

Open Synchronous Cellular Learning Automata

Hamid Beigy^{1,2}

Mohammad Reza Meybodi^{2,3}

¹Computer Engineering Department, Sharif University of Technology, Tehran, Iran

²School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran

³Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran

Abstract

In this paper, we introduce open cellular learning automata and then study its convergence behavior. It is shown that for a class of rules called commutative rules, the open cellular learning automata in stationary external environments converges to a stable and compatible configuration. The numerical results also confirm the theory.

Keywords: open synchronous cellular learning automata, learning automata, commutative rules

1. Introduction

In recent years cellular automata have frequently been used to model the dynamics of spatially extended physical systems. Examples include a wide range of topics, such as prebiotic evolution [4], the development of pigment patterns in mollusks [5], and growth of clonal plants [6], to mention a few. Cellular automata are a collection of cells that each adapts one of a finite number of states. Single cells change in state following a local rule that depends on the environment of the cell. The environment of a cell usually taken to be a small number of neighboring cells. The dynamics of cellular automata is generated by repeatedly applying the local rule to all cells in the cellular automata. The cellular automata evolve in discrete steps, changing the states of all its cells according to the local rule, homogenously applied at each step. Cellular automata perform complex computation with high degree of efficiency and robustness.

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 interaction of the learning automaton and its environment is shown in figure 1. 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 $\underline{LR-I}$, 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 $\underline{p}(k)$. In $\underline{LR-I}$ algorithm, the action probability vector is updated according the following rule.

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

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. Parameter $0 < \alpha < 1$ represents *step length* and r is the number of actions for LA [19]. LA have been used successfully in many applications such as telephone and data network routing [22], solving NP-Complete problems [20], capacity assignment [21], neural network engineering [10,11] and call admission in cellular networks [3] to mention a few.

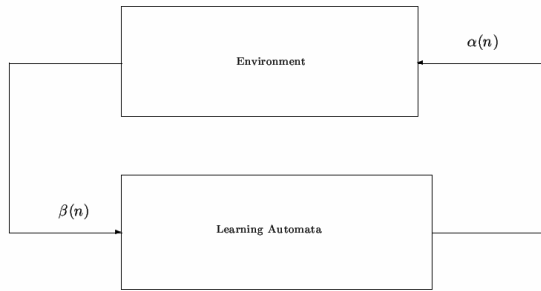


Figure 1. The interaction of a learning automaton and its environment

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 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 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 [12], CA and LA are 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 learning agents [12]. Simple, agents which have learning capability, act together to produce complicated behavioral patterns. A CLA is a CA in which a learning automaton will be assigned to its every cell. The learning automaton residing in each cell determines the state of the cell on the basis of its action probability vector. Like CA, there is a rule that CLA operate under it. The rule of CLA and the actions selected by the neighboring LAs of any cell determine the reinforcement signal to the LA residing in that cell. In CLA, the neighboring LAs of any cell constitute its local environment. This environment is nonstationary because of the fact that it changes as action probability vectors of neighboring LAs 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 automaton 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 $\mathcal{A} = (\mathcal{Z}^d, \Phi, \mathcal{A}, \mathcal{N}, \mathcal{F})$, where

1. \mathcal{Z}^d is a lattice of d -tuples of integer numbers.
2. Φ is a finite set of states.
3. \mathcal{A} is the set of LAs each of which is assigned to one cell of the CLA.
4. $\mathcal{N} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m\}$ is a finite subset of \mathcal{Z}^d called neighborhood vector, where $\bar{x}_i \in \mathcal{Z}^d$.
5. $\mathcal{F} : \Phi^m \rightarrow \underline{\beta}$ is the local rule of the cellular learning automata, where $\underline{\beta}$ is the set of values that the reinforcement

signal can take. It computes the reinforcement signal for each LA based on the actions selected by the neighboring LAs. A number of applications for CLA have been developed recently such as rumor diffusion [17], image processing [7,8,13,14], modeling of commerce networks [15], fixed channel assignment in cellular networks [2], and VLSI Placement [16]. The CLA can be classified into *synchronous* and *asynchronous*. In synchronous CLA, all cells are synchronized with a global clock and executed at the same time. In [1], a mathematical methodology to study the behavior of the synchronous CLA is given and its convergence properties have 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 a type of cellular learning automata in which reinforcement signal of each cell in next stage of its evolution not only depends on the local environment (actions of its neighbors) but it also depends on the external environments. We call such a CLA as *open CLA*. In this paper, we introduce open synchronous CLA and study its convergence behavior. It is shown that for a class of rules called commutative rules, the open synchronous CLA converges to a globally stable state in stationary external environments.

The rest of this paper is organized as follows. In section 2, the open synchronous CLA is presented. Section 3 presents the convergence behavior of open CLA. In section 4, the behavior of the open CLA when the commutative rules are used is studied. Section 5 presents the numerical example and section 6 concludes the paper.

