ITRC

10th Annual
Conference of Computer Society of Iran
February 15-17, 2005

انجمن کامپیوتر ایران
Computer Society of Iran

Asynchronous Cellular Learning Automata

Hamid Beigy
Computer Engineering Department, Sharif University of Technology
Institute for Studies in Theoretical Physics and Mathematics (IPM) School of Computer Science
M. R. Meybodi
Soft Computing Laboratory, Computer Engineering Department, Amirkabir University of Technology
Institute for Studies in Theoretical Physics and Mathematics (IPM) School of Computer Science

Abstract— **In this paper, we introduce asynchronous cellular learning automata and then study its convergence behavior. It is shown that for class of rules called commutative rules, the asynchronous cellular learning automata converges to a stable and compatible configuration. The numerical results also confirms the theory.**

I. Introduction

CELLULAR automata are a set of identical components, called *cells*, each of which occupies a node of a regular, discrete, infinite spatial network [1]. The simple components act together to produce complicated patterns of behavior. Cellular automata perform complex computation with high degree of efficiency and robustness. They are specially suitable for modelings natural systems that can be described as massive collections of simple objects interacting locally with each other [2], [3]. Each cell can assume a state from a finite set, the cellular automata evolves in discrete steps, changing the states of all its cells according to a local rule, homogenously applied at each step. The new state of a cell depends on the previous states of a set of cells, which can include the cell itself, and constitutes its neighborhood [4].

In other hand, learning automata are simple agents for doing simple things. The learning automata have finite set of actions and each stage choose one of them. The choice of an action depends on the state of automaton which is usually represented by an action probability vector. For each action chosen by the automaton, the environment gives a reinforcement signal with fixed unknown probability distribution, which specified the goodness of the applied action. Then upon receiving the reinforcement signal, the learning automaton updates its action probability vector by employing a learning algorithm. The learning algorithm is a recurrence relation and is used to modify action probability vector $\underline{p}$. Various learning algorithms have been reported in the literature. Below, a learning algorithm, called L_{R-I}, for updating the action probability vector, which will be used later in this paper, is given. Let α_i be the action chosen at time k as a sample realization from probability distribution $p(k)$. In L_{R-I} algorithm, the action probability vector is updated according the following rule.

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

when $\beta(k) = 0$ i.e. environment rewards the chosen action of learning automaton and the action probability vector remains unchanged when $\beta(k) = 1$, i.e. environment penalizes the chosen action of learning automaton. Parameters $0 < a < 1$ represent *step lengths* and r is the number of actions for LA [5]. LA have been used successfully in many applications such as telephone and data network routing [6], solving NP-Complete problems [7], capacity assignment [8] and neural network engineering [9], [10] to mention a few.

Learning automata are, by design, "simple agents for doing simple things". The full potential of a LA is realized when multiple automata 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 [11], the 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 learning automata (LA) to adjust the state transition probability of stochastic CA.

Cellular learning automata (CLA) is a mathematical model for dynamical complex systems that consists of large number of simple components [11]. 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 a nonstationary environment because of the fact that it changes as action probability vectors of neighboring LAs vary.

The operation of cellular learning automata 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 vectors of the learning automata 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 cellular automata determines the reinforcement signal to each learning automaton residing in that cell. Finally, each learning automaton 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. Formally a $d-$dimensional CLA is given below.

Definition 1: A $d-$dimensional cellular learning automata is a structure $\mathcal{A} = (Z^d, \Phi, A, N, \mathcal{F})$, 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 a finite subset of Z^d called neighborhood vector, where $\bar{x}_i \in Z^d$. The neighborhood vector determines the relative position of the neighboring lattice cells from any given cell u in the lattice Z^d. The neighbors of a particular cell u are set of cells $\{u + \bar{x}_i | i = 1, 2, \cdots, \bar{m}\}$. We assume that, there exists a neighborhood function $\bar{N}(u)$ that maps a cell u to the set of its neighbors, that is

$$\bar{N}(u) = (u + \bar{x}_1, u + \bar{x}_2, \cdots, u + \bar{x}_{\bar{m}}). \qquad (2)$$

For the sake of simplicity, we assume that the first element of neighborhood vector (i.e. $\bar{x}_1$) is equal to d-tuple $(0, 0, \cdots, 0)$ or equivalently $u + \bar{x}_1 = u$.

5. $\mathcal{F} : \underline{\Phi}^{\bar{m}} \to \underline{\beta}$ is the local rule of the cellular automata, where $\underline{\beta}$ is the set of values that the reinforcement signal can take. It gives the reward (reinforcement) signal to each LA from the current actions selected by its neighboring LAs.

A number of applications for CLA have been developed recently such as rumor diffusion [12], image processing [13], [14], [15], and fixed channel assignment in cellular networks [16]. In [17], a mathematical methodology to study the behavior of the synchronous CLA is given and its convergence 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 [18] and dynamic channel assignment in cellular networks [16], a type of cellular learning automata in which the LA in different cells are activated asynchronously is needed. We call such a CLA as *asynchronous CLA*. In this paper, we introduce asynchronous CLA and study its convergence behavior. It is shown that for commutative rules, the asynchronous CLA converges to a globally stable state.

The rest of this paper is organized as follows. In section II, the asynchronous CLA is presented. Section III presents the convergence behavior of asynchronous CLA. In section IV, the behavior of the ACLA when the commutative rules are used is studied. Sections V presents the numerical example and section VI concludes the paper.

II. 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 learning automata may be activated in either *time-driven* or *step-driven* manner. In time-driven asynchronous CLA, each cell is assumed to have an internal clock which wakes up the learning automaton associated to that cell while in step-driven asynchronous CLA, a cell is selected in fixed or random sequence. The asynchronous CLA in which cells are selected randomly is of more interest to us because of its application to cellular mobile networks. Formally a $d-$dimensional asynchronous step–driven CLA is given below.

Definition 2: A $d-$dimensional asynchronous step–driven cellular learning automata is a structure $\mathcal{A} = (Z^d, \Phi, A, N, \mathcal{F}, \underline{\rho})$, 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. $\mathcal{F} : \underline{\Phi}^{\bar{m}} \rightarrow \underline{\beta}$ is the local rule of the cellular automata.
6. $\underline{\rho}$ is an $n \times n$ diagonal matrix and represents the step–driven vector, where ρ_{ii} is the activation probability of learning automata in cell i. When $\rho_{ii} = 1$, for $i = 1, \dots, n$, the CLA is synchronous.

