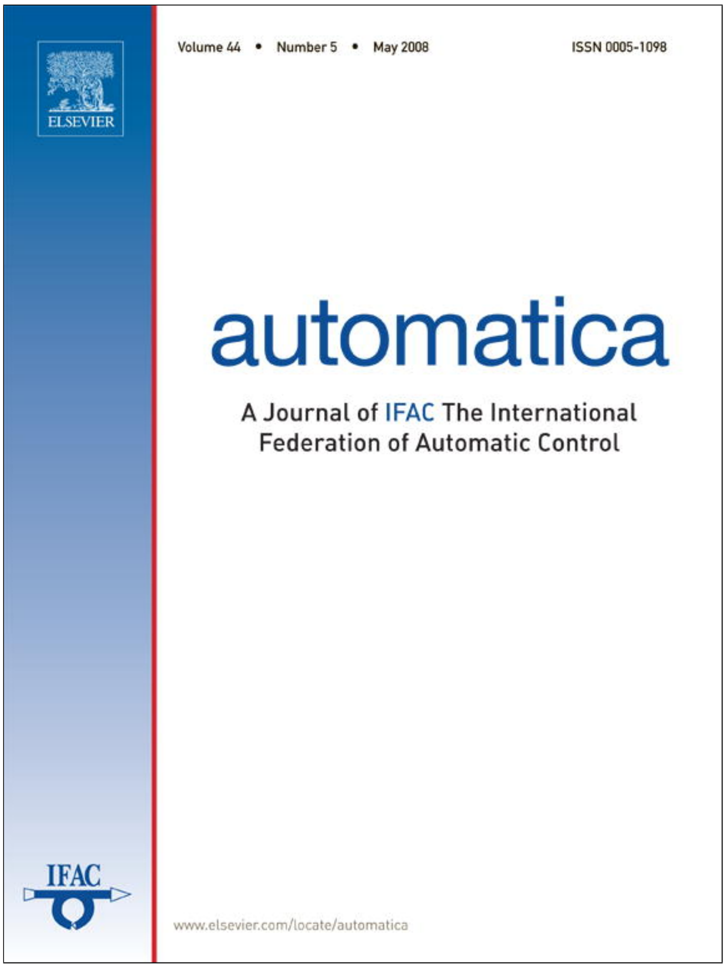
Volume 44 • Number 5 • May 2008
ISSN 0005-1098
ELSEVIER
automatica
A Journal of IFAC The International Federation of Automatic Control
IFAC
www.elsevier.com/locate/automatica

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Automatica 44 (2008) 1350–1357
automatica
www.elsevier.com/locate/automatica

Brief paper

Asynchronous cellular learning automata[☆]

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

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

Received 4 January 2004; received in revised form 27 August 2007; accepted 30 September 2007
Available online 11 March 2008

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.

Keywords: 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, and 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 are referred to as *updating methods*. In *synchronous updating*, the state of all cells are changed at the same time and in *asynchronous updating*, the state of cells are changed one after another (Schonfisch & Ross, 1999). Asynchronous updating requires the specification of an order in which cells are updated.

On the 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 the action chosen by the LA, the environment gives a reinforcement signal with unknown probability distribution. Then based on this signal, the LA updates its action probability vector using a learning algorithm. A class of LAs called *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 (Narendra & Thathachar, 1989). In *linear reward-inaction learning algorithm* (L_{R-I}), the action probability vector is updated using

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

when the environment rewards and the action probability vector remains unchanged when the environment penalizes the action. Parameters $b \in (0, 1)$ and r represent *learning parameter* and the number of actions for LA, respectively and α_i is the action chosen at stage k as a sample realization from probability distribution $\underline{p}(k)$ (Narendra & Thathachar, 1989). LA have been used successfully in many applications such as solving NP-Complete problems (Oommen & de St Croix, 1996), call admission in cellular mobile networks (Beigy &

[☆] This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Antonio E. de Barros Ruano under the direction of Editor Miroslav Krstic.