2. Open Synchronous CLA

CLA studied so far are closed, because they don't take into account the interaction between the CLA and the external environments. In this section, a new class of CLA called *open CLA* (OSCLA), in which the evolution of CLA is influenced by the external environments, is introduced. Two types of environments can be considered in the open CLA: global environment and exclusive environment. Each CLA has one global environment that influences all cells and an exclusive environment for each particular cell. The interconnection of a typical cell in the open CLA and its various types of environments is shown in the following figure.

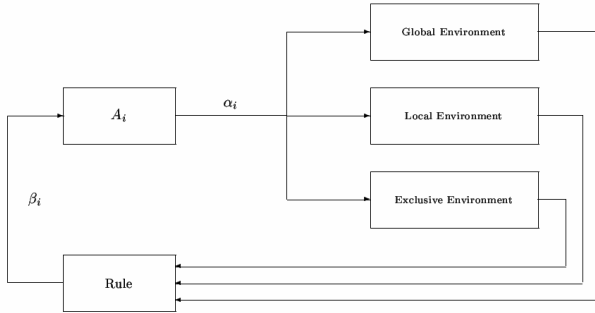


Figure 2. The interconnection of a typical cell in open CLA with its various environments

Formally, a d -dimensional open cellular learning automata is defined as given below.

Definition 2. A d -dimensional open cellular learning automata is a structure $\mathcal{A} = (\mathbb{Z}^d, \Phi, \mathcal{A}, E^G, E^E, N, \mathcal{F})$,

where

1. \mathbb{Z}^d is a lattice of d -tuples of integer numbers.
2. Φ is a finite set of states.
3. \mathcal{A} is the set of LAs each of which is assigned to each cell of the CA.
4. E^G is the global environment.
5. $E^E = \{E_1^E, E_2^E, \dots, E_n^E\}$ is the set of exclusive environments, where E_i^E is the exclusive environment for cell i .
6. $N = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m\}$ is neighborhood vector.
7. $\mathcal{F} : \Phi^m \times \underline{O(G)} \times \underline{O(E)} \rightarrow \underline{\beta}$ is the local rule of the cellular automata, where $\underline{O(G)}$ and $\underline{O(E)}$ is the set of signals of global and exclusive environments, respectively.

The operation of an open CLA takes place as iterations of the following steps. At iteration k , each learning automaton chooses one of its actions. Let α_i be the action chosen by learning automaton A_i . The actions of all learning automata

are applied to their corresponding local environments (neighboring learning automata) as well as global environment and their corresponding exclusive environment. Then all learning automata receive their reinforcement signal, which is combination of the responses from local, global and exclusive environments. These responses are combined using the local rule. Finally, all learning automata update their action probability vectors based on the received reinforcement signal. Note that the local environment for each learning automaton is nonstationary while global and exclusive environments may be stationary or nonstationary. In what follows, we consider an open CLA 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 the open CLA. Let cardinality of $\underline{\alpha}_i$ be m_i and the state of the open CLA 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 open CLA can be very complex.

Based on set $\underline{\beta}$, the OSCLA can be classified into three groups: P-model, Q-model, and S-model OSCLA. When $\underline{\beta} = \{0, 1\}$, we refer to OSCLA as *P-model OSCLA*,

when $\underline{\beta} = \{b_1, \dots, b_l\}$, (for $l < \infty$), we refer to

OSCLA as *Q-model OSCLA*, and when $\underline{\beta} = [b_1, b_2]$, we

refer to OSCLA as *S-model OSCLA*. If learning automaton A_i uses learning algorithm L_i , we denote OSCLA by the

$OSCLA(L_1, \dots, L_n)$. If $L_i = L$ for all

$i = 1, \dots, n$, then we denote the OSCLA by the

$OSCLA(L)$.

In the following subsections, we give some definitions and notations, which will be used later in section 3 to analysis the behavior of OSCLA.

2.1 Definitions and notations

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

Definition 3. A configuration of OSCLA is a map $\mathcal{K} : \mathbb{Z}^d \rightarrow \underline{p}$ that associates an action probability vector with every cell. We will denote the set of all configurations of \mathcal{A} by $\mathcal{K}(\mathcal{A})$ or simply \mathcal{K} .

Definition 4. A configurations \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 OSCLA are

$$\mathcal{K}^* = \{ \underline{p} | \underline{p} = (\underline{p}'_1, \underline{p}'_2, \dots, \underline{p}'_n)', \underline{p}_i = (p_{i1}, \dots, p_{im_i})', \\ p_{iy} = 0 \text{ or } 1 \ \forall y, i, \sum_y p_{iy} = 1 \ \forall i \}$$

and

$$\mathcal{K} = \{ \underline{p} | \underline{p} = (\underline{p}'_1, \underline{p}'_2, \dots, \underline{p}'_n)', \underline{p}_i = (p_{i1}, \dots, p_{im_i})', \\ 0 \leq p_{iy} \leq 1 \ \forall y, i, \sum_y p_{iy} = 1 \ \forall i \},$$

respectively.

In the following lemma, it is shown that \mathcal{K} is a convex hull of \mathcal{K}^* .