In what follows, we consider ACLA with n cells and neighborhood function $\bar{N}(i)$. A learning automaton denoted by A_i, which has a finite action set $\underline{\alpha}_i$, is associated to cell i (for $i = 1, \dots, n$) of ACLA. Let cardinality of $\underline{\alpha}_i$ be m_i and the state of ACLA represented by $\underline{p} = (\underline{p}_1', \underline{p}_2', \dots, \underline{p}_n')'$, where $\underline{p}_i = (p_{i1}, \dots, p_{im_i})'$ is the action probability vector of A_i. It is evident that the local environment for each learning automaton is the learning automata residing in its neighboring cells. From the repeated application of simple local rules and simple learning algorithms, the global behavior of ACLA can be very complex.

The operation of ACLA takes place as the following iterations. At iteration k, each learning automaton A_i is activated with probability ρ_i and the activated learning automata choose one of their actions. The activated automata use their current actions to execute the rule (computing 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 and the reinforcement signal received by learning automaton A_i, respectively. The reinforcement signal is produced by the application of local rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2} \dots, \alpha_{i+\bar{x}_{\bar{m}}}) \rightarrow \underline{\beta}$. The higher value of β_i means that the chosen action of A_i is more rewarded. Finally, activated learning automata update their action probability vectors and the process repeats. Since each set $\underline{\alpha}_i$ is finite, rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \dots, \alpha_{i+\bar{x}_{\bar{m}}}) \rightarrow \underline{\beta}$ can be represented by a hyper matrix of dimensions $m_1 \times m_2 \times \dots \times m_{\bar{m}}$. These n hyper matrices together constitutes what we call the rule of ACLA. When all of these n hyper matrices are equal, the rule is uniform; otherwise the rule is nonuniform. For the sake of simplicity in our presentation, the rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2} \dots, \alpha_{i+\bar{x}_{\bar{m}}})$ is denoted by $\mathcal{F}^i(\alpha_1, \alpha_2 \dots, \alpha_{\bar{m}})$. Based on set $\underline{\beta}$, the ACLA can be classified into three groups: P-model, Q-model, and S-model Cellular learning automata. When $\underline{\beta} = \{0, 1\}$, we refer to ACLA as *P-model cellular learning automata*, when $\underline{\beta} = \{b_1, \dots, b_l\}$, (for $l < \infty$), we refer to ACLA as *Q-model cellular learning automata*, and when $\underline{\beta} = [b_1, b_2]$, we refer to ACLA as *S-model cellular learning automata.* If learning automaton A_i uses learning algorithm L_i, we denote ACLA by the $ACLA(L_1, \dots, L_n)$. If $L_i = L$ for all $i = 1, \cdots, n$, then we denote the ACLA by the $ACLA(L)$.

In the following subsections, we give some definitions and notations, which will be used later in section III to analysis the behavior of ACLA.

A. Definitions and Notations

In this subsection, we give some definitions and then derive some preliminary results regarding ACLA which will be used later in this paper for the analysis of ACLA.

Definition 3: A configuration of the ACLA at stage k is denoted by $\underline{p}(k) = (\underline{p}_1'(k), \underline{p}_2'(k), \dots, \underline{p}_n'(k))'$, where $\underline{p}_i(k)$ is the action probability vector of learning automaton A_i.

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

$$\begin{aligned} \mathcal{K}^* &= \{\underline{p} | \underline{p} = (\underline{p}_1', \underline{p}_2', \dots, \underline{p}_n')', \underline{p}_i = (p_{i1}, \dots, p_{im_i})' \\ &, \quad p_{iy} = 0 \text{ or } 1 \ \ \forall y, i, \sum_y p_{iy} = 1 \ \forall i\}, \text{ and} \\ \mathcal{K} &= \{\underline{p} | \underline{p} = (\underline{p}_1', \underline{p}_2', \dots, \underline{p}_n')', \underline{p}_i = (p_{i1}, \dots, p_{im_i})' \\ &, \quad 0 \le p_{iy} \le 1 \ \ \forall y, i, \sum_y p_{iy} = 1 \ \forall i\}, \text{ respectively.} \end{aligned}$$

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

Definition 5: The global behavior of a ACLA is a mapping $\mathcal{G} : \mathcal{K} \rightarrow \mathcal{K}$ that describes the dynamics of the ACLA.

Definition 6: The evolution of the ACLA from a given initial configuration $\underline{p}(0) \in \mathcal{K}$ is a sequence of configurations $\{\underline{p}(k)\}_{k \ge 0}$, such that $\underline{p}(k+1) = \mathcal{G}(\underline{p}(k))$.

Definition 7: The average reward for action r of automaton A_i for configuration $\underline{p} \in \mathcal{K}$ is defined as

$$d_{ir}(\underline{p}) = \sum_{y_2} \dots \sum_{y_{\bar{m}}} \mathcal{F}^i(r, y_2, \dots, y_{\bar{m}}) \prod_{\substack{l \in \bar{N}(i) \\ l \ne i}} p_{l y_l}, \tag{3}$$

and the average reward for learning automaton A_i

is defined a

$$D_i(\underline{p}) = \sum_r d_{ir}(\underline{p})p_{ir}. \quad (4)$$

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

Definition 8: A configuration $\underline{p} \in \mathcal{K}$ is compatible if

$$\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir} \quad (5)$$

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

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

Definition 9: The total average reward for the ACLA at configuration $\underline{p} \in \mathcal{K}$ is the sum of the average rewards for all the learning automata in the ACLA, that is,

$$\mathcal{D}(\underline{p}) = \sum_i D_i(\underline{p}). \quad (6)$$

Lemma 1: The ACLA has at least one compatible configuration.