[*] Corresponding author at: Computer Engineering Department, Sharif University of Technology, Tehran, Iran

E-mail addresses: beigy@ce.sharif.edu (H. Beigy), mmeybodi@aut.ac.ir (M.R. Meybodi).

0005-1098/$ - see front matter © 2008 Published by Elsevier Ltd
doi:10.1016/j.automatica.2007.09.018

Meybodi, 2005), capacity assignment (Oommen & Roberts, 2000) and neural network engineering (Meybodi & Beigy, 2002) to mention a few.

Learning automata are, by design, "simple agents for doing simple things". The full potential of LAs 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. On the other hand, CA are mathematical models for systems consisting of large numbers of simple identical components with local interactions. In Meybodi, Beigy, and Taherkhani (2003), 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. A CLA is a CA in which a LA is assigned to its every cell. The LA 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. A number of applications for synchronous CLA have been developed recently such as channel assignment in cellular networks (Beigy & Meybodi, 2003). In Beigy and Meybodi (2004), 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. Section 5 presents the numerical example and an application of ACLA to cellular mobile networks 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. 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 activated in fixed or random sequence. The step-driven ACLA is of more interest to us because of its applications. Formally a 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 Z^d is a lattice of d-tuples of integer numbers, $\underline{\Phi}$ is a finite set of states, A is the set of LAs assigned to cells, $N = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{\bar{m}}\}$ is neighborhood vector, $F : \underline{\Phi}^{\bar{m}} \to \underline{\beta}$ is the local rule, where $\bar{m}$ is the number of neighboring cells of each cell, and $\underline{\rho}$ is an n-dimensional vector called activation probability vector, where ρ_i is the probability that the LA in cell i (for $i = 1, \ldots, n$) to be activated in each stage. We assume that, there exists a neighborhood function $\bar{N}(u)$ mapping cell u to the set of its neighbors, that is $\bar{N}(u) = (u + \bar{x}_1, u + \bar{x}_2, \ldots, u + \bar{x}_{\bar{m}})$.

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 $\underline{p} = (\underline{p}_1^{\mathrm{T}}, \underline{p}_2^{\mathrm{T}}, \ldots, \underline{p}_n^{\mathrm{T}})^{\mathrm{T}}$, where $\underline{p}_i = (p_{i1}, \ldots, p_{im_i})^{\mathrm{T}}$ is the action probability vector of LA A_i. 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 $F^i\left(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2} \ldots, \alpha_{i+\bar{x}_{\bar{m}}}\right)$ is denoted by $F^i\left(\alpha_1, \alpha_2 \ldots, \alpha_{\bar{m}}\right)$.

The operation of ACLA takes place as the following iterations. At stage k, each LA A_i is activated with probability ρ_i and the activated LAs choose one of their actions. Then local rule determines the reinforcement signal based on the most recently selected actions of activated LAs. The higher value of this signal means that the chosen action is more rewarded. Finally, each activated LA updates its action probability vector on the basis of the reinforcement signal and the chosen action. This process continues until the desired result is obtained.

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.

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

Definition 3. Let $\underline{p}$ be a configuration where $\underline{p} = (\underline{p}_1^{\mathrm{T}}, \underline{p}_2^{\mathrm{T}}, \ldots, \underline{p}_n^{\mathrm{T}})^{\mathrm{T}}$ and $\underline{p}_i = (p_{i1}, \ldots, p_{im_i})^{\mathrm{T}}$. 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 $K^* = \{\underline{p} | p_{iy} \in \{0, 1\}\ \forall y, i\}$, and $K = \{\underline{p} | p_{iy} \in [0, 1]\ \forall y, i\}$, 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^* (Beigy & Meybodi, 2004).

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

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

and the average reward for this LA 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.

Lemma 6. *ACLA has at least one compatible configuration.*

Proof. The proof for this lemma is similar to the proof of Lemma 2 in Beigy and Meybodi (2004). ■

Lemma 7. *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 Beigy and Meybodi (2004). ■

Lemma 8. *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 Beigy and Meybodi (2004). ■

Theorem 9. *Configuration $\underline{p}$ is compatible if and only if for all $\underline{q} \in K$, $\sum_i \sum_y d_{iy}(\underline{p}) \left[p_{iy} - q_{iy}\right] \geq 0$ holds.*

Proof. The proof for this theorem is given in Beigy and Meybodi (2004). ■

This Theorem states that, when the action probability vector of all LAs except 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 10. *A corner $\underline{p} = \left(\underline{e}^{\mathrm{T}}_{t^1}, \dots, \underline{e}^{\mathrm{T}}_{t^n}\right)^{\mathrm{T}}$ is compatible if and only if for all $i = 1, \dots, n$ and $r^i \neq t^i$ we have $F^i(t^i, t_2, \dots, t_{\bar{m}}) \geq F^i(r^i, t_2, \dots, t_{\bar{m}})$.*

Proof. We first show that if $\underline{p}$ is compatible then $F^i(t^i, t_2, \dots, t_{\bar{m}}) \geq F^i(r^i, t_2, \dots, t_{\bar{m}})$. To show this, we assume that $\underline{q} = \left(\underline{e}^{\mathrm{T}}_{t^1}, \dots, \underline{e}^{\mathrm{T}}_{r^i}, \dots, \underline{e}^{\mathrm{T}}_{t^n}\right)^{\mathrm{T}}$ 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 Eq. (3), we obtain $F^i(t^i, \dots, t_{\bar{m}}) \geq F^i(r^i, \dots, t_{\bar{m}})$, concluding this assertion. Conversely, assume that $F^i(t^i, t_2, \dots, t_{\bar{m}}) \geq F^i(r^i, t_2, \dots, t_{\bar{m}})$ but $\underline{p}$ is not compatible. Multiplying both sides of the above inequality by $\prod_{l \in \bar{N}(i), l \neq i} p_{l t_l}$ and summing over all actions of neighboring LAs of cell i, we obtain $\sum_{t_2} \dots \sum_{t_{\bar{m}}} F^i(t^i, t_2, \dots, t_{\bar{m}}) \prod_{l \in \bar{N}(i), l \neq i} p_{l t_l} \geq \sum_{t_2} \dots \sum_{t_{\bar{m}}} F^i(r^i, t_2, \dots, t_{\bar{m}}) \prod_{l \in \bar{N}(i), l \neq i} p_{l t_l}$. 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, 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 11. *A corner $\underline{p} = \left(\underline{e}^{\mathrm{T}}_{t_1}, \underline{e}^{\mathrm{T}}_{t_2}, \dots, \underline{e}^{\mathrm{T}}_{t_n}\right)^{\mathrm{T}}$ is fully compatible if and only if $F^i(t_1, t_2, \dots, t_{\bar{m}}) > F^i(r, t_2, \dots, t_{\bar{m}})$ for all $r \neq t_i$.*

Proof. The proof is trivial from Theorem 10. ■

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} $(\cdots < T_{t_i} < T_{t_i+1} < \cdots)$, 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 $k \in \left[T_{t_i}, T_{t_i+1}\right)$. 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 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.

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

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 $\underline{B} = (b_1, b_2, \dots, b_n)^{\mathrm{T}}$ to denote the learning parameters of LAs. Let $\underline{\Theta}$ be 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 ρ_i $(i = 1, \dots, n)$ is time invariant through the operation of ACLA. Define $\Delta \underline{p}(k) = \mathrm{E}[\underline{p}(k+1) - \underline{p}(k) | \underline{p}(k) = \underline{p}] = \underline{\Lambda \Theta} \underline{g}(\underline{p}(k), \underline{\beta}(k))$. 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

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

Using (1), the components of $\underline{p}(k)$ can be obtained as

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

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

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

The interpolated process $\{\underline{\tilde{p}}^{B\rho}(t)\}_{t \geq 0}$ is a sequence of random variables that takes values from $R^{m_1 \times \cdots \times m_n}$, where $R^{m_1 \times \cdots \times m_n}$ 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 $R^{m_1 \times \cdots \times m_n}$. The objective is to study the limit of the sequence $\{\underline{\tilde{p}}^{B\rho}(t)\}_{t \geq 0}$ as $\max\{b_1, \ldots, b_n\} \to 0$. Since $0 < \rho_i \leq 1$, when $\max\{b_1, \ldots, b_n\}$ 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 Eq. (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.

$$\frac{\mathrm{d} p_{iy}}{\mathrm{d} t} = p_{iy} \left[d_{iy}(\underline{p}) - D_i(\underline{p})\right]. \tag{11}$$

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. First, we solve ODE (10) and then, we characterize its solution. 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 $\max\{b_1, \ldots, b_n\}$ 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 12. *Sequence $\{\underline{\tilde{p}}^{B\rho}(.)\}$ converges weakly to the solution of*

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

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

Proof. Using Algorithm (5), we have : (a) $\{\underline{p}(k), (\underline{\alpha}(k-1), \underline{\beta}(k-1))\}_{k \geq 0}$ is a Markov process. (b) $(\underline{\alpha}(k), \underline{\beta}(k))$ takes values in a compact metric space. (c) $\underline{g}$ is bounded, continuous and independent of $\underline{B}$. (d) ODE (12) has a unique solution for each initial condition $\underline{X}(0)$. (e) 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 (Kushner, 1984, Theorem 3.2), sequence $\{\underline{\tilde{p}}^{B\rho}(.)\}$ converges weakly, as $\max\{b_1, \ldots, b_n\} \to 0$ to the solution of $\frac{\mathrm{d}\underline{X}}{\mathrm{d}t} = \underline{\bar{f}}(\underline{X})$, $X(0) = X_0$, where $\underline{\bar{f}}(\underline{p}(k)) = \mathrm{E}_p \underline{f}(\underline{p}(k), \underline{\alpha}(k), \underline{\beta}(k))$ and E_p denotes the expectation with respect to the invariant measure $M^{\bar{p}}$. Since $\underline{p}(k) = \underline{\hat{p}}$, $(\underline{\alpha}(k), \underline{\beta}(k))$ 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 12 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 $\max\{b_1, \ldots, b_n\} \to 0$. As the length of the time interval increases and $\max\{b_1, \ldots, b_n\} \to 0$, 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 $\max\{b_1, \ldots, b_n\} \to 0$. Although the speed of convergence depends on the specific value of $\underline{B}$. The above point is summarized in Lemma 13.

Lemma 13. *For large k and small enough value of $\max\{b_1, \ldots, b_n\}$, 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 12 says that $\mathrm{E}\left[h_T(\underline{\tilde{p}})^{B\rho}\right] \to \mathrm{E}\left[h_T(\underline{X})\right] = 0$ as $\max\{b_1, \ldots, b_n\} \to 0$.

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

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

Since $\underline{p}^o$ is an asymptotically stable equilibrium point of ODE (10), then for all initial configurations in a 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 Eq. (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 14. *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 $\underline{p}_i$ cannot be zero at the same time. 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 $d_{iq}(\underline{p}) - d_{i1}(\underline{p}) = \cdots = d_{iq}(\underline{p}) - d_{im_i}(\underline{p}) = d_0$. 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, Eq. (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 15. *All compatible configurations are equilibrium points of $\underline{f}(.)$.*

Proof. Let $\underline{p}$ be a compatible configuration. Then from Lemma 8, $\underline{d}_{ir}(\underline{p}) = D_i(\underline{p})$ 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 Lemmas 14 and 15, all equilibrium points of (10) are known. In order to study the stability of 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 16. *A corner $\underline{p}^o \in K^*$ is fully compatible if and only if it is uniformly asymptotically stable.*

Proof. Let configuration $\underline{p}^o = \left(\underline{e}_{t_1}^{\mathrm{T}}, \ldots, \underline{e}_{t_n}^{\mathrm{T}}\right)^{\mathrm{T}}$ be a fully compatible corner of K. 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

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

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 10, we have $F^i(y, t_2, \ldots, t_{\tilde{m}}) - F^i(t_i, t_2, \ldots, t_{\tilde{m}}) < 0$ for all i. Thus, Eq. (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 asymptotically stable. Conversely, assume that $\underline{p}^o$ is uniformly asymptotically 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 $\tilde{f}_{iy} = F^i(y, t_2, \ldots, t_{\tilde{m}}) - F^i(t_i, t_2, \ldots, t_{\tilde{m}})$ for all i. Since $\underline{p}^o$ is uniformly asymptotically stable, A should have eigenvalues with negative real parts and hence $\tilde{f}_{iy} < 0$. Using Theorem 10, this implies that $\underline{p}^o$ is fully compatible. ∎

Lemma 17. *Incompatible equilibrium points of $\underline{f}(.)$ are unstable.*

Proof. Let $\underline{p}^o$ be an equilibrium point of $f(.)$ which is not compatible. Then by Lemma 7, 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 a small open neighborhood around $\underline{p}^o$. Using (11), it is implied that for all points in this neighborhood $\frac{dp_{jy}}{\mathrm{d}t} > 0$ if $p_{jy} \neq 0$. Hence, no matter how small this neighborhood we take, there will be infinitely many points starting from which, $\underline{p}(k)$ will eventually leave that neighborhood, which implies that $\underline{p}^o$ is unstable. ■

3.3. Convergence results

We study the convergence of ACLA for the following different initial configurations, which covers all points in K: (a) $\underline{p}(0)$ is close to a compatible corner $\underline{p}^o$. By Lemma 16, 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. (b) $\underline{p}(0)$ is close to an incompatible corner $\underline{p}^o$. By Lemma 17, no matter how small a 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 (d) below. (c) $\underline{p}(0) \in K^*$. Using the convergence properties of L_{R-I} learning algorithm (Narendra & Thathachar, 1989), no matter whether $\underline{p}(0)$ is compatible or not, the ACLA will be absorbed to $\underline{p}(0)$. (d) $\underline{p}(0) \in K - K^*$. The convergence results of the ACLA for these initial configurations is stated in Theorem 18.

Theorem 18. *Suppose there is a bounded differential function $D : R^{m_1+\cdots+m_n} \to R$ such that for some constant $c > 0$, $\frac{\partial D}{\partial p_{ir}}(\underline{p}) = c d_{ir}(\underline{p})$ for all i and r, and $\rho_i > 0$. 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

$$\frac{\mathrm{d}D}{\mathrm{d}t} = \sum_i \sum_y \frac{\partial D}{\partial p_{iy}} \frac{\partial p_{iy}}{\partial t}$$
$$= c \sum_i \sum_y \sum_{r>y} p_{iy} p_{ir} \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p})\right]^2 \geq 0. \quad (16)$$

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 $\frac{\mathrm{d}D}{\mathrm{d}t} = 0$. 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 14 and 15, 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 incompatible equilibrium configurations are unstable, the theorem follows. ■

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 19. Rule F^i is called commutative if and only if we have

$$F^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \alpha_{i+\bar{x}_3}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) = F^i(\alpha_{i+\bar{x}_{\bar{m}}}, \alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}-1}})$$
$$= \cdots = F^i(\alpha_{i+\bar{x}_2}, \alpha_{i+\bar{x}_3}, \alpha_{i+\bar{x}_4}, \ldots, \alpha_{i+\bar{x}_1}).$$

To simplify the algebraic manipulations, we study linear ACLA with neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. Theorem 20 is an additional property for compatible configurations using commutative rules.

Theorem 20. *For an 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 $D(\underline{p})$ does not increase as one moves away from $\underline{p}$. Thus we have $\left.\frac{\mathrm{d}D(\gamma \underline{q} + (1-\gamma)\underline{p})}{\mathrm{d}\gamma}\right|_{\gamma=0} \leq 0$. 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 is 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 9, implying compatibility of $\underline{p}$. ■

Now, using Section 3, we can state the main theorem for the convergence of ACLA using commutative rules.

Theorem 21. *An ACLA, which uses uniform and commutative rules, 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 the 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 18 the convergence can be concluded. ■

5. Computer experiments

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

5.1. Numerical examples

In this section, we study the patterns formed by the evolution of ACLA from a random initial configurations. Each cell has a LA with action-set $\{0, 1, \ldots, m-1\}$. Hence, the configuration

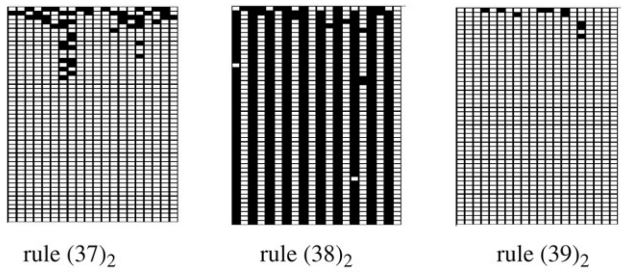

Fig. 1. Time–space diagram for asynchronous CLA.

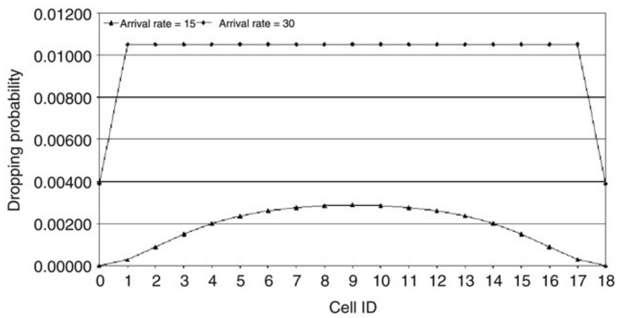

Fig. 2. The dropping probability of handoff calls for dynamic guard channel algorithm.

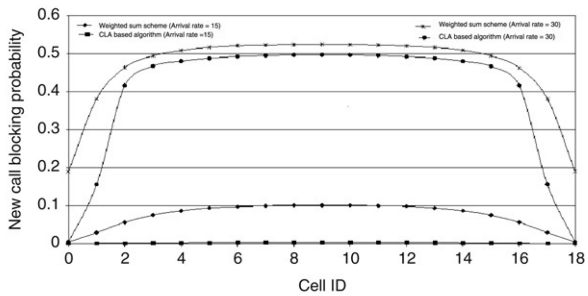

Fig. 3. The blocking probability of new calls for dynamic guard channel algorithm and weighted sum scheme.

of each cell and its neighbors forms a $\bar{m}$ digits number in interval $\left[0, m^{\bar{m}} - 1\right]$. 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, we use the 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 (Beigy & Meybodi, 2004). Fig. 1 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 ACLA uses the neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. The probability of activation of LA is the following fixed vector $\underline{\rho} = (0.1, 0.1, 0.2, 0.1, 0.3, 0.5, 0.6, 0.8, 0.1, 0.1, 0.2, 0.1, 0.3, 0.5, 0.6, 0.8, 0.1, 0.1, 0.2, 0.1)^{\mathrm{T}}$. The simulation results show that the ACLA converges to a configuration in K^* rather than to a configuration in $K - K^*$.

5.2. *Call admission in mobile cellular networks*

In cellular networks, call admission algorithms control *blocking probability of new calls* (B_n) and *dropping probability of handoff calls* (B_h). Since the dropping of handoff calls is more important than the blocking of new calls, these algorithms usually put restriction on the acceptance of new calls. *Guard channel algorithm* is a call admission algorithm that reserves a subset of the channels allocated to the cell, called *guard channels*, for sole use of the handoff calls. This algorithm accepts handoff calls as long as the channels are available and rejects new calls as long as the channel occupancy exceeds a certain threshold until the channel occupancy goes below that threshold. This algorithm decreases the B_h, but the B_n may be degraded to a great extent. This algorithm 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 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 network into an ACLA with n cells and a LA with m actions using L_{R-I} algorithm is assigned to each cell. The LA associated to a cell will be responsible for adaptation of the number of guard channels for that cell in the network. The neighbors of a cell are those cells which affect the rate of handovers to that cell. The local rule for ACLA is given by

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

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 if 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. If the cell has at least g_u free channels, then the call will be accepted; otherwise it will be blocked. Finally, the local rule is computed according to (17) and then the action probability vector of LA A_u will be updated accordingly. In what follows, we compare the performance of the weighted sum scheme (Peha & Sutivong, 2001) and the proposed algorithm. The simulation is conducted on the homogeneous linear 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 λ and the duration of calls and their dwell time 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 λ. The results of these simulations are shown in Figs. 2 and 3. 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.

6. Conclusions

In this paper, ACLA was proposed and shown that for a class of rules, called commutative rules, it converges to a configuration for which its average rewards are maximized. To further study the behavior of ACLA, we conducted an experiment in which the patterns formed by the evolution of ACLA from a random initial configuration was studied. The results obtained from experiments confirm the theory. Finally, an application of ACLA to call admission control in cellular networks was presented. An algorithm based on ACLA for call admission control in 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.

References

Beigy, H., & Meybodi, M. R. (2003). *Springer-verlag lecture notes in computer science*: *vol. 2690. A self-organizing channel assignment algorithm: A Cellular learning automata approach* (pp. 119–126). Springer-Verlag.

Beigy, H., & Meybodi, M. R. (2005). An adaptive call admission algorithm for cellular networks. *Electerical and Computer Engineering*, *31*, 132–151.

Kushner, H. J. (1984). *Approximation and weak convergence methods for random processes*. Cambridge: MIT Press.

Meybodi, M. R., & Beigy, H. (2002). A note on learning automata based schemes for adaptation of BP parameters. *Journal of Neurocomputing*, *48*, 957–974.

Beigy, H., & Meybodi, M. R. (2004). A mathematical framework for cellular learning automata. *Journal of Advances in Complex Systems*, *7*, 295–320.

Meybodi, M. R., Beigy, H., & Taherkhani, M. (2003). Cellular learning automata and its applications. *Sharif Journal of Science and Technology*, *19*(25), 54–77.

Narendra, K. S., & Thathachar, K. S. (1989). *Learning automata: An introduction*. New York: Printice-Hall.

Oommen, B. J., & de St Croix, E. V. (1996). Graph partitioning using learning automata. *IEEE Transactions on Computers*, *45*, 195–208.

Oommen, B. J., & Roberts, T. D. (2000). Continuous learning automata solutions to the capacity assignment problem. *IEEE Transactions on Computers*, *49*, 608–620.

Peha, J. M., & Sutivong, A. (2001). Admission control algorithms for cellular systems. *Wireless Networks*, *7*, 117–125.

Schonfisch, B., & Ross, A. D. (1999). Synchronous and asynchronous updating in cellular automata. *Journal of Biosystem*, *51*, 123–143.

Hamid Beigy received the B.S. and M.S. degrees in Computer Engineering from the Shiraz University in Iran, in 1992 and 1995, respectively. He also received the Ph.D. degree in Computer Engineering from the Amirkabir University of Technology in Iran, in 2004. Currently, he is an Assistant Professor in Computer Engineering Department at the Sharif University of Technology, Tehran, Iran. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, and soft computing.

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 from 1985 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.