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

Proof: Let $\underline{M} = \sum_i m_i$ and \underline{e}_k be a unit vector of appropriate dimension in the k th direction. Then any configuration $\underline{p} \in \mathcal{K}$ can be expressed by

$$\underline{p} = \sum_{i_1=1}^{m_{i_1}} \sum_{i_2=1}^{m_{i_2}} \dots \sum_{i_n=1}^{m_{i_n}} ((p_{1i_1} \underline{e}_{i_1}), (p_{2i_2} \underline{e}_{i_2}), \dots, (p_{ni_n} \underline{e}_{i_n})). \quad (2)$$

Since each \underline{M} -vector $(\underline{e}_{i_1}, \underline{e}_{i_2}, \dots, \underline{e}_{i_n})$ is in \mathcal{K}^* , then

the above sum can be interpreted as a convex combination of the elements of \mathcal{K}^* .

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

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

Definition 6. The evolution of the OSCLA from a given initial configuration $\underline{p}(0) \in \mathcal{K}$ is a sequence of configurations $\{\underline{p}(k)\}_{k \geq 0}$, 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

$$d_{ir}(\underline{p}) = \sum_{y_2} \dots \sum_w \mathcal{F}^i(r, y_2, \dots, y_{\bar{m}}, z, w) d_z^G d_w^E \prod_{\substack{i \in \bar{N}(i) \\ i \neq i}} p_{iy_i}, \quad (3)$$

where $z \in O(G)$ and $w \in O(E_i)$ are the signals produced by the global and the exclusive environments of cell i , respectively and d_z^G and d_w^E are the probability of producing the responses z and w by the global and the exclusive environments, respectively. The average reward for learning automaton A_i is equal to

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

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

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

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 OSCLA have any reason to change its action.

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

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

Lemma 2 The OSCLA 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 $i = 1, \dots, n$ and

$r = 1, \dots, m_i$. Note that $\psi_{ir}(\underline{p})$ and $\phi_{ir}(\underline{p})$ are

continuous functions on \mathcal{K} . Introducing the mapping $T : \mathcal{K} \rightarrow \mathcal{K}$ given by

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

for $i = 1, \dots, n$ and $r = 1, \dots, m_i$. 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 OSCLA and conversely every compatible configuration of the OSCLA 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 $\sum_r d_{ir}(\underline{p}) p_{ir} \geq \sum_r d_{ir}(\underline{q}) q_{ir}$ for all $i = 1, \dots, n$.

Configuration \underline{q} also includes $\underline{q} = (\underline{q}'_1, \dots, \underline{q}'_n)'$ for

fixed $i (i = 1, \dots, n)$. 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 $i = 1, \dots, n$ and $r_i = 1, \dots, m$ 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 \bar{p}_i such that $\bar{p} = (\underline{p}'_1, \dots, \bar{p}'_i, \dots, \underline{p}'_n)'$ and

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

Let $y_i (1 \leq i \leq m_i)$ be an action for which $d_{ir}(\underline{p})$ attains its maximum value. Then $D_i(\underline{p})$ 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 $\bar{p}_{iy_i} < \bar{p}_{iy_i}$, because $\sum_j \phi_{ij}(\underline{p}) > 0$ and contradicts the hypothesis that \underline{q} is a fixed point of T .

Lemma 3 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 $\underline{q} = (\underline{p}'_1, \dots, \underline{q}_{r_i}, \dots, \underline{p}'_n)'$ for fixed $i (i = 1, \dots, n)$ 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 $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$ (for $i = 1, \dots, n$ and $r_i = 1, \dots, m$) 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 $\underline{q} = (\underline{p}'_1, \dots, \underline{q}_i, \dots, \underline{p}'_n)'$ 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 4 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 3, 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 $\underline{p} \in \mathcal{K}$ 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 $\underline{q} = (\underline{p}'_1, \dots, \underline{q}_{r_i}, \dots, \underline{p}'_n)'$ for fixed i (for $i = 1, \dots, n$) 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 $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$ (for $i = 1, \dots, n$ and $r_i = 1, \dots, m$) but \underline{p} is not compatible. Then for some learning automaton i with action probability vector \underline{q}_i there

exists an action \underline{y}_i such that $\underline{q} = (\underline{q}'_1, \dots, \underline{q}_i, \dots, \underline{q}'_n)'$ and $d_{i\underline{y}_i}(\underline{p}) > D_i(\underline{q})$. Action \underline{y}_i denotes the action for which $d_{i\underline{y}_i}(\underline{p})$ attains its maximum value. Since \underline{q}_i is a probability vector, then $D_i(\underline{q})$ is bounded from the above with $d_{i\underline{y}_i}(\underline{p})$ and arrives at strict inequality $D_i(\underline{p}) < D_i(\underline{q}) < d_{i\underline{y}_i}(\underline{p})$. But this contradicts the hypothesis that $d_{i\underline{r}_i}(\underline{p}) \leq D_i(\underline{p})$, which concludes that \underline{p} is a compatible configuration.