Proof: Let $\psi_{ir}(\underline{p}) = d_{ir}(\underline{p}) - D_i(\underline{p})$ and $\phi_{ir}(\underline{p}) = \max\{\psi_{ir}(\underline{p}), 0\}$ for $i = 1, \ldots, n$ and $r = 1, \ldots, m_i$. Note that $\psi_{ir}(\underline{p})$ and $\phi_{ir}(\underline{p})$ are continuous functions on $\mathcal{K}$. Introducing the mapping $T : \mathcal{K} \to \mathcal{K}$ given by

$$\bar{p}_{ir} = \frac{p_{ir} + \phi_{ir}}{1 + \sum_{j=1}^{m} \phi_{ij}} \quad (7)$$

for $i = 1, \ldots, n$ and $r = 1, \ldots, m_i$. It is evident that T is a continuous mapping. Since $\mathcal{K}$ is closed, bounded and convex, we can use the *Brouwer's fixed point theorem* to show that every mapping T has at least one fixed point. We now show that every fixed point of T is necessarily a compatible configuration of the ACLA and conversely every compatible configuration of the ACLA is a fixed point of T, that is, $\underline{p} = T(\underline{p})$ thereby concluding the proof of the lemma. We first verify the latter assertion: if $\underline{p} \in \mathcal{K}$ is a compatible configuration, then for every $\underline{q} \in \mathcal{K}$, we have $\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir}$ for all $i = 1, \ldots, n$. Configuration $\underline{q}$ also includes $\underline{q} = (\underline{p}_1', \ldots, \underline{e}_{r_i}, \ldots, \underline{p}_n')'$ for fixed i $(i = 1, \ldots, n)$. Since $d_{ir_i}(\underline{p})$ is independent of $\underline{p}_i$, we obtain $\psi_{ir_i}(\underline{p}) \leq 0$. Hence, $\phi_{ir_i} = 0$ for all $i = 1, \ldots, n$ and $r_i = 1, \ldots, m$ and we have $\underline{p} = T(\underline{p})$, which concluding that $\underline{p}$ is a fixed point of T.

Conversely, suppose that $\underline{p} \in \mathcal{K}$ is a fixed point of T, but not a compatible configuration. Then for some i $(1 \leq i \leq n)$, there exists an action probability vector $\underline{\tilde{p}}_i$ such that $\underline{\tilde{p}} = (\underline{p}_1', \ldots, \underline{\tilde{p}}_i', \ldots, \underline{p}_n')'$ and

$$\sum_r d_{ir}(\underline{p})p_{ir} < \sum_r d_{ir}(\underline{p})\tilde{p}_{ir} \quad (8)$$

Let y_i $(1 \leq i \leq m_i)$ be an action for which $d_{ir}(\underline{p})$ attains its maximum value. Then $D_i(\underline{\tilde{p}})$ can be bounded from above by $d_{iy_i}(\underline{p})$, thus implies $\psi_{ir_i}(\underline{p}) > 0$, which implies $\phi_{ir_i}(\underline{p}) > 0$. But since $\phi_{ir_i}(\underline{p})$ is nonnegative for all r_i, then $\sum_j \phi_{ij}(\underline{p}) > 0$. Let r_i $(1 \leq i \leq m_i)$ be an action for which $d_{ir}(\underline{p})$ attains its minimum value. Then by using inequality (8), it can be shown that $D_i(\underline{p})$ bounded below by $d_{ir_i}(\underline{p})$. This implies $\psi_{iy_i}(\underline{p}) < 0$, which implies $\phi_{iy_i}(\underline{p}) = 0$, which when used in (7) yield the conclusion $\bar{p}_{iy_i} < \tilde{p}_{iy_i}$, because $\sum_j \phi_{ij}(\underline{p}) > 0$ and contradicts the hypothesis that $\underline{q}$ is a fixed point of T. ■

Lemma 2: Configuration $\underline{p} \in \mathcal{K}$ is compatible if and only if

$$d_{ir}(\underline{p}) \leq D_i(\underline{p}),$$

for all i and r.

Proof: If $\underline{p} \in \mathcal{K}$ is a compatible configuration, then from (5), for every $\underline{q} \in \mathcal{K}$ and $1 \leq i \leq n$, we have $\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir}$. Since, $\underline{q}$ includes $\underline{q} = (\underline{p}_1', \ldots, \underline{e}_{r_i}, \ldots, \underline{p}_n')'$ for fixed i $(i = 1, \ldots, n)$ and $d_{ir_i}(\underline{p})$ is independent of $\underline{p}_i$, then we obtain $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$.

Conversely, suppose that $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$ $(i = 1, \ldots, n$ and $r_i = 1, \ldots, m)$ but $\underline{p}$ is not compatible. Then for some learning automaton i with action probability vector $\underline{q}_i$ there exists an action y_i such that $\underline{q} = (\underline{p}_1', \ldots, \underline{q}_i, \ldots, \underline{p}_n')'$ and $d_{iy_i}(\underline{p}) > D_i(\underline{q})$. Action y_i denotes the action for which $d_{ir_i}(\underline{p})$ attains its maximum value. Since $\underline{q}_i$ is a probability vector, then $D_i(\underline{q})$ is bounded from the above with $d_{iy_i}(\underline{p})$ and arrives at strict inequality $D_i(\underline{p}) < D_i(\underline{q}) < d_{iy_i}(\underline{p})$. But this contradicts the hypothesis that $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$, which concludes that $\underline{p}$ is a compatible configuration. ■

Lemma 3: Let $\underline{p} \in \mathcal{K}$ be a compatible configuration. Then for each i, we have

$$d_{ir}(\underline{p}) = D_i(\underline{p}),$$

for all r such that $p_{ir} > 0$.

Proof: From lemma 2, we have

$$d_{ir}(\underline{p}) \leq D_i(\underline{p}),$$

for all i and r. Suppose that for at least one action y of automaton A_j, the above inequality is strict. Thus we have

$$d_{jy}(\underline{p}) < D_j(\underline{p}).$$

From the above inequality and equation (4), we obtain

$$D_i(\underline{p}) = \sum_{\substack{r=1\\p_{ir}>0}}^{m_i} d_{ir}(\underline{p})p_{ir} < D_i(\underline{p}) \sum_{\substack{r=1\\p_{ir}>0}}^{m_i} p_{ir} = D_i(\underline{p}).$$

The above contradiction completes the proof of the lemma. ■

Theorem 1: A configuration $\underline{p} \in \mathcal{K}$ is compatible if and only if $\sum_i \sum_y d_{iy}(\underline{p}) [p_{iy} - q_{iy}] \geq 0$ holds for all $\underline{q} \in \mathcal{K}$.

Proof: If $\underline{p}$ is compatible, then from (5), we have

$$\sum_y d_{iy}(\underline{p})p_{iy} \geq \sum_y d_{iy}(\underline{p})q_{iy},$$

for any $\underline{q} \in \mathcal{K}$. Summing over i we obtain

$$\sum_i \sum_y d_{iy}(\underline{p})p_{iy} \geq \sum_i \sum_y d_{iy}(\underline{p})q_{iy}.$$

Conversely, if inequality (5) is solved by $\underline{p}$, then for any $\underline{q} \in \mathcal{K}$, fixed l, $1 \leq l \leq n$, and set $\underline{q} = (\underline{p}'_1, \ldots, \underline{q}'_l, \ldots, \underline{p}'_n)'$. Then $\underline{q} \in \mathcal{K}$ and so

$$\begin{aligned}\sum_i \sum_y d_{iy}(\underline{p}) [p_{iy} - q_{iy}] &= \sum_y d_{ly}(\underline{p}) [p_{ly} - q_{ly}] \\ &\geq 0.\end{aligned}$$

Since l is arbitrary, then the above inequality implies that $\underline{p}$ is compatible. ■

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

Theorem 2: A corner $\underline{p} = (\underline{e}_{t_1}, \underline{e}_{t_2}, \ldots, \underline{e}_{t_n})'$ is compatible if and only if

$$\mathcal{F}^i(t_1, t_2, \ldots, t_{\bar{m}}) \geq \mathcal{F}^i(r, t_2, \ldots, t_{\bar{m}})$$

for all $r \neq t_i$.

Proof: Let $\underline{q} = (\underline{e}_{t_1}, \underline{e}_{t_2}, \ldots, \underline{e}_{r_i}, \ldots, \underline{e}_{t_n})'$ for $r_i \neq t_i$ be a compatible corner. From definition 8, we have

$$\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir}. \tag{9}$$

Since $\underline{p}$ and $\underline{q}$ are two corners, then the above inequality can be simplified as

$$d_{it_i}(\underline{p}) \geq d_{ir_i}(\underline{p}). \tag{10}$$

Substituting $d_{ir}(\underline{p})$ from equation (4), we obtain

$$\mathcal{F}^i(t_1, t_2, \ldots, t_{\bar{m}}) \geq \mathcal{F}^i(r_1, t_2, \ldots, t_{\bar{m}}).$$

Conversely, assume that $\mathcal{F}^i(t_1, t_2, \ldots, t_{\bar{m}}) \geq \mathcal{F}^i(r_1, t_2, \ldots, t_{\bar{m}})$ but $\underline{p}$ is not compatible. From definition 8 and by some algebraic simplifications we obtain

$$\sum_r \left[\mathcal{F}^i(t_1, t_2, \ldots, t_{\bar{m}} - \mathcal{F}^i(r_1, t_2, \ldots, t_{\bar{m}})\right] q_{ir} \geq 0.$$

Since each term of the above inequality is nonnegative, thus the summation is also nonnegative, which contradicts our assumption and hence $\underline{p}$ is compatible. ■

III. Behavior of Asynchronous Cellular Learning Automata

In this section, we analyze the asynchronous CLA in which all learning automata use the L_{R-I} learning algorithm. We denote this ACLA by ACLA(L_{R-I}). For analyzing the behavior of ACLA, we need to count how many times a learning automaton is activated. To do this, we introduce the concept of *local time* for every learning automaton (cell). The local time for every automaton starts with 1 and incremented by 1 every time that learning automaton is activated. Let t_i be the local time for learning automaton 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 learning automaton A_i activated at local time t_i. Let $\alpha_i(T_{t_i})$ be the action chosen by learning automaton A_i at global time instant T_{t_i} in t_i^{th} activation of A_i. We assume that the action of a learning automaton is the action chosen in its last activation and is held be constant between its two activations, 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 $M \times M$ diagonal binary matrix with $\pi_{jj} = 1$ for $\sum_{l=1}^{i-1} m_l < j \leq \sum_{l=1}^{i} m_l$ if the learning automaton A_i is activated at each time instant and $\pi_{jj} = 0$ elsewhere. Using L_{R-I} learning algorithm, process $\{\underline{p}(k)\}_{k\geq 0}$ is Markovian and can be described by the following difference equation.

$$\underline{p}(k+1) = \underline{p}(k) + \underline{a}\underline{\Pi}\underline{g}(\underline{p}(k), \underline{\beta}(k)), \tag{11}$$

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, $\underline{a}$ is a $M \times M$ diagonal matrix with $a_{jj} = a_i$ for $\sum_{l=1}^{i-1} m_l < i \leq \sum_{l=1}^{i} m_l$, and a_i represents the learning parameter for learning automaton A_i. Let ρ_i $(0 < \rho_i \leq 1)$ be the probability that the learning automaton A_i activated to choose its action and let $\underline{\rho}$ is an $M \times M$ diagonal matrix with $\rho_{jj} = \rho_i$ for $\sum_{l=1}^{i-1} m_l < j \leq \sum_{l=1}^{i} m_l$. Let

$$\underline{\rho} = \mathrm{E}\left[\underline{\Pi}\right]. \tag{12}$$

We assume that ρ_i $(i = 1, \ldots, n)$ be time invariant through the operation of ACLA. Now define

$$\begin{aligned} \Delta \underline{p}(k) &= \mathrm{E}[\underline{p}(k+1) - \underline{p}(k) | \underline{p}(k = \underline{p}] \\ &= \underline{a}\rho g(\underline{p}(k), \underline{\beta}(k)). \end{aligned}$$

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

$$\Delta \underline{p}(k) = \underline{a}\rho \underline{f}\left(\underline{p}(k)\right). \tag{13}$$

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

$$\Delta p_{iy}(k) \quad = \quad a_i \rho_{ii} f_{iy}(\underline{p}), \tag{14}$$

where

$$f_{iy}(\underline{p}) \quad = \quad p_{iy}(k)\left[d_{iy}(\underline{p}) - D_i(\underline{p})\right]. \tag{15}$$

The details of the above derivation can be found in [19]. For different values of $\underline{a}$ and $\underline{\rho}$, equation (11) generates different process and we shall use $\underline{p}^{a\rho}(k)$ to denote this process whenever the value of $\underline{a}$ and $\underline{\rho}$ are to be specified explicitly. Define a continuous-time interpolation of $\underline{p}_i(k)$, denoted by $\tilde{\underline{p}}_i^{a\rho}(t)$, and called *interpolated process*, whose components are defined by

$$\tilde{\underline{p}}_i^{a\rho}(t) = \underline{p}_i(k) \qquad t \in [ka_i\rho_i, (k+1)a_i\rho_i], \tag{16}$$

where a_i is the learning parameter of the L_{R-I} algorithm and ρ_i is the activation probability for learning automaton A_i. The interpolated process $\{\tilde{\underline{p}}^a(t)\}_{t\geq 0}$ is a sequence of random variables that takes values from $\mathcal{R}^{m_1\times\cdots\times m_n}$, where $\mathcal{R}^{m_1\times\cdots\times m_n}$ 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 $\mathcal{K}$, which is a bounded subset of $\mathcal{R}^{m_1\times\cdots\times m_n}$. The objective is to study the limit of sequence $\{\tilde{\underline{p}}^a(t)\}_{t\geq 0}$ as $\max\{\underline{a}\} \to 0$. Since $0 < \rho_i \leq 1$, when $\max\{\underline{a}\}$ is sufficiently small, then $\{\tilde{\underline{p}}^a(t)\}_{t\geq 0}$ will be a good approximation to the asymptotic behavior of (16) and equation (13) can be written as the following ordinary differential equation (ODE).

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

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

$$\frac{dp_{iy}}{dt} = p_{iy}\left[d_{iy}(\underline{p}) - D_i(\underline{p})\right] \tag{18}$$

We are interested in characterizing the long term behavior of $\underline{p}(k)$ and hence the asymptotic behavior of ODE (17). The analysis of process $\{\underline{p}(k)\}_{k\geq 0}$ is done in two stages. In the first stage, we solve ODE (17) and in the second stage, we characterize the solution of this ODE. The solution of ODE (17) 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 $\tilde{\underline{p}}^a$ as $\max\{\underline{a}\}$ is sufficiently small. We show that the sequence of interpolated process $\{\tilde{\underline{p}}^a(t)\}$ converges weakly to the solution of ODE (17) with initial configuration $\underline{p}(0)$. This implies that asymptotic behavior of $\underline{p}(k)$ can be obtained from the solution of ODE (17).

Theorem 3: Sequence $\{\tilde{\underline{p}}^a(.)\}$ converges weakly to the solution of

$$\frac{d\underline{X}}{dt} = \underline{f}(\underline{X}) \tag{19}$$

with initial condition $\underline{X}(0) = X_0$ as $a \to 0$, where $X_0 = \tilde{\underline{p}}^a(0)$ and $a = \max\{\underline{a}\}$.

Proof: The following conditions are satisfied by the learning algorithm (11).

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 a.
4. ODE (19) has a unique solution for each initial condition $\underline{X}(0)$.
5. If $\underline{p}(k) = \bar{\underline{p}}$ 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 [20], sequence $\{\tilde{\underline{p}}^a(.)\}$ converges weakly, as $\max\{\underline{a}\} \to 0$ to the solution of

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

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

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

and hence the theorem. ■

Theorem 3 enables us to understand the long term behavior of $\underline{p}(k)$. The weak convergence in this theorem implies that path $\underline{p}^a(t)$ will closely follow the solution to the ODE on any finite interval with an arbitrarily high probability as $\max\{\underline{a}\} \to 0$. As the length of the time interval increases and $\max\{\underline{a}\} \to 0$, 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, $\underline{p}^a(.)$ 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 CLA follows the path of the ODE goes to infinity as $\max\{\underline{a}\} \to 0$. The speed of convergence depends on the specific value of $\underline{a}$. The above point is summarized in the following lemma.

Lemma 4: For large k and small enough value of $\max\{\underline{a}\}$, the asymptotic behavior of $\underline{p}(k)$ generated by the CLA can be well approximated by the solution to ODE (19) with the same initial configuration.

Proof: Let $\underline{X}(.)$ be the solution of ODE (19) with initial condition $\underline{X}(0) = X_0$ sufficiently close to an asymptotically stable configuration of the ODE, say $\underline{p}^o \in \mathcal{K}$. For any $\underline{Y}(t) \in \mathcal{K}$, $t \geq 0$ 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 $\mathcal{K}$. Then theorem 3 says that $\mathrm{E}\left[h_T(\tilde{\underline{p}})^a\right] \to \mathrm{E}\left[h_T(\underline{X})\right] = 0$ as $\max\{\underline{a}\} \to 0$. 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 (16), 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 a_0 such that

$$\text{Prob}\left[\sup_{k_1 \leq k \leq k_2} ||\underline{p}(k) - \underline{p}^o|| > \epsilon\right] = 0 \qquad \forall a < a_0,$$

where $a = \max\{\underline{a}\}$. Since $\underline{p}^o$ is an asymptotically stable equilibrium point of ODE (17), 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 (17), then study the stability property of equilibrium points of ODE (17), and finally state a main theorem about the convergence of the ACLA.

A. Equilibrium Points

The equilibrium points of equation (15) 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 $\underline{f}(\underline{p})$, which are studied in the following two lemmas.

Lemma 5: All the corners of $\mathcal{K}$ are equilibrium points of $\underline{f}(.)$. All the other equilibrium points $\underline{p}$ of $\underline{f}(.)$ satisfy

$$d_{iy}(\underline{p}) = d_{ir}(\underline{p}), \tag{20}$$

for all $r, y \in \{1, 2, \ldots, m_i\}$, and for all $i = 1, \ldots, n$.

Proof: From equation (14), it is obvious that $f_{iy} = 0$ (for $i = 1, 2, \ldots, n$) if $\underline{p}_i$ is a unit vector and hence all corners of $\mathcal{K}$ are equilibrium points of $\underline{f}(.)$. In order to find other equilibrium points of $\underline{f}(.)$, from (14) it is obvious that $f_{iy} = 0$ if $p_{iy} = 0$. Since $\underline{p}_i$ is a probability vector, then all components of $\underline{p}_i$ cannot be at the same time zero. Hence, when $p_{iy} \neq 0$, the following equation must hold.

$$\sum_{r \neq y} p_{ir}(k)\left[d_{iy}(\underline{p}) - d_{ir}(\underline{p})\right] = 0 \tag{21}$$

Using some algebraic simplification, we obtain

$$\sum_{r \neq q}\left[d_{iq}(\underline{p}) - d_{ir}(\underline{p})\right] p_{ir}(k) = d_{iq}(\underline{p}) - d_{iy}(\underline{p}) \tag{22}$$

for $y = 1, \ldots, m_i$ and $y \neq q$. The details of the above derivation can be found in [19]. The left hand side of the above equation is same, say as d_0, for all $y = 1, \ldots, m_i$ and $y \neq q$. Thus, for all $y \neq q$, we have

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

When $d_0 \neq 0$, equation (22) 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 the following

$$d_{iq}(\underline{p}) - d_{iy}(\underline{p}) = 0,$$

or equivalently

$$d_{iq}(\underline{p}) = d_{iy}(\underline{p}),$$

for $\forall i = 1, 2, \ldots, n$ and $\forall y \neq q$. When some p_{iy} are zero, for $\underline{f}$ to be zero, equation (21) must be satisfied for all $1 \leq y \leq m_i$ such that $p_{iy} \neq 0$ for each i, which completes the proof of this lemma. ■

Lemma 6: All compatible configurations are equilibrium points of $\underline{f}(.)$.

Proof: Let $\underline{p}$ be a compatible configuration. Then by lemma 3, for each i, either $p_{ir} = 0$ or $d_{ir}(\underline{p}) = D_i(\underline{p})$. Hence, $f_{ir}(\underline{p}) = 0$ for all i and r. ■

B. The Stability Property

In this subsection we characterize the stability of equilibrium configurations of ACLA, that is the equilibrium points of the ODE (17). From the lemmas 5 and 6, all the equilibrium points of (17) are known. In order to study the stability of the equilibrium points of (17), 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 the equilibrium points of ODE (17).

Lemma 7: A corner $\underline{p}^o \in \mathcal{K}^*$ is a fully compatible configuration if and only if it is uniformly asymptotically stable.

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

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

the origin is translated to $\underline{p}^0$. Since $\underline{p}_i$ $(1 \leq i \leq n)$ 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 $1 \leq i \leq n$) be the independent components. Using Taylor's expansion, f_{iy} can be expressed as

$$\begin{aligned} f_{iy} &= \tilde{p}_{iy}[\mathcal{F}^i(y, t_2, \dots, t_{\bar{m}}) - \mathcal{F}^i(t_i, t_2, \dots, t_{\bar{m}})] \\ &+ \text{ high order terms.} \end{aligned} \tag{23}$$

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 $V\left(\underline{\tilde{p}}\right) \geq 0$ 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 a fully compatible configuration, then from Theorem 2 we have $\mathcal{F}^i(y, t_2, \dots, t_{\bar{m}}) - \mathcal{F}^i(t_i, t_2, \dots, t_{\bar{m}}) < 0$ for $i = 1, 2, \dots, n$. Thus, equation (23) 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 an uniformly asymptotical stable configuration.

Conversely, assume that $\underline{p}^o$ is an uniformly asymptotical stable configuration, then the linear approximation of ODE (17) can be written as $\dot{\underline{\tilde{p}}} = A\underline{\tilde{p}}$, where $A = \operatorname{diag}\left(\tilde{f}_{iy}\right)$ and $\tilde{f}_{iy} = \mathcal{F}^i(y, t_2, \dots, t_{\bar{m}}) - \mathcal{F}^i(t_i, t_2, \dots, t_{\bar{m}})$ for $i = 1, 2, \dots, n$. Since $\underline{p}^o$ is uniformly asymptotical stable, A should have eigenvalues with negative real parts and hence $\tilde{f}_{iy} < 0$. Using Theorem 2, this implies that $\underline{p}^o$ is a fully compatible configuration. This completes the proof of this lemma. ■

Lemma 8: Non-compatible 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 3, there is a learning automaton 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 (18), 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. ■

C. Convergence Results

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

1. $\underline{p}(0)$ is close to a compatible corner $\underline{p}^o$. By lemma 7, 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 non-compatible corner $\underline{p}^o$. By lemma 8, no matter how small neighborhood we take around $\underline{p}^o$, the solution of (17) will leave that neighborhood and enter $\mathcal{K} - \mathcal{K}^*$. The convergence when the initial configuration is in $\mathcal{K} - \mathcal{K}^*$ is discussed in case 4 below.
3. $\underline{p}(0) \in \mathcal{K}^*$. Using the convergence properties of L_{R-I} learning algorithm [5], no matter whether $\underline{p}(0)$ is compatible or not, the ACLA will be absorbed to $\underline{p}(0)$.
4. $\underline{p}(0) \in \mathcal{K} - \mathcal{K}^*$. The convergence results of the ACLA for these initial configurations is stated in theorem 4.

Theorem 4: Suppose there is a bounded differential function $\mathcal{D} : \mathcal{R}^{m_1 + \dots + m_{\bar{m}}} \to \mathcal{R}$ such that for some constant $c > 0$, $\frac{\partial \mathcal{D}}{\partial p_{ir}}(\underline{p}) = c d_{ir}(\underline{p})$ for all i and r, and $\rho_i > 0$ for all i. Then asynchronous CLA for any initial configuration in $\mathcal{K} - \mathcal{K}^*$ and with sufficiently small value of learning parameter $(\max\{\underline{a}\} \to 0)$, always converges to a configuration, that is stable and compatible.

Proof: Consider the variation of $\mathcal{D}$ along the solution paths of ODE (17), $\mathcal{D}$ is non-decreasing because

$$\frac{d\mathcal{D}}{dt} = \sum_i \sum_y \frac{\partial \mathcal{D}}{\partial p_{iy}} \frac{\partial p_{iy}}{\partial t} \geq 0.$$

The details of the above derivation can be found in [19]. The ACLA updates the action probabilities in a such a way that $\underline{p}(k) \in \mathcal{K}$ for all $\underline{p}(0) \in \mathcal{K}$ and $k > 0$. Since $\mathcal{K}$ is a compact subset of $\mathcal{R}^{m_1 + \dots + m_{\bar{m}}}$, asymptotically all solutions of ODE (17) will be in $\mathcal{K}$. Inequality (24) shows that ACLA updates the configuration probabilities in

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

Fig. 1. The linear ACLA.

gradient ascent manner and hence, converges to a maximum of $\mathcal{D}$, where $\frac{d\mathcal{D}}{dt} = 0$. From (24), the derivative of F is zero if and only if for all i, y, r, we have $p_{ir}p_{iy} = 0$ or $p_{iy} = p_{ir}$. From lemmas 5 and 6, these configurations are equilibrium points of $f_{iy}(\underline{p})$. Thus the solution to ODE (17) for any initial configuration in $\mathcal{K} - \mathcal{K}^*$ will converge to a set containing only equilibrium points of the ODE (17). Since all equilibrium configurations that are not compatible are unstable, the theorem follows. ∎

IV. Asynchronous Cellular Learning Automata using Commutative Rules

In this section, we study the behavior of the ACLA when the commutative rules are used. Commutativity is a property of hyper matrix $\mathcal{F}^i$ as given in the following definition.

Definition 10: A rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}})$ is called commutative if and only if

$$\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) = \ldots = \mathcal{F}^i(\alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_1}). \tag{24}$$

In order to simplify the algebraic manipulations, we give the analysis for the linear ACLA. The linear ACLA, as shown in figure 1, uses the 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 5: For a ACLA, which uses a commutative rule, a configuration $\underline{p}$ at which $\mathcal{D}(\underline{p})$ is a local maximum, is compatible.

Proof: Since $\mathcal{K}$ is convex, then for every $0 \leq \lambda \leq 1$ and $\underline{q} \in \mathcal{K}$, we have $\lambda\underline{q} + (1-\lambda)\underline{p} \in \mathcal{K}$. Suppose that $\underline{p}$ is a configuration for which $\mathcal{D}(\underline{p})$ is a local maximum, then $\mathcal{D}(\underline{p})$ doesn't increase as one moves away from $\underline{p}$. Thus we have

$$\left.\frac{d\mathcal{D}(\lambda\underline{q} + (1-\lambda)\underline{p})}{d\lambda}\right|_{\lambda=0} \leq 0. \tag{25}$$

Thus using chain rule, we obtain $\nabla\mathcal{D}(\underline{p})(\underline{q} - \underline{p}) \leq 0$. $\nabla\mathcal{D}(\underline{q})$ has M elements in which (l, r)th component of $\nabla F(\underline{q})$ is denoted by q_{lr} and calculated by the following equation. Let $y = \alpha_{i+\bar{x}_1}$, $x = \alpha_{i+\bar{x}_2}$, $z = \alpha_{i+\bar{x}_3}$, $j = i-1$, and $k = i+1$.

$$q_{lr} = \frac{\partial}{\partial p_{lr}} \sum_i \sum_y \sum_x \sum_z \mathcal{F}^i(y, x, z) p_{jx} p_{iy} p_{kz}$$

$$= 3d_{lr}(\underline{p}).$$

The details of the above derivation can be found in [19]. Thus, $\nabla\mathcal{D}(\underline{p})(\underline{q} - \underline{p}) \leq 0$, which implies that

$$\nabla\mathcal{D}(\underline{p})(\underline{q} - \underline{p}) = 3\sum_i \sum_y d_{iy}(\underline{p})\,[q_{iy} - p_{iy}] \leq 0.$$

The above inequality is true for all $\underline{q} \in \mathcal{K}$. So, $\underline{p}$ satisfies the condition of theorem 1, and hence $\underline{p}$ is a compatible configuration. ∎

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

Theorem 6: An asynchronous CLA, which uses uniform and commutative rule, starting from $\underline{p}(0) \in \mathcal{K} - \mathcal{K}^*$ and with sufficiently small value of learning parameter, $(\max\{\underline{a}\} \to 0)$, always converges to a deterministic configuration, that is stable and compatible.

Proof: Let function $\mathcal{D} : \mathcal{R}^{m_1+\cdots+m_{\bar{m}}} \to \mathcal{R}$ be the total average reward for the ACLA. Hence, we have $\frac{\partial\mathcal{D}}{\partial p_{ir}}(\underline{p}) = 3d_{ir}(\underline{p})$ for all i and r. Using theorem 4 convergence of ACLA can be concluded. ∎

V. Numerical Examples

This section discusses patterns formed by the evolution of cellular learning automata from a random initial configuration. For the sake of simplicity in our presentation, we use the following notation to specify the rules for the cellular learning automata for which each cell has a learning automaton with m actions. The actions of each learning automaton are represented by integers in the interval $[0, m-1]$. Hence, the configuration of each cell and its neighbors form 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. We identify a rule by the decimal representation of this $m^{\bar{m}}$-bit number. We use notation $(j)_m$ to specify the rules of the ACLA, where j is a decimal number representing the rule and m is the number of actions of the learning automaton. For example, table I represents the rule $(22)_2$ for a linear ACLA with two-actions learning automata and the neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. In this table, 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 corresponding reinforcement signal (β) to be output to the learning automata.

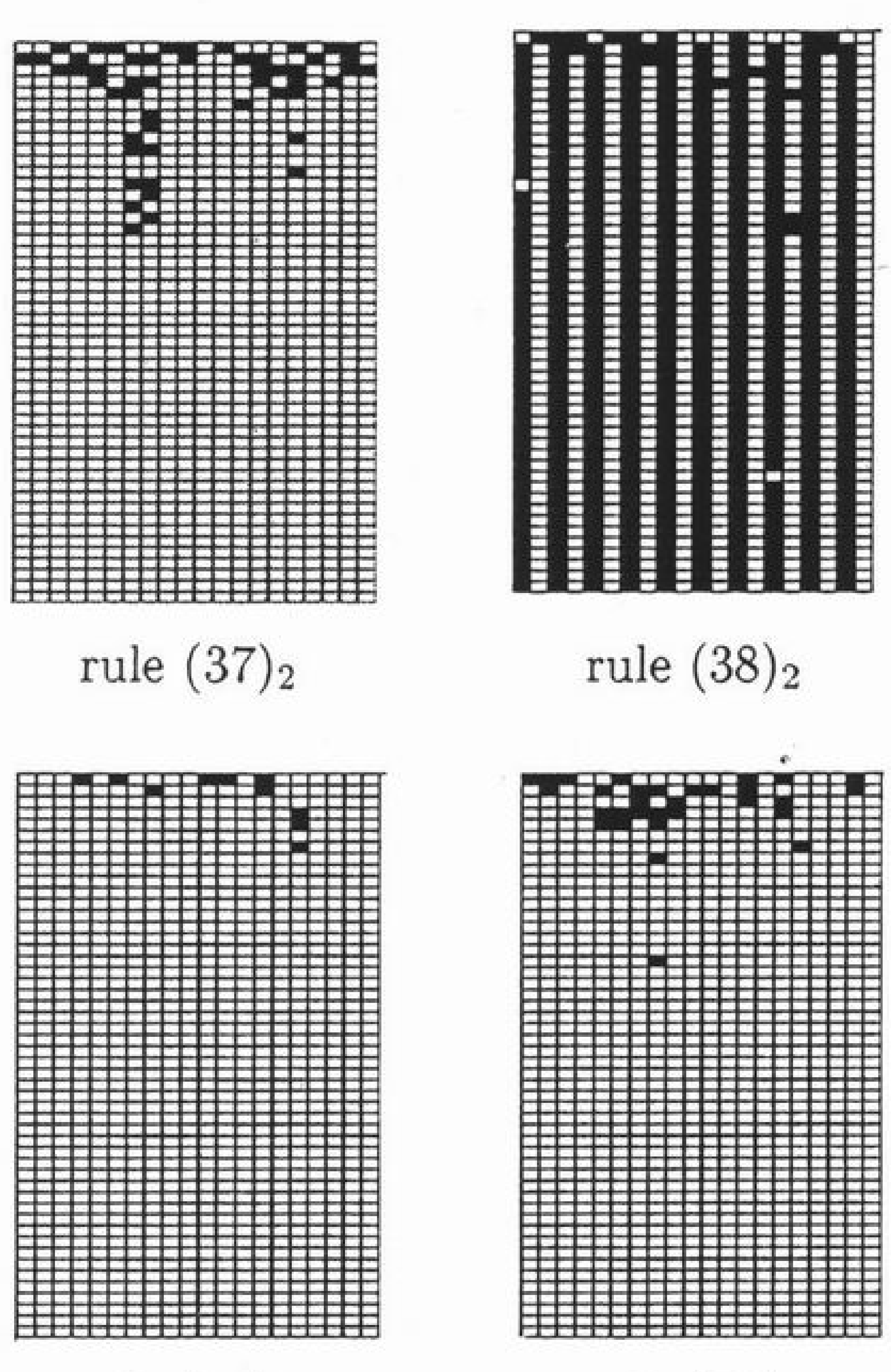
In the experiments presented below, the asynchronous CLA are considered. Figure 2 shows the time-space diagram evolution of ACLA with 20

TABLE I
THE SCHEME FOR THE RULE NUMBERING FOR TWO ACTIONS LA

$\underline{\alpha}$	111	110	101	100	011	010	001	000
β	0	0	0	1	0	1	1	0

cells and a two-actions L_{R-I} learning automaton in each cell. The probability of activation of learning automata is the following fixed vector

$$\begin{aligned} \underline{\rho} &= (0.1, 0.1, 0.2, 0.1, 0.3, 0.5, 0.6, 0.8, 0.1, \\ & \quad 0.1, 0.2, 0.1, 0.3, 0.5, 0.6, 0.8, 0.1, 0.1, 0.2, 0.1)'. \end{aligned}$$



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

rule $(39)_2$ rule $(47)_2$

Fig. 2. Time-space diagram for asynchronous CLA

VI. CONCLUSIONS

In this paper, the asynchronous cellular learning automata was introduced and its convergence behavior was studied. It was shown that for commutative rules, the asynchronous cellular learning automata converges to a stable configuration for which the average reward for the ACLA is maximum. The numerical results also confirms the theory.

REFERENCES

[1] E. Fredkin, "Digital Machine: A Informational Process Based on Reversible Cellular Automata," *Physica D*, vol. 45, pp. 254–270, 1990.

[2] N. H. Packard and S. Wolfram, "Two-Dimensional Cellular Automata," *Journal of Statistical Physics*, vol. 38, pp. 901–946, 1985.

[3] M. Mitchell, "Computation in Cellular Automata: A Selected Review," tech. rep., Santa Fe Institute, Santa Fe, NM, USA, Sept. 1996.

[4] J. Kari, "Reversability of 2D Cellular Automata is Undecidable," *Physica D*, vol. 45, pp. 379–385, 1990.

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

[6] P. R. Srikantakumar and K. S. Narendra, "A Learning Model for Routing in Telephone Networks," *SIAM Journal of Control and Optimization*, vol. 20, pp. 34–57, Jan. 1982.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] M. R. Meybodi and M. Taherkhani, "Application of Cellular Learning Automata in Modeling of Rumor Diffusion," in *Proceedings of 9th Conference on Electrical Engineering, Power and Water Institute of Technology, Tehran, Iran*, pp. 102–110, May 2001.

[13] M. R. Kharazmi and M. R. Meybodi, "Image Segmentation Using Cellular Learning Automata," in *Proceedings of 10th Iranian Conference on Electrical Engineering, ICEE-96, Tabriz, Iran*, May 2001.

[14] M. R. Kharazmi and M. R. Meybodi, "Image Restoration Using Cellular Learning Automata," in *Proceedings of the Second Iranian Conf. on Machine Vision, Image Processeing and Applications, Tehran, Iran*, pp. 261–270, 2003.

[15] M. R. Meybodi and M. R. Kharazmi, "Application of Cellular Learning Automata to Image Processing," *Journal of Amirkabir, to be published*, 2003.

[16] 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.

[17] M. R. Meybodi, H. Beigy, and M. Taherkhani, "A Mathematical Framework for Cellular Learning Automata," *Journal of Advances in Complex Systems, to be appreared*.

[18] H. Beigy and M. R. Meybodi, *Call Admission in Cellular Networks: A Learning Automata Approach*, vol. 2510 of *Springer-Verlag Lecture Notes in Computer Science*, pp. 450–457. Springer-Verlag, Oct. 2002.

[19] H. Beigy and M. R. Meybodi, "Asynchronous Cellular Learning Automata," Tech. Rep. TR-CE-2003-006, Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran, 2003.

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