

$$D_i(\underline{p}) = \sum_r d_{ir}(\underline{p}) p_{ir}. \tag{3}$$

Let $D(\underline{p}) = \sum_i D_i(\underline{p})$ be the average reward for ACLA. The above definition implies that if LA $A_j$ is not a neighboring LAs for $A_i$, then $d_{ir}(\underline{p})$ does not depend on $\underline{p}_j$.

**Definition 5** *Configuration $\underline{p}$ is called compatible if*

$$\sum_r d_{ir}(\underline{p}) p_{ir} \geq \sum_r d_{ir}(\underline{p}) q_{ir} \tag{4}$$

*for all $\underline{q} \in K$ and all $i$. $\underline{p} \in K$ is said to be* fully compatible*, if the above inequalities are strict.*

The compatibility of a configuration implies that no LA in ACLA have any reason to change its action.

**Definition 6** *Configuration $\underline{p}$ is called admissible if for all $\underline{q} \in K$ and all $i$ we have $D_i(\underline{p}) \geq D_i(\underline{q})$.*

The compatibility is a local concept and can be calculated by looking only into the neighboring LAs, but the admissibility is a global concept.

**Corollary 7** *Admissibility implies compatibility but the converse is not true, i.e, every admissible configuration is compatible but every compatible configuration is not necessarily admissible.*

**Proof:** The proof is trivial from Definitions 5 and 6. ■

**Corollary 8** *Configuration $\underline{p}$ is admissible if and only if for all $\underline{q} \in K$, we have $D(\underline{p}) \geq D(\underline{q})$.*

**Proof:** The proof is trivial from Definitions 6 and 4. ■

**Lemma 9** *ACLA has at least one compatible configuration.*

**Proof:** The proof for this Lemma is similar to the proof of Lemma 2 in [11]. ■

**Lemma 10** *Configuration $\underline{p}$ is compatible if and only if for all $i$ and $r$, we have $d_{ir}(\underline{p}) \leq D_i(\underline{p})$,*

**Proof:** The proof for this Lemma is given in [11]. ■

**Lemma 11** *If configuration $\underline{p}$ is compatible, then for each $i$ and all $r$ and $p_{ir} > 0$, we have $d_{ir}(\underline{p}) = D_i(\underline{p})$.*

**Proof:** The proof for this Lemma is given in [11]. ■

This Lemma provides a means of finding compatible configurations in the ACLA.

**Theorem 12** *Configuration $\underline{p}$ is compatible if and only if for all $\underline{q} \in K$, $\sum_i \sum_y d_{iy}(\underline{p}) [p_{iy} - q_{iy}] \geq 0$ holds.*

**Proof:** The proof for this Theorem is given in [11]. ■

This Theorem states that, when the action probability vector of all learning automata except the specific $A_i$ are held fixed, then the configuration reached by the ACLA at the point, where the average reward of $A_i$ is maximum, is compatible.

**Theorem 13** *A corner* $\underline{p} = \left(\underline{e}_{t^1}^T, \ldots, \underline{e}_{t^n}^T\right)^T$ *is compatible if and only if for all* $i = 1, \ldots, n$ *and* $r^i \neq t^i$ *we have* $F^i(t^i, t_2, \ldots, t_{\bar{m}}) \geq F^i(r^i, t_2, \ldots, t_{\bar{m}})$ .

**Proof:** We first show that if $\underline{p}$ is compatible then $F^i(t^i, t_2, \ldots, t_{\bar{m}}) \geq F^i(r^i, t_2, \ldots, t_{\bar{m}})$. In order to show this assertion, we assume that $\underline{q} = \left(\underline{e}_{t^1}^T, \ldots, \underline{e}_{r^i}^T, \ldots, \underline{e}_{t^n}^T\right)^T$ for $r^i \neq t^i$ is not a compatible corner. From Definition 5, we have $\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir}$. Since $\underline{p}$ and $\underline{q}$ are two corners, then the above inequality can be simplified as $d_{it^i}(\underline{p}) \geq d_{ir^i}(\underline{p})$. Substituting $d_{ir}(\underline{p})$ from equation (3), we obtain $F^i(t^i, \ldots, t_{\bar{m}}) \geq F^i(r^i, \ldots, t_{\bar{m}})$, which concluding this assertion.

Conversely, assume $F^i(t^i, t_2, \ldots, t_{\bar{m}}) \geq F^i(r^i, t_2, \ldots, t_{\bar{m}})$ but $\underline{p}$ is not compatible. Multiplying both sides of the above inequality by $\prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l}$ and summing over all actions of neighboring automata of cell $i$, we obtain $\sum_{t_2} \cdots \sum_{t_{\bar{m}}} F^i(t^i, t_2, \ldots, t_{\bar{m}}) \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l} \geq \sum_{t_2} \cdots \sum_{t_{\bar{m}}} F^i(r^i, t_2, \ldots, t_{\bar{m}}) \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l}$. Using the above inequality and Definition 4, we obtain $d_{it^i}(\underline{p}) \geq d_{ir^i}(\underline{p})$. Since $\underline{p}$ and $\underline{q}$ are two corners, hence multiplying both sides of the above inequality by one and adding zero doesn't change the value of the above inequality, thus we obtain $d_{it^i}(\underline{p})p_{it^i} + \sum_{r \neq t^i} d_{ir}(\underline{p})p_{ir} \geq d_{ir^i}(\underline{p})q_{ir^i} + \sum_{r \neq r^i} d_{ir}(\underline{p})q_{ir}$. Simplifying this inequality, we obtain $\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir}$, which contradicts the assumption that $\underline{p}$ is not compatible. ■

**Corollary 14** *A corner* $\underline{p} = \left(\underline{e}_{t_1}^T, \underline{e}_{t_2}^T, \ldots, \underline{e}_{t_n}^T\right)^T$ *is fully compatible if and only if* $F^i(t_1, t_2, \ldots, t_{\bar{m}}) > F^i(r, t_2, \ldots, t_{\bar{m}})$ *for all* $r \neq t_i$.

**Proof:** The proof is trivial from Theorem 13. ■

## 3 Steady State Behavior of ACLA

In this section, we study the behavior of ACLA in which all LAs use the $L_{R-I}$ learning algorithm. To study the behavior of ACLA, we use the number of times each LA is activated during the operation of ACLA. To do this, we introduce the concept of *local time* for every LA (cell). The local time for every LA starts with one and incremented by one every time that LA is activated. Let $t_i$ be the local time for LA $A_i$. We now define a sequence of random variable $T_{t_i}$ $(\ldots < T_{t_i} < T_{t_i+1} < \cdots)$, where $T_{t_i}$ is the global time for the LA $A_i$ activated at local time $t_i$. Let $\alpha_i(T_{t_i})$ be the action chosen by LA $A_i$ at global time instant $T_{t_i}$ in its $t_i^{th}$ activation. We assume that the action of a LA at any time is the action chosen in its last activation, i.e. $\alpha_i(k) = \alpha_i(T_{t_i})$ for all $k \in [T_{t_i}, T_{t_i+1})$. Let $\underline{\Pi}$ be an $M \times M$ binary matrix with $\pi_{jj} = 1$ for $\sum_{l=1}^{i-1} m_l < j \leq \sum_{l=1}^{i} m_l$ if the LA $A_i$ is activated at each time instant and $\pi_{jj} = 0$ elsewhere, where $M = \sum_i m_i$. Using $L_{R-I}$ learning algorithm, process $\{\underline{p}(k)\}_{k \geq 0}$ is Markovian and can be described by the following difference equation.

$$\underline{p}(k+1) = \underline{p}(k) + \underline{\Lambda\Pi g}(\underline{p}(k), \underline{\beta}(k)), \tag{5}$$

where $\underline{\beta}(k)$ is composed of components $\beta_{iy}(k)$ (for $1 \leq i \leq n$ and $1 \leq y \leq m_i$), which are dependent on $\underline{p}$. $\underline{g}$ represents the learning algorithm, $\underline{\Lambda}$ is a $M \times M$ diagonal matrix with $\lambda_{jj} = b_i$ for $\sum_{l=1}^{i-1} m_l < i \leq \sum_{l=1}^{i} m_l$ and $\lambda_{jj} = 0$ elsewhere, and $b_i$ represents the learning parameter for LA $A_i$. Let $\underline{B} = (b_1, b_2, \ldots, b_n)^T$ to denote the learning parameters of learning automata in the ACLA. Let $\underline{\Theta}$ is an $M \times M$ diagonal matrix with $\theta_{jj} = \rho_i$ for $\sum_{l=1}^{i-1} m_l < j \leq \sum_{l=1}^{i} m_l$. So we have $\underline{\Theta} = \mathrm{E}\left[\underline{\Pi}\right]$. We assume that $\rho_i$ $(i = 1, \ldots, n)$ be time invariant through the operation of ACLA. Now define

$$\Delta\underline{p}(k) = \mathrm{E}[\underline{p}(k+1) - \underline{p}(k) | \underline{p}(k) = \underline{p}] = \underline{\Lambda\Theta g}(\underline{p}(k), \underline{\beta}(k)).$$

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

$$\Delta\underline{p}(k) = \underline{\Lambda\Theta f}\left(\underline{p}(k)\right). \tag{6}$$

Now using $L_{R-I}$ algorithm, the components of $\underline{p}(k)$ can be obtained as follows.

$$\begin{aligned} \Delta p_{iy}(k) &= b_i \rho_i p_{iy}(k) \left[1 - p_{iy}(k)\right] \mathrm{E}\left[\beta_{iy}(k)\right] \\ &\quad - b_i \sum_{r \neq y} p_{ir}(k) p_{iy}(k) \mathrm{E}\left[\beta_{ir}(k)\right] = b_i \rho_i f_{iy}(\underline{p}), \end{aligned} \tag{7}$$

where

$$\begin{aligned} f_{iy}(\underline{p}) &= p_{iy}(k) \sum_{r \neq y} p_{ir}(k) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p})\right] \\ &= p_{iy}(k) \left[d_{iy}(\underline{p}) - D_i(\underline{p})\right]. \end{aligned} \tag{8}$$

For different values of $\underline{B}$ and $\underline{\rho}$, equation (5) generates different processes and we shall use $p^{B\rho}(k)$ to denote these processes whenever the value of $\underline{B}$ and $\underline{\rho}$ are to be specified explicitly. Define a continuous-time interpolation of $\underline{p}_i(k)$, denoted by $\tilde{\underline{p}}_i^{B\rho}(t)$, and called *interpolated process*, whose components are defined by

$$\tilde{\underline{p}}_i^{B\rho}(t) = \underline{p}_i(k) \qquad t \in [kb_i\rho_i, (k+1)b_i\rho_i]. \tag{9}$$

The interpolated process $\{\underline{\tilde{p}}^{B\rho}(t)\}_{t\geq 0}$ is a sequence of random variables that takes values from $R^{m_1\times\dots\times m_n}$, where $R^{m_1\times\dots\times m_n}$ is the space of all functions that, at each point, are continuous on the right and have a limit on the left over $[0,\infty)$ and take values in $K$, which is a bounded subset of $R^{m_1\times\dots\times m_n}$. The objective is to study the limit of sequence $\{\underline{\tilde{p}}^{B\rho}(t)\}_{t\geq 0}$ as $\max\{b_1,\dots,b_n\}\to 0$. Since $0<\rho_i\leq 1$, when $\max\{b_1,\dots,b_n\}$ is sufficiently small, then $\{\underline{\tilde{p}}^{B\rho}(t)\}_{t\geq 0}$ will be a good approximation to the asymptotic behavior of (9) and (6) can be written as the following ordinary differential equation (ODE).

$$\underline{\dot{p}} = \underline{f}(\underline{p}), \tag{10}$$

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

$$\frac{dp_{iy}}{dt} = p_{iy}\left[d_{iy}(\underline{p}) - D_i(\underline{p})\right] \tag{11}$$

We are interested in characterizing the long term behavior of $\underline{p}(k)$ and hence the asymptotic behavior of ODE (10). The analysis of process $\{\underline{p}(k)\}_{k\geq 0}$ is done in two stages. In the first stage, we solve ODE (10) and in the second stage, we characterize the solution of this ODE. The solution of ODE (10) approximates the asymptotic behavior of $\underline{p}(k)$ and the characteristics of this solution specify the long term behavior of $\underline{p}(k)$. The following Theorem gives the asymptotic behavior of $\underline{\tilde{p}}^{B\rho}$ as $\max\{b_1,\dots,b_n\}$ is sufficiently small. We show that the sequence of interpolated process $\{\underline{\tilde{p}}^{B\rho}(t)\}$ converges weakly to the solution of ODE (10) with initial configuration $\underline{p}(0)$. This implies that asymptotic behavior of $\underline{p}(k)$ can be obtained from the solution of ODE (10).

**Theorem 15** *Sequence $\{\underline{\tilde{p}}^{B\rho}(.)\}$ converges weakly to the solution of*

$$\frac{d\underline{X}}{dt} = \underline{f}(\underline{X}) \tag{12}$$

*with initial condition $\underline{X}(0)=X_0$ as $\max\{b_1,\dots,b_n\}\to 0$, where $X_0=\underline{\tilde{p}}^{B\rho}(0)$.*

**Proof:** The following conditions are satisfied by the learning algorithm (5).

(1) $\{\underline{p}(k),(\underline{\alpha}(k-1),\underline{\beta}(k-1))\}_{k\geq 0}$ is a Markov process.
(2) $(\underline{\alpha}(k),\underline{\beta}(k))$ takes values in a compact metric space.
(3) $\underline{g}$ is bounded, continuous and independent of $\underline{B}$.
(4) ODE (12) has a unique solution for each initial condition $\underline{X}(0)$.
(5) If $\underline{p}(k)=\underline{\bar{p}}$ is a constant, then $\{(\underline{\alpha}(k),\underline{\beta}(k))\}_{k\geq 0}$ is an independent identically distributed sequence. Let $M^{\bar{p}}$ be the distribution of process $\{(\underline{\alpha}(k),\underline{\beta}(k))\}_{k\geq 0}$.

Then using the weak convergence Theorem (Theorem 3.2)[12], sequence $\{\underline{\tilde{p}}^{B\rho}(.)\}$ converges weakly, as $\max\{b_1,\dots,b_n\}\to 0$ to the solution of

$$\frac{d\underline{X}}{dt} = \underline{\bar{f}}(\underline{X}), X(0)=X_0,$$

where $\underline{\bar{f}}(\underline{p}(k)) = \mathrm{E}_p f(\underline{p}(k),\underline{\alpha}(k),\underline{\beta}(k))$ and $\mathrm{E}_p$ denotes the expectation with respect to the invariant measure $M^{\bar{p}}$. Since for $\underline{p}(k)=\underline{\hat{p}}$, $(\underline{\alpha}(k),\underline{\beta}(k))$ is an independent identically distributed sequence whose distribution depends only on $\underline{\hat{p}}$ and the rule of ACLA, then we have

$$\underline{\bar{f}}(\underline{p}) = \mathrm{E}\left[\underline{f}(\underline{p}(k),\underline{\alpha}(k),\underline{\beta}(k))\right] = \underline{f}(\underline{p}),$$

and hence the Theorem. ■

Theorem 15 enables us to understand the long term behavior of $\underline{p}(k)$. The weak convergence in this Theorem implies that path $\underline{p}^{B\rho}(t)$ will closely follow the solution to the ODE on any finite interval with an arbitrarily high probability as $\max\{b_1,\dots,b_n\}\to 0$. As the length of the time interval increases and $\max\{b_1,\dots,b_n\}\to 0$, the fraction of time that the path of the ODE must eventually spend in a small neighborhood of $\underline{p}^o$, the solution of the ODE, goes to one. Thus, $\underline{p}^{B\rho}(.)$ will eventually (with an arbitrarily high probability) spend all of its time in a small neighborhood of $\underline{p}^o$ as well. The time interval over which the evolution of the ACLA follows the path of the ODE goes to infinity as $\max\{b_1,\dots,b_n\}\to 0$. Although the speed of convergence depends on the specific value of $\underline{B}$. The above point is summarized in Lemma 16.

**Lemma 16** *For large $k$ and small enough value of $\max\{b_1,\dots,b_n\}$, the asymptotic behavior of $\underline{p}(k)$ generated by the ACLA can be well approximated by the solution to ODE (12) with the same initial configuration.*

**Proof:** Let $\underline{X}(.)$ be the solution of ODE (12) with initial condition $\underline{X}(0)=X_0$ sufficiently close to an asymptotically stable configuration of the ODE, say $\underline{p}^o\in K$. For any $\underline{Y}(t)\in K$, $t\geq 0$ and any positive $T<\infty$, define

$$h_T(\underline{Y}) = \sup_{t\leq T} ||\underline{Y}(t)-\underline{X}(t)||.$$

Function $h_T(.)$ is continuous on $K$. Then Theorem 15 says that $\mathrm{E}\left[h_T(\underline{\tilde{p}})^{B\rho}\right]\to\mathrm{E}\left[h_T(\underline{X})\right]=0$ as $\max\{b_1,\dots,b_n\}\to 0$. The limit is zero since the value of $h_T(\underline{X})$ on the paths of limit process is zero with probability one. Thus, the sup over $t\in[0,T]$ of the distance between the original sequence $\underline{p}(t)$ and $\underline{X}(t)$ goes to zero in probability as $k\to\infty$. With particular initial condition used, let $\underline{p}^o$ be the equilibrium configuration to which the solution of the ODE converges. Using this and the nature of interpolation, given in (9), it is implied that for the given initial configuration and any $\epsilon>0$

and integers $k_1$ and $k_2$ ($0 < k_1 < k_2 < \infty$), there exists a $b_0$ such that for all $\max\{b_1, \ldots, b_n\} < b_0$, we have

$$\text{Prob}\left[\sup_{k_1 \le k \le k_2} \left|\left|\underline{p}(k) - \underline{p}^o\right|\right| > \epsilon\right] = 0.$$

Since $\underline{p}^o$ is an asymptotically stable equilibrium point of ODE (10), then for all initial configurations in small neighborhood of $\underline{p}^o$, the ACLA converges to $\underline{p}^o$. ∎

In the following subsections, we first find the equilibrium points of ODE (10), then study the stability property of equilibrium points of ODE (10), and finally state a main Theorem about the convergence of the ACLA.

### *3.1 Equilibrium Points*

The equilibrium points of equation (6) are those points that satisfy the set of equations $\Delta p_{ij}(k) = 0$ for all $i, j$, where the expected changes in the probabilities are zero. In other words, the equilibrium points are zeros of $\underline{f}\left(\underline{p}\right)$, which are studied in the following two Lemmas.

**Lemma 17** *All corners of $K$ are equilibrium points of $\underline{f}(.)$. Other equilibrium points $\underline{p}$ satisfy $d_{iy}(\underline{p}) = d_{ir}(\underline{p})$, for all $r, y \in \{1, 2, \ldots, m_i\}$, and for all $i$.*

**Proof:** From (8) it is obvious that each of the $f_{iy}$ (for all $i = 1, \ldots, n$ and $y = 1, \ldots, m_i$) is zero if $\underline{p}_i$ is a unit vector. To find the other zeros of $\underline{f}(.)$, it is obvious from (8) that $f_{iy} = 0$ if $p_{iy} = 0$. But $\underline{p}_i$ is a probability vector and all components of $\underline{p}_i$ can not be zero at the same time because $\sum_y p_{iy} = 1$. When $p_{iy} \neq 0$, for $f_{iy}$ to be zero, we must have

$$\sum_{r \neq y} p_{ir}(k)\left[d_{iy}(\underline{p}) - d_{ir}(\underline{p})\right] = 0. \tag{13}$$

This can be rewritten as $\sum_{r\neq y} p_{ir}(k)\left[d_{iy}(\underline{p}) - d_{ir}(\underline{p})\right] = d_{iy}(\underline{p}) - d_{iq}(\underline{p}) + \sum_{r \neq q}\left[d_{iq}(\underline{p}) - d_{ir}(\underline{p})\right] p_{ir}(k)$. Thus, we obtain

$$\sum_{r \neq q}\left[d_{iq}(\underline{p}) - d_{ir}(\underline{p})\right] p_{ir}(k) = d_{iq}(\underline{p}) - d_{iy}(\underline{p}), \tag{14}$$

for $y = 1, \ldots, m_i$ and $y \neq q$. The left hand side of the above equation for all $y \neq q$ is the same, say as $d_0$. Thus, we have

$$d_{iq}(\underline{p}) - d_{i1}(\underline{p}) = \ldots = d_{iq}(\underline{p}) - d_{im_i}(\underline{p}) = d_0.$$

When $d_0 \neq 0$, (14) implies that $\sum_{r\neq q} p_{ir}(k) = 0$, corresponding to the unit vector $\underline{e}_q$ and considered already. When $d_0 = 0$, then the $\underline{p}$ that results $\underline{f}(\underline{p})$ be zero must satisfy $d_{iq}(\underline{p}) = d_{iy}(\underline{p})$, for all $i = 1, 2, \ldots, n$ and all $y \neq q$. When some $p_{iy}$ are zero, for $\underline{f}$ to be zero, equation (13) must be satisfied for all $1 \le y \le m_i$ such that $p_{iy} \neq 0$ for each $i$, which completes the proof. ∎

**Lemma 18** *All compatible configurations are equilibrium points of $\underline{f}(.)$.*

**Proof:** Let $\underline{p}$ be a compatible configuration. Then from Lemma 11, $d_{ir}(\underline{p}) = D_i(\underline{p})$ and hence $f_{ir}(\underline{p}) = 0$ for all $i$ and $r$. ∎

### *3.2 The Stability Property*

In this subsection we characterize the stability of equilibrium configurations of ACLA, that is the equilibrium points of the ODE (10). From the Lemmas 17 and 18, all equilibrium points of (10) are known. In order to study the stability of the equilibrium points of (10), the origin is transferred to the equilibrium point under consideration and then the linear approximation of the ODE is studied. The following two Lemmas are concerned with the stability properties of these equilibrium points.

**Lemma 19** *A corner $\underline{p}^o \in K^*$ is fully compatible if and only if it is uniformly asymptotically stable.*

**Proof:** Let configuration $\underline{p}^o = \left(\underline{e}^T_{t_1}, \ldots, \underline{e}^T_{t_n}\right)^T$ be a corner of $K$ that is a fully compatible. Using the transformation defined by

$$\tilde{p}_{iy} = \begin{cases} p_{iy} & \text{if } y \neq t_i \\ 1 - p_{iy} & \text{if } y = t_i \end{cases}$$

the origin is translated to $\underline{p}^0$. Since $\underline{p}_i$ ($1 \le i \le n$) is a probability vector, then only $\sum_i (m_i - 1)$ components of $\underline{p}^o$ are independent. Suppose that $p_{ir}$ for $r \neq t_i$ (for $1 \le i \le n$) be the independent components. Using Taylor's expansion, $f_{iy}$ can be expressed as

$$\begin{aligned} f_{iy} =& \tilde{p}_{iy}[F^i(y, t_2, \ldots, t_{\bar{m}}) - F^i(t_i, t_2, \ldots, t_{\bar{m}})] \\ &+ \text{high order terms.} \end{aligned} \tag{15}$$

We consider the following positive definite Lyapunov function $V\left(\underline{\tilde{p}}\right) = \sum_i \sum_{y \neq t_i} \tilde{p}_{iy}$, where $V\left(\underline{\tilde{p}}\right) \ge 0$ and is zero when $\tilde{p}_{iy} = 0$ for all $i, y$, and its derivative is equal to $\dot{V}\left(\underline{\tilde{p}}\right) = \sum_i \sum_{y\neq t_i} f_{iy}$. Since corner $\underline{p}^o$ is fully compatible, then from Theorem 13, we have $F^i(y, t_2, \ldots, t_{\bar{m}}) - F^i(t_i, t_2, \ldots, t_{\bar{m}}) < 0$ for all $i$. Thus, equation (15) implies that there is a neighborhood around $\underline{p}^o$ such that the linear terms dominate the high order terms. Hence, $\dot{V}\left(\underline{\tilde{p}}\right) < 0$ and $\underline{p}^o$ is uniformly asymptotical stable.

Conversely, assume that $\underline{p}^o$ is uniformly asymptotical stable, then the linear approximation of ODE (10) can be written as $\dot{\underline{\tilde{p}}} = A\underline{\tilde{p}}$, where $A = \text{diag}\left(\tilde{f}_{iy}\right)$ and $\tilde{f}_{iy} = F^i(y, t_2, \ldots, t_{\bar{m}}) - F^i(t_i, t_2, \ldots, t_{\bar{m}})$ for all $i$. Since $\underline{p}^o$ is uniformly asymptotical stable, $A$ should have eigenvalues with negative real parts and hence $\tilde{f}_{iy} < 0$. Using Theorem 13, this implies that $\underline{p}^o$ is fully compatible. ■

**Lemma 20** *Incompatible equilibrium points of $\underline{f}(.)$ are unstable.*

**Proof:** Let $\underline{p}^o$ be an equilibrium point of $\underline{f}(.)$ which is not compatible. Then by Lemma 10, there is a LA $A_j$ and an action $y$ such that $d_{jy}(\underline{p}) > D_j(\underline{p})$. Since $d_{jy}(\underline{p})$ and $D_j(\underline{p})$ are continuous, then inequality $d_{jy}(\underline{p}) > D_j(\underline{p})$ will hold in small open neighborhood around $\underline{p}^o$. Using (11), it is implied that for all points in this neighborhood $\frac{dp_{jy}}{dt} > 0$ if $p_{jy} \neq 0$. Hence, no matter how small this neighborhood we take, there will be infinity many points starting from which, $\underline{p}(k)$ will eventually leave that neighborhood, which implies that $\underline{p}^o$ is unstable. ■

**Remark 21** *In Lemmas 19 and 20, the solution of ODE (10) well characterized and it is shown that full compatibility implies uniformly asymptotic stability of the corners. In order to obtain necessary and sufficient conditions for uniformly asymptotic stability, it is essential to consider in detail the nonlinear terms in the differential equation, which appears to be a difficult problem.*

**Remark 22** *Almost sure convergence method [13] can be used to show the convergence of ACLA. Using this method, it can be shown that the evolution of ACLA essentially follow the solution to the ODE (for large k) and also if ACLA configuration enters the domain of attraction of an asymptotically stable configuration $\underline{p}^o$ infinitely often, it will eventually converges to $\underline{p}^o$. Therefore if we use almost sure methods then the stability analysis performed above is not needed.*

### *3.3 Convergence Results*

We study convergence of ACLA for the following different initial configurations, which covers all points in $K$.

(1) $\underline{p}(0)$ is close to a compatible corner $\underline{p}^o$. By Lemma 19, there is a neighborhood around $\underline{p}^o$ entering which, the ACLA will be absorbed by that corner. Thus, the ACLA converges to a compatible configuration.
(2) $\underline{p}(0)$ is close to a corner $\underline{p}^o$ which is not compatible. By Lemma 20, no matter how small neighborhood we take around $\underline{p}^o$, the solution of (10) will leave that neighborhood and enter $K - K^*$. The convergence when the initial configuration is in $K - K^*$ is discussed in case 4 below.
(3) $\underline{p}(0) \in K^*$. Using the convergence properties of $L_{R-I}$ learning algorithm [2], no matter whether $\underline{p}(0)$ is compatible or not, the ACLA will be absorbed to $\underline{p}(0)$.
(4) $\underline{p}(0) \in K - K^*$. The convergence results of the ACLA for these initial configurations is stated in Theorem 23.

**Theorem 23** *Suppose there is a bounded differential function $D : R^{m_1+\cdots+m_n} \to R$ such that for some constant $c > 0$, $\frac{\partial D}{\partial p_{ir}}(\underline{p}) = cd_{ir}(\underline{p})$ for all $i$ and $r$, and $\rho_i > 0$ for all $i$. Then ACLA for any initial configuration in $K - K^*$ and with sufficiently small value of learning parameter ($\max\{b_1, \ldots, b_n\} \to 0$), always converges to a configuration, that is stable and compatible.*

**Proof:** Consider the variation of $D$ along the solution paths of ODE (10), $D$ is non-decreasing because

$$\begin{aligned}\frac{dD}{dt} &= \sum_i \sum_y \frac{\partial D}{\partial p_{iy}} \frac{\partial p_{iy}}{\partial t} \\ &= c \sum_i \sum_y \sum_{r>y} p_{iy} p_{ir} \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p})\right]^2 \geq 0.\end{aligned} \tag{16}$$

ACLA updates the action probabilities in a such a way that $\underline{p}(k) \in K$ for all $\underline{p}(0) \in K$ and $k > 0$. Since $K$ is a compact subset of $R^{m_1+\cdots+m_n}$, asymptotically all solutions of ODE (10) will be in $K$. Inequality (16) shows that ACLA updates the configuration probabilities in gradient ascent manner and hence, converges to a maximum of $D$, where $\frac{dD}{dt} = 0$. From (16), the derivative of $D$ is zero if and only if for all $i, y, r$, we have $p_{ir}p_{iy} = 0$ or $d_{iy}(\underline{p}) = d_{ir}(\underline{p})$. From Lemmas 17 and 18, these configurations are equilibrium points of $f_{iy}(\underline{p})$. Thus the solution to ODE (10) for any initial configuration in $K - K^*$ will converge to a set containing only equilibrium points of the ODE (10). Since all equilibrium configurations that are not compatible are unstable, the Theorem follows. ■

**Remark 24** *If the ACLA satisfies the sufficiency conditions needed for Theorem 23, then ACLA will converge to a compatible configuration. When the ACLA doesn't satisfy this sufficiency condition, convergence to compatible configurations cannot be guaranteed and the ACLA may exhibit a limit cycle behavior [14].*

## 4 ACLA using Commutative Rules

In this section, we study the behavior of the ACLA when the commutative rules are used. Commutativity is a property of $F^i$ as given in the following Definition.

**Definition 25** *Rule $F^i$ is called commutative if and only if we have $F^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \alpha_{i+\bar{x}_3}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) = F^i(\alpha_{i+\bar{x}_{\bar{m}}}, \alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}-1}}) = \ldots\ldots = F^i(\alpha_{i+\bar{x}_2}, \alpha_{i+\bar{x}_3}, \alpha_{i+\bar{x}_4}, \ldots, \alpha_{i+\bar{x}_1})$.*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | $\cdots$ | $i-1$ | $i$ | $i+1$ | $\cdots$ | $n-1$ | $n$ |

Fig. 1. The linear ACLA.

In order to simplify the algebraic manipulations, we analyze linear ACLA. The linear ACLA (figure 1) uses neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. The following Theorem is an additional property for compatible configurations in ACLA using commutative rules.

**Theorem 26** *For a ACLA, which uses a commutative rule, a configuration $\underline{p}$ at which $D(\underline{p})$ is a local maximum, is compatible.*

**Proof:** Since $K$ is convex, then for every $0 \leq \gamma \leq 1$ and $\underline{q} \in K$, we have $\gamma\underline{q} + (1-\gamma)\underline{p} \in K$. Suppose that $\underline{p}$ is a configuration that $D(\underline{p})$ is local maximum, then $D(\underline{p})$ doesn't increase as one moves away from $\underline{p}$. Thus we have $\left.\frac{dD(\gamma\underline{q}+(1-\gamma)\underline{p})}{d\gamma}\right|_{\gamma=0} \leq 0$. Thus using chain rule, we obtain $\nabla D(\underline{p})(\underline{q}-\underline{p}) \leq 0$. $\nabla D(\underline{q})$ has $M$ elements in which its $(l,r)^{th}$ component, denoted by $q_{lr}$, where $q_{lr} = \frac{\partial}{\partial p_{lr}} D(\underline{p}) = 3d_{lr}(\underline{p})$. Thus, $\nabla D(\underline{p})(\underline{q}-\underline{p}) \leq 0$ implies that $\nabla D(\underline{p})(\underline{q}-\underline{p}) = 3\sum_i \sum_y d_{iy}(\underline{p})\left[q_{iy} - p_{iy}\right] \leq 0$. This inequality is true for all $\underline{q} \in K$. So, $\underline{p}$ satisfies the condition of Theorem 12, implying compatibility of $\underline{p}$. ■

**Remark 27** *In general, when rules are not commutative, local maxima for $D(\underline{p})$ still exist, but they may not be compatible.*

Now, using the analysis given in section 3, we can state the main Theorem for the convergence of the ACLA when it uses commutative rules.

**Theorem 28** *An ACLA, which uses uniform and commutative rule, starting from $\underline{p}(0) \in K - K^*$ and with sufficiently small value of learning parameter,* $(\max\{b_1, \ldots, b_n\} \to 0)$, *always converges to a deterministic configuration, that is stable and also compatible.*

**Proof:** Let $D : R^{m_1+\cdots+m_n} \to R$ be average reward of ACLA. Hence, for all $i$ and $r$ we have $\frac{\partial D}{\partial p_{ir}}(\underline{p}) = 3d_{ir}(\underline{p})$. Using Theorem 23 the convergence can be concluded. ■

**Remark 29** *From Theorem 28, we can conclude that the ACLA converges to one of its compatible configurations, if any. If this compatible configuration is unique, then ACLA converges to this configuration for which $D(\underline{p})$ is the maximum. If there are more than one compatible configurations, then the ACLA depending on the initial configuration $\underline{p}(0)$ may converge to one of its compatible configurations for which $D(\underline{p})$ is a local maximum, .*

**Remark 30** *Theorem 28 guarantees that limit cycle for ACLA does not exist and ACLA always converges to an equilibrium of ODE.*

## 5 Computer Experiments

In this section, we give two computer experiments: 1) patterns formed by the evolution of ACLA from random initial configuration, and 2) application of ACLA to call admission control in cellular mobile networks.

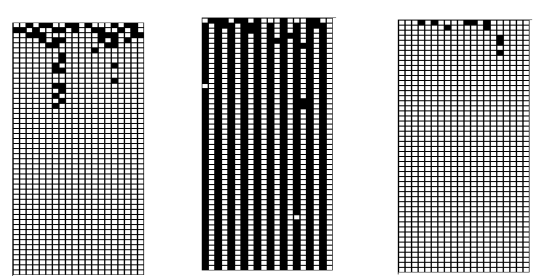
### *5.1 Numerical Examples*

This section discusses configurations to which ACLA converges. Different ACLA rules are found to converges different configurations. Each cell has a LA with $m$ actions. The actions of each LA are represented by integers in interval $[0, m-1]$. Hence, the configuration of each cell and its neighbors forms a $\bar{m}$ digits number in interval $[0, m^{\bar{m}} - 1]$ with $m^{\bar{m}}$ possible values. The value of reinforcement signal for all of the above $m^{\bar{m}}$ configurations constitute an $m^{\bar{m}}$ bit number. Then the rule can be identified by decimal representation of this $m^{\bar{m}}$-bit number. For the sake of simplicity in our presentation, we use notation $(j)_m$ to specify the rules in the ACLA, where $j$ is a decimal number representing the rule when the LA has $m$ actions. For example, the following table represents the rule 22 for a linear ACLA with two-actions LA and the neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. For this case, each of the eight possible configurations for a cell and its neighbors appear on the first row, while the second row gives the value of the reinforcement signal to be output to the central cell on the next time step.

Table 1
The scheme for the rule numbering for two actions LA

| Configuration | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 |
|---|---|---|---|---|---|---|---|---|
| $\beta$ | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

Figure 2 shows the time-space diagram evolution of ACLA with 20 cells and a LA with two-actions using $L_{R-I}$ learning algorithm in each cell. The probability of activation of LA is the following fixed vector $\rho = ($ 0.1, 0.1, 0.2, 0.1, 0.3,0.5, 0.6, 0.8, 0.1, 0.1, 0.2, 0.1, 0.3, 0.5, 0.6, 0.8, 0.1, 0.1, 0.2, 0.1$)^T$. The simulation results show that the ACLA converges to a configuration in $K^*$ rather than to a configuration in $K - K^*$.



rule $(37)_2$ rule $(38)_2$ rule $(39)_2$

Fig. 2. Time-space diagram for asynchronous CLA

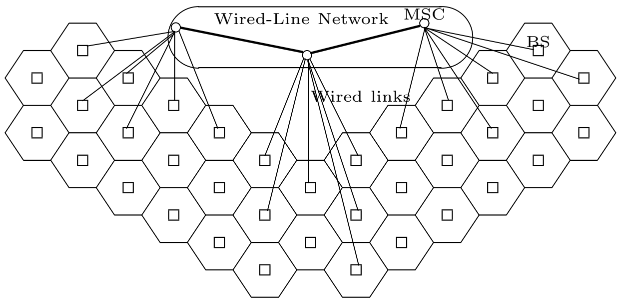



Fig. 3. System model of cellular networks.

### 5.2 Call Admission in Mobile Cellular Networks

Introduction of micro cellular networks (figure 3) leads to efficient use of channels but increases the expected rate of handovers per call. As a consequence, some network performance parameters such as *blocking probability of new calls* ($B_n$) and *dropping probability of handoff calls* ($B_h$) are affected. Call admission algorithms control $B_n$ and $B_h$ by putting some restrictions on the allocation of channels to the incoming calls. Since the dropping of handoff calls is more important than the blocking of new calls, call admission algorithms usually put restriction on the acceptance of new calls. Assume that the given cell has $C$ full duplex channels. The simplest call admission algorithm, which is called *guard channel algorithm*, reserves a subset of the channels allocated to the cell, called *guard channels*, for sole use of the handoff calls. In the guard channel algorithm, when the channel occupancy exceeds certain threshold $T$, then new calls are rejected until the channel occupancy goes below threshold $T$ [15]. The guard channel algorithm accepts handoff calls as long as channels are available. It has been shown that there is an optimal threshold $T^*$ for which the blocking probability of the new calls is minimized subject to the constraint on the dropping probability of handoff calls [15]. Although, the guard channel algorithm decreases the dropping probability of the handoff calls, but the blocking probability of the new calls may be degraded to a great extent.

The guard channel scheme assumes that the input traffic is a stationary process with known parameters. But in real applications, the input traffic is not a stationary process with known parameters and hence the optimal number of guard channels may be different for different traffic. In such cases the *dynamic guard channel* algorithm, in which the number of guard channels varies during the operation of the cellular network, can be used. In this section, we consider the dynamic guard channel algorithms for cellular networks and present an algorithm based on ACLA to determine the number of guard channels when the traffic parameters are unknown and possibly time varying. We assume that the cellular network has $n$ cells each of which with $C$ assigned channels and at most $m-1$ guard channels will be used. We embed the cellular network into an ACLA with $n$ cells and a LA with $m$ actions using $L_{R-I}$ learning algorithm is assigned to each cell. This way, each cell of the ACLA is mapped into one of the cells of the cellular network. The LA associated to a cell will be responsible for adaptation of the number of guard channels for the that cell in the network. The neighbors of a cell are those cells which affect the rate of handovers to that cell, for examples cells with common border or cells whose centers are at a given distance. The local rule for ACLA is given by

$$F^u(.) = \begin{cases} 1 & \text{if } B_h < p_h \\ 1 - \left| \dfrac{\sum_{\substack{v \in \tilde{N}(u) \\ v \neq u}} [g_v - g_u]}{[\bar{m}-1]C} \right| & \text{otherwise,} \end{cases} \tag{17}$$

where $g_u$ is the number of guard channels used in cell $u$, which takes values in interval $[0, m-1]$ and $p_h$ is the predefined upper bound for dropping probability of handoff calls. The above rule is designed using the following points.

(1) When the number of guard channels in a cell is less than the average number of guard channels in its neighboring cells, one may conclude that the neighboring cells have faced high handoff traffic rate, which in this case the number of guard channels in this cell must be increased.
(2) When the number of guard channels in a cell is greater than the average number of guard channels in its neighboring cells, one may conclude that the neighboring cells have faced low handoff traffic rate, which in this case the number of guard channels in this cell must be decreased

The algorithm performed by cell $u$ can be described as follows. When a handoff call arrives, it will be accepted provided that the cell has free channels. When a new call arrives at cell $u$, the LA associated to that cell chooses one of its actions, say $g_u$, and based on this action, the number of guard channels is determined. If the cell has at least $g_u$ free channels, then the call will be accepted; otherwise the call will be blocked. Finally, the local rule of the ACLA is computed according to (17) and depending on the result of the local rule, the action probability vector of cell $u$ will be updated accordingly. The reinforcement signal to an action chosen by a cell is produced in two stages. In the first stage, the base station of cell $u$ computes the current estimate of the dropping probability and in the second stage, the reinforcement signal is computed using (17).

#### 5.2.1 Numerical Example

In what follows, we compare the performance of the weighted sum scheme [16] and the proposed algorithm. The simulation is conducted on the homogenous linear cellular network with 18 cells. Each cell has 8 full duplex channels ($C = 8$). We assume that the arrival of calls is Poisson process with rate $\lambda$ and the duration of calls and their dwell time [1] are exponentially distributed with mean 18 and 6, respectively. We further assume that the mobile users in the network are moved with constant speed in random direction. We conducted simulations for $p_h = 0.01$ and different values of $\lambda$. The results of these simulations are shown in figures 4 and 5. The simulation results show that the blocking probability of new calls for the proposed algorithm is lower than the blocking probability of new calls of the weighted sum scheme while the level of QoS is maintained by two algorithms.

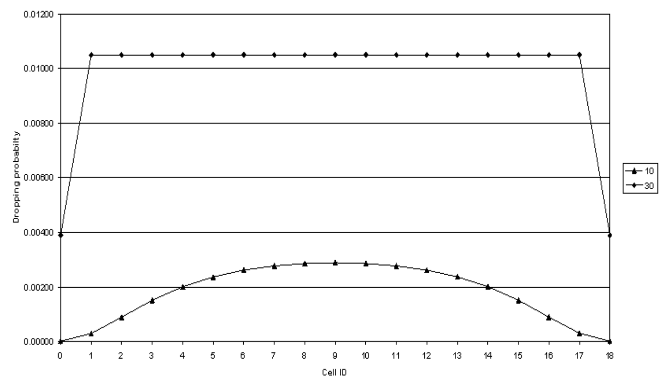



Fig. 4. The dropping probability of handoff calls for dynamic guard channel algorithm.

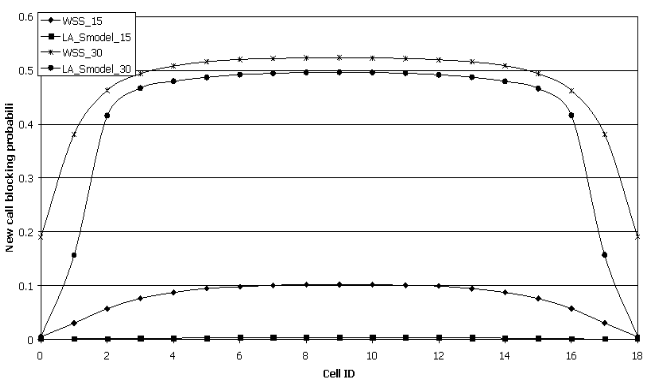



Fig. 5. The blocking probability of new calls for dynamic guard channel algorithm and weighted sum scheme.

## 6 Conclusions

In this paper, the ACLA was proposed. It was shown that for a class of rules, called commutative rules, the ACLA converges to a configuration for which its average rewards is maximized. To further study the behavior of ACLA, we conducted an experiment in which we studied the patterns formed by the evolution of ACLA from a random initial configurations. The results obtained from experiments confirms the theory. Finally, an application of ACLA to call admission control in mobile cellular networks was presented. An algorithm based on ACLA for call admission control in mobile cellular networks was proposed. Comparing the results of the experiments with the results obtained for one of the efficient existing call admission algorithms showed the superiority of the proposed algorithm.

[1] The average time that a mobile user resides in each cell.

## References

[1] B. Schonfisch and A. D. Ross, "Synchronous and Asynchronous Updating in Cellular Automata," *Journal of Biosystem*, vol. 51, pp. 123–143, 1999.

[2] K. S. Narendra and K. S. Thathachar, *Learning Automata: An Introduction*. New York: Printice-Hall, 1989.

[3] B. J. Oommen and E. V. de St. Croix, "Graph Partitioning Using Learning Automata," *IEEE Transactions on Computers*, vol. 45, pp. 195–208, Feb. 1996.

[4] H. Beigy and M. R. Meybodi, "An Adaptive Call Admission Algorithm for Cellular Networks," *Electerical and Computer Engineering*, vol. 31, pp. 132–151, Mar. 2005.

[5] B. J. Oommen and T. D. Roberts, "Continuous Learning Automata Solutions to the Capacity Assignment Problem," *IEEE Transactions on Computers*, vol. 49, pp. 608–620, June 2000.

[6] M. R. Meybodi and H. Beigy, "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters," *Journal of Neurocomputing*, vol. 48, pp. 957–974, Nov. 2002.

[7] M. R. Meybodi and H. Beigy, "New Learning Automata Based Algorithms for Adaptation of Backpropagation Algorithm Parameters," *International Journal of Neural Systems*, vol. 12, pp. 45–68, Feb. 2002.

[8] M. R. Meybodi, H. Beigy, and M. Taherkhani, "Cellular Learning Automata and its Applications," *Sharif Journal of Science and Technology*, vol. 19, no. 25, pp. 54–77, 2003.

[9] M. R. Meybodi and M. R. Kharazmi, "Application of Cellular Learning Automata to Image Processing," *Journal of Amirkabir*, vol. 14, no. 56A, pp. 1101–1126, 2004.

[10] H. Beigy and M. R. Meybodi, *A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach*, vol. 2690 of *Springer-Verlag Lecture Notes in Computer Science*, pp. 119–126. Springer-Verlag, 2003.

[11] M. R. Meybodi and H. Beigy, "A Mathematical Framework for Cellular Learning Automata," *Journal of Advances in Complex Systems*, vol. 7, pp. 295–320, Sept. 2004.

[12] H. J. Kushner, *Approximation and Weak Convergence Methods for Random Processes.* Cambridge: MIT Press, 1984.

[13] M. F. Norman, *Markovian Process and Learning Models.* New York: Academic Press, 1972.

[14] K. S. Narendra and A. M. Annaswarmy, *Stable Adaptive Systems.* New York: Printice-Hall International Inc., 1989.

[15] R. Ramjee, D. Towsley, and R. Nagarajan, "On Optimal Call Admission Control in Cellular Networks," *Wireless Networks*, vol. 3, pp. 29–41, Mar. 1997.

[16] J. M. Peha and A. Sutivong, "Admission Control Algorithms for Cellular Systems," *Wireless Networks*, vol. 7, pp. 117–125, Mar. 2001.