This theorem states that, when the action probability vector of all learning automata except the specific \underline{A}_i are held fixed, then the configuration reached by the OSCLA at the point, where the average reward of \underline{A}_i is maximum, is compatible.

Theorem 2 A corner $\underline{p} = (\underline{e}_{t_1}, \underline{e}_{t_2}, \dots, \underline{e}_{t_n})'$ is compatible if and only if $\mathcal{F}^i(t_1, t_2, \dots, t_{\bar{m}}, z, w) \geq \mathcal{F}^i(r_1, t_2, \dots, t_{\bar{m}}, z, w)$ for all $z \in \underline{O}(G), w \in \underline{O}(E^i), i = 1, \dots, n$ and $r_i \neq t_i$.

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

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

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

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

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

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

Conversely, assume that

$$\mathcal{F}^i(t_1, t_2, \dots, t_{\bar{m}}, z, w) \geq \mathcal{F}^i(r_1, t_2, \dots, t_{\bar{m}}, z, w)$$

but \underline{p} is not compatible. From definition 8 and by some algebraic simplification we obtain

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

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.

Corollary 1 A corner $\underline{p} = (\underline{e}_{t_1}, \underline{e}_{t_2}, \dots, \underline{e}_{t_n})'$ is fully compatible if and only if $\mathcal{F}^i(t_1, t_2, \dots, t_{\bar{m}}, z, w) \geq \mathcal{F}^i(r_1, t_2, \dots, t_{\bar{m}}, z, w)$ for all $z \in \underline{O}(G), w \in \underline{O}(E^i), i = 1, \dots, n$ and $r_i \neq t_i$.

Proof: The proof is trivial from the proof of theorem 2.

3. Behavior of Open Synchronous CLA

In this section, we analyze the open synchronous OSCLA in which all learning automata use the \underline{L}_{R-I} learning algorithm and operates under stationary global and exclusive environments. We denote this OSCLA by OSCLA(\underline{L}_{R-I}).

Using \underline{L}_{R-I} learning algorithm, process $\{\underline{p}(k)\}_{k \geq 0}$ is Markovian and can be described by the following difference equation.

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

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 \underline{A}_i . Now, define

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

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

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

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

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

where

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

For different values of \underline{a} , equation (11) generates different process and we shall use $\underline{p}^a(k)$ to denote this process whenever the value of \underline{a} is to be specified explicitly. Define a sequence of continuous-time interpolation of (11), denoted by $\bar{\underline{p}}^a(t)$ and called **interpolated process**, whose components are defined by

$$\bar{\underline{p}}_i^a(t) = \underline{p}_i(k) \quad t \in [ka_i, (k+1)a_i], \quad (16)$$

where a_i is the learning parameter of the L_{R-I} algorithm for learning automaton A_i . The objective is to study the limit of sequence $\{\bar{\underline{p}}^a(t)\}_{t \geq 0}$ as $\max\{a\} \rightarrow 0$, which will be a good approximation to the asymptotic behavior of (16). When learning parameter a_i is sufficiently small for all $i = 1, 2, \dots, n$, then equation (13) can be written as the following ordinary differential equation (ODE).

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

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

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

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 $\bar{\underline{p}}^a$ as $\max\{a\}$ is sufficiently small. We show that the sequence of interpolated process $\{\bar{\underline{p}}^a(t)\}$ 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 $\{\bar{\underline{p}}^a(\cdot)\}$ converges weakly to the solution of

$$\frac{dX}{dt} = \underline{f}(X) \quad (19)$$

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

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{q} is bounded, continuous and independent of a .
4. ODE (19) has a unique solution for each initial condition $X(0)$.
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 $M^{\bar{\underline{p}}}$ be the distribution of process $\{(\underline{\alpha}(k), \underline{\beta}(k))\}_{k \geq 0}$.

Then using the weak convergence theorem [9], sequence $\{\bar{\underline{p}}^a(\cdot)\}$ converges weakly, as $\max\{a\} \rightarrow 0$ to the solution of

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

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

$$\bar{\underline{f}}(\bar{\underline{p}}) = E[\underline{f}(\bar{\underline{p}}(k), \underline{\alpha}(k), \underline{\beta}(k))] = \underline{f}(\bar{\underline{p}}), \quad \text{and hence the theorem.}$$

Theorem 3 enables us to understand the long term behavior of $\underline{p}(k)$. The weak convergence in this theorem implies that path $\bar{\underline{p}}^a(t)$ will closely follow the solution to the ODE on any finite interval with an arbitrarily high probability as $\max\{a\} \rightarrow 0$. As the length of the time interval increases and $\max\{a\} \rightarrow 0$, 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, $\bar{\underline{p}}^a(\cdot)$ will eventually (with an arbitrarily high probability) spend all of its time in a small neighborhood of \underline{p}^o as well. As

$\max\{\underline{a}\} \rightarrow 0$, the time interval over which the evolution of the CLA follows the path of the ODE goes to infinity. Although the speed of convergence depends on the specific value of \underline{a} . The above point is summarized in the following lemma.

Lemma 5 For large k and small enough value of $\max\{\underline{a}\}$, the asymptotic behavior of $\underline{p}(k)$ generated by the CLA can be approximated by the solution to ODE (19) with the same initial configuration.

Proof: Let $\underline{X}(\cdot)$ be the solution of ODE (19) with initial condition $\underline{X}(0) = \underline{X}_0$ 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 $h_T(\cdot)$ is continuous on \mathcal{K} . Then theorem 3 says that $E[h_T(\underline{p})^a] \rightarrow E[h_T(\underline{X})] = 0$ as $\max\{\underline{a}\} \rightarrow 0$. 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 of the distance between the original sequence $\underline{p}(t)$ and $\underline{X}(t)$ goes to zero in probability as $k \rightarrow \infty$. 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

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

where $a = \max\{\underline{a}\}$. Since \underline{p}^o is an asymptotically stable equilibrium point of ODE (19), then for all initial configurations in small neighborhood of \underline{p}^o , the OSCLA 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 OSCLA.

3.1 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 6 All the corners of \mathcal{K} are equilibrium points of $\underline{f}(\cdot)$. All the other equilibrium points \underline{p} of $\underline{f}(\cdot)$ satisfy

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

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

Proof: From equation (14), it is obvious that $f_{iy} = 0$ (for $i = 1, 2, \dots, n$) if \underline{p}_i is a unit vector and hence all corners of \mathcal{K} are equilibrium points of $\underline{f}(\cdot)$. In order to find other equilibrium points of $\underline{f}(\cdot)$, 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.

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

The above equation can be rewritten as

$$\begin{aligned} & \sum_{r \neq y} p_{ir}(k) [d_{iy}(\underline{p}) - d_{ir}(\underline{p})] \\ &= \sum_{r \neq y} p_{ir}(k) d_{iy}(\underline{p}) - \sum_{r \neq y} p_{ir}(k) d_{ir}(\underline{p}) = 0. \end{aligned} \quad (22)$$

Thus, we obtain

$$\sum_{r \neq q} [d_{iq}(\underline{p}) - d_{ir}(\underline{p})] p_{ir}(k) = d_{iq}(\underline{p}) - d_{iy}(\underline{p}), \quad (23)$$

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

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

When $d_0 \neq 0$, equation (23) 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

$$d_{iq}(\underline{p}) = d_{iy}(\underline{p}), \text{ for } \forall i = 1, 2, \dots, n \text{ and } \forall y \neq q.$$

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 7 All compatible configurations are equilibrium points of $\underline{f}(\cdot)$.

Proof: Let \underline{p} be a compatible configuration. Then by lemma 4, 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 .

3.2 The Stability property

In this subsection we characterize the stability of equilibrium configurations of OSCLA that is the equilibrium points of the ODE (17). From lemmas 6 and 7, 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 8 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 $\underline{p}^o = (\underline{e}'_1, \dots, \underline{e}'_n)'$ be a corner of \mathcal{K} that is a fully compatible configuration. Using the transformation defined by

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

the origin is translated to \underline{p}^o . 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

$$f_{iy} = \bar{p}_{iy} [\mathcal{F}^i(y, t_2, \dots, t_m, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \dots, t_m, t_g, t_e)] + \text{high order terms.} \quad (24)$$

We consider the following positive definite Lyapunov function $V(\underline{\bar{p}}) = \sum_i \sum_{y \neq t_i} \bar{p}_{iy}$, where $V(\underline{\bar{p}}) \geq 0$

and is zero when $\bar{p}_{iy} = 0$ for all i, y , and its derivative is

equal to $\dot{V}(\underline{\bar{p}}) = \sum_i \sum_{y \neq t_i} f_{iy}$. Since corner \underline{p}^o is a fully compatible configuration, then from theorem 2, we

have $\mathcal{F}^i(y, t_2, \dots, t_m, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \dots, t_m, t_g, t_e) < 0$ for $i = 1, 2, \dots, n$. Thus, equation (24) implies that there

is a neighborhood around \underline{p}^o such that the linear terms

dominate the high order terms. Hence, $\dot{V}(\underline{\bar{p}}) < 0$ 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 $\dot{\underline{\bar{p}}} = \underline{A}\underline{\bar{p}}$, where $\underline{A} = \text{diag}(\bar{f}_{iy})$ and

$$\bar{f}_{iy} = \mathcal{F}^i(y, t_2, \dots, t_m, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \dots, t_m, t_g, t_e) \text{ for } i = 1, 2, \dots, n$$

. Since \underline{p}^o is uniformly asymptotical stable, \underline{A} should have

eigenvalues with negative real parts and hence $\bar{f}_{iy} < 0$.

Using theorem 2, this implies that \underline{p}^o is a fully compatible configuration. This completes the proof of this lemma.

Lemma 9 Incompatible equilibrium points of $\underline{f}(\cdot)$ are unstable.

Proof: Let \underline{p}^o be an equilibrium point of $\underline{f}(\cdot)$ which is not

compatible. Then by lemma 4, there is a learning automaton A_j and an action y such that $d_{jy}(\underline{p}) > D_j(\underline{p})$. Since

$d_{jy}(\underline{p})$ and $D_j(\underline{p})$ are continuous, then inequality

$d_{jy}(\underline{p}) > D_j(\underline{p})$ will hold in small open neighborhood

around \underline{p}^o . Using (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.

Remark 1 In lemmas 8 and 9, the solution of ODE (17) well characterized and it is shown that full compatibility implies uniformly asymptotic stability of the corners. In order to obtain necessary and sufficient conditions for uniformly asymptotic stability, it is essential to consider in detail the nonlinear terms in the differential equation, which appears to be a difficult problem.

3.3 Convergence results

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

1. $\underline{p}(0)$ is close to a compatible corner \underline{p}^o . By lemma 8,

there is a neighborhood around \underline{p}^o entering which, the

OSCLA will be absorbed by that corner. Thus, the OSCLA converges to a compatible configuration.

2. $\underline{p}(0)$ is close to a incompatible corner \underline{p}^o . By lemma 9, 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 [19], no matter whether $\underline{p}(0)$ is compatible or not, the OSCLA will be absorbed to $\underline{p}(0)$.
 4. $\underline{p}(0) \in \mathcal{K} - \mathcal{K}^*$. The convergence results of the OSCLA for these initial configurations are stated in theorem 4.
- Theorem 4** Suppose there is a bounded differential function $\mathcal{D} : \mathcal{R}^{m_1 + \dots + m_n} \times \mathcal{O}(\mathcal{G}) \times \mathcal{O}(\mathcal{E}) \rightarrow \mathcal{R}$ 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 $p_i > 0$ for all i . Then OSCLA for any initial configuration in $\mathcal{K} - \mathcal{K}^*$ and with sufficiently small value of learning parameter ($\max\{\underline{a}\} \rightarrow 0$), always converges to a configuration, that is stable and compatible.

Proof: In order to prove the convergence of OSCLA, we use an additional dimension for representing the global and exclusive environments. For the sake of simplicity, we use a linear OSCLA as an example. Consider a linear OSCLA with n cells and neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$.

For global environment, we add an extra row to this OSCLA, say row 0, containing n identical cells and for exclusive environment, we add again an extra row, say row 2, containing n cells. Now, the original OSCLA becomes row 1 of the new CLA. To consider the effects of the global and exclusive environments on each learning automaton, the neighborhood function must also be modified. The modified neighborhood functions is

$$\bar{N}_1(i, j) = \{(i, j), (i, j-1), (i, j+1), (i-1, j), (i+1, j)\},$$

where operators $+$ and $-$ for index i are modula-3 operators. Since, the global and exclusive environments are random; we model each of them using a learning automaton. Since, the global environment is identical for all learning automata, the probability vectors of all learning automata representing the global environment (row 0) and the mechanism for choosing their actions are the same. In order to model the global and exclusive environments, the characteristics of these environments are set as a priori information in action probability vector of their corresponding learning automata. Since, the global and exclusive environments are stationary; the action probability vectors of all learning automata representing global and exclusive environments must be unchanged during the

operation of CLA. Hence, in order to use the model of synchronous CLA to prove the convergence of open CLA, we use the zero value for the learning parameter of learning automata representing the global and exclusive environments.

(2, 1)	(2, 2)	...	(2, i-1)	(2, i)	(2, i+1)	...	(2, n-1)	(2, n)
(1, 1)	(1, 2)	...	(1, i-1)	(1, i)	(1, i+1)	...	(1, n-1)	(1, n)
(0, 1)	(0, 2)	...	(0, i-1)	(0, i)	(0, i+1)	...	(0, n-1)	(0, n)

Figure 3. Equivalent representation of linear open cellular learning automata

Now consider the variation of \mathcal{D} along the solution paths of ODE (17), \mathcal{D} is nondecreasing because

$$\begin{aligned} \frac{d\mathcal{D}}{dt} &= \sum_i \sum_y \frac{\partial \mathcal{D}}{\partial p_{iy}} \frac{\partial p_{iy}}{\partial t}, \\ &= c \sum_i \sum_y \sum_{r>y} p_{iy} p_{ir} [d_{iy}(\underline{p}) - d_{ir}(\underline{p})]^2 \geq 0. \end{aligned} \quad (25)$$

The OSCLA 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 $\mathcal{R}^{m_1 + \dots + m_n}$, asymptotically all solutions of ODE (17) will be in \mathcal{K} . Inequality (25) shows that OSCLA updates the configuration probabilities in gradient ascent manner and hence, converges to a maximum of \mathcal{D} , where $\frac{d\mathcal{D}}{dt} = 0$. From (25), the derivative of \mathcal{D} 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 6 and 7, 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.

Remark 2 If the OSCLA satisfies the sufficiency condition needed for theorem 4, then the OSCLA will converge to a compatible configuration. When the OSCLA doesn't satisfy this sufficiency condition, its convergence to compatible configurations cannot be guaranteed and the OSCLA may exhibit a limit cycle behavior [18].

4. Open Synchronous CLA Using Commutative Rules

In this section, we study the behavior of the OSCLA 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 $\mathcal{F}^i(a_{i+i_1}, a_{i+i_2}, \dots, a_{i+i_n})$ is called commutative if and only if

$$\begin{aligned} \mathcal{F}^i(\alpha_{i+\bar{x}_1}, \dots, \alpha_{i+\bar{x}_m}) &= \mathcal{F}^i(\alpha_{i+\bar{x}_m}, \dots, \alpha_{i+\bar{x}_1}) \\ &= \dots = \mathcal{F}^i(\alpha_{i+\bar{x}_2}, \dots, \alpha_{i+\bar{x}_1}). \end{aligned} \quad (26)$$

In order to simplify the algebraic manipulations, we give the analysis for the linear OSCLA. The linear OSCLA, as shown in figure 4, uses the neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. The following theorem is an additional property for compatible configurations in OSCLA using commutative rules.

1	2	...	$i-1$	i	$i+1$...	$n-1$	n
---	---	-----	-------	-----	-------	-----	-------	-----

Figure 4. The linear OSCLA

Theorem 5 For a OSCLA, 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

$$\left. \frac{d\mathcal{D}(\lambda \underline{q} + (1 - \lambda) \underline{p})}{d\lambda} \right|_{\lambda=0} \leq 0. \quad (27)$$

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 $\nabla \mathcal{D}(\underline{q})$ is denoted by q_{lr} and calculated by the following

equation. Let $\underline{y} = \alpha_{i+\bar{x}_1}$, $\underline{x} = \alpha_{i+\bar{x}_2}$, $\underline{z} = \alpha_{i+\bar{x}_3}$,

$j = i - 1$, $k = i + 1$, $\underline{u} \in \underline{O(G)}$, $\underline{v} \in \underline{O(E^i)}$, \underline{a} be

the index for global environment and \underline{b} be the index for the exclusive environment E^i .

$$\begin{aligned} q_{lr} &= \frac{\partial}{\partial p_{lr}} \mathcal{D}(\underline{q}), \\ &= 5 \sum_{\underline{x}} \sum_{\underline{z}} \sum_{\underline{u}} \sum_{\underline{v}} \mathcal{F}^i(r, x, z, u, v) d_{\underline{u}}^G d_{\underline{v}}^E p_{jz} p_{kz}, \\ &= 5 d_{lr}(\underline{p}), \end{aligned}$$

when cell l represents a cell of cellular learning automata and $q_{lr} = 0$, when the cell l represents global or exclusive

environments. Thus, $\nabla \mathcal{D}(\underline{p})(\underline{q} - \underline{p}) \leq 0$, which implies that

$$\begin{aligned} \nabla \mathcal{D}(\underline{p})(\underline{q} - \underline{p}) &= 5 \sum_i \sum_{\underline{y}} d_{i\underline{y}}(\underline{p}) [q_{i\underline{y}} - p_{i\underline{y}}], \\ &\leq 0. \end{aligned}$$

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.

Remark 3 In general, when the rule of OSCLA is not commutative, local maxima for $\mathcal{D}(\underline{p})$ still exist, but they

may not be compatible.

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

Theorem 6 An open 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, $(\max\{\underline{a}\} \rightarrow 0)$

, always converges to a deterministic configuration, that is stable and compatible.

Proof: Let function $\mathcal{D} : \mathcal{R}^{m_1 + \dots + m_m} \times \underline{O(G)} \times \underline{O(E)} \rightarrow \mathcal{R}$ be the total average reward for the OSCLA. Hence, we have $\frac{\partial \mathcal{D}}{\partial p_{lr}}(\underline{p}) = 5 d_{lr}(\underline{p})$ for all i and r . Using theorem 4

convergence of OSCLA can be concluded.

Remark 4 From the proof of theorem 6, we can conclude that the OSCLA converges to one of its compatible configurations, if any. If the OSCLA has only one compatible configuration, then OSCLA converges to this configuration for which $\mathcal{D}(\underline{p})$ is the maximum. If there are

more than one compatible configurations, then the OSCLA depending on the initial configuration $\underline{p}(0)$ converges to

one of its compatible configurations for which $\mathcal{D}(\underline{p})$ is a local maximum, .

Remark 5 Theorem 6 guarantees that limit cycle for OSCLA does not exist and OSCLA always converges to equilibrium of ODE.

5. Numerical Examples

This section discusses patterns formed by the evolution of open synchronous 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. 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 constitutes a $m^{\bar{m}}$ 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 OSCLA, where j

is a decimal number representing the rule and m is the number of actions of the learning automaton. For example, table 1 represents the rule $(22)_2$ for a linear OSCLA 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 to be output to the learning automata.

Table 1. The scheme for the rule numbering for two actions learning automata

Configuration	111	110	101	100	011	010	001	000
θ	0	0	0	1	0	1	1	0

In the experiments presented below, the OSCLA with neighborhood function $\bar{N}(i) = \{i-1, i\}$ are considered.

Figure 5 shows the time-space diagram evolution of CLA with 20 cells and a two-actions L_{R-I} learning automaton in

each cell. The global environment chooses its actions 1 and 2 with constant probability of 0.6 and 0.4, respectively. The exclusive environments choose their actions with constant probability of 0.3 and 0.7, respectively.

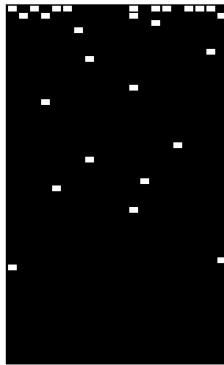


Figure 5. Time-space diagram for open CLA

6. Conclusions

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

References

[1] H. Beigy and M. R. Meybodi, "A Mathematical Framework to Study the Evolution of Cellular Learning

Automata," *Advances in Complex Systems*, vol. 7, no. 3-4, pp. 295-319, 2004.

- [2] H. Beigy and M. R. Meybodi, "A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach," volume 2690 of *Springer-Verlag Lecture Notes in Computer Science*, pages 119-126. Springer-Verlag, 2003.
- [3] H. Beigy and M. R. Meybodi, "Adaptive Uniform Fractional Channel Algorithms," *Iranian Journal of Electrical and Computer Engineering*, 3(1), pp. 47-53, Winter-Spring 2004.
- [4] M. C. Borelijst and P. Hogeweg, "Spiral Wave Structure in Pre-Biotic Evolution: Hypercycles Stable Against Parasites," *Physica D* 48, pp. 17-28, 1991.
- [5] Y. Gunji, "Pigment Color Patterns of *Mulucs* as an Autonomous Process Generated by Asynchronous Automata," *BioSystems* 23, pp. 317-334, 1990.
- [6] O. Inghe, Genet and Ramet, "Survivorship Under Different Mortality Regimes- A Cellular Automata Model," *Journal of Theoretical Biology* 138, pp. 257-270, 1989.
- [7] M. R. Kharazmi and M. R. Meybodi, "Image Segmentation Using Cellular Learning Automata," *Proceedings of 10th Iranian Conference on Electrical Engineering, ICEE-96*, Tabriz, Iran, May 2001.
- [8] M. R. Kharazmi and M. R. Meybodi, "Image estoration Using Cellular Learning Automata," *Proceedings of the Second Iranian Conf. on Machine Vision, Image Processing and Applications*, Tehran, Iran, pp. 261-270, 2003.
- [9] H. J. Kushner, *Approximation and Weak Convergence Methods for Random Processes*. MIT Press, Cambridge, 1984.
- [10] M. R. Meybodi and H. Beigy, "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters," *Neurocomputing* 48(3), pp. 957-974, 2002.
- [11] M. R. Meybodi and H. Beigy, "New Learning utomata Based Algorithms for Adaptation of Backpropagation Algorithm Parameters," *International Journal of Neural Systems* 12(1), pp. 45-68, 2002.
- [12] M. R. Meybodi, H. Beigy, and M. Taherkhani, Cellular Learning Automata and its Applications," *Sharif Journal of Science and Technology*, 19(25), pp. 54-77, 2003.
- [13] M. R. Meybodi and M. R. Kharazmi, "Image estoration Using Cellular Learning Automata," *Proceedings of the First Iranian Conf. on Machine Vision, Image Processing and Applications*, Birjand, Iran, pp. 244-254, 2000.
- [14] M. R. Meybodi and M. R. Kharazmi, "Application of Cellular Learning Automata to Image Processing," *Journal of Amirkabir*, vol. 14, no. 56A, pp. 1101-1126, 2004.

- [15] M. R. Meybodi and M. R. Khojasteh. "Application of Cellular Learning Automata in Modelling of Commerce Networks" *Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC-2001*, Isfahan, Iran, pp. 284-295, 2001.
- [16] M. R. Meybodi and F. Mehdipour, "VLSI Placement Using Cellular Learning Automata," *Proceedings of 8th Annual International Computer Society of Iran Computer Conference CSICC-2001*, Mashad, Iran, pp 195-203, 2003.
- [17] M. R. Meybodi and M. Taherkhani. "Application of Cellular Learning Automata in Modeling of Rumor Diffusion," *Proceedings of 9th Conference on Electrical Engineering, Power and Water Institute of Technology*, Tehran, Iran, pp 102-110, 2001.
- [18] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems*, New York, Printice-Hall International Inc, 1989.
- [19] K. S. Narendra and K. S. Thathachar. *Learning Automata: An Introduction*, New York, Printice-Hall, 1989.
- [20] B. J. Oommen and E. V. de St. Croix, "Graph Partitioning Using Learning Automata". *IEEE Transactions on Computers* 45(2), pp. 195-208, 1996.
- [21] B. J. Oommen and T. D. Roberts, "Continuous Learning Automata Solutions to the Capacity Assignment Problem," *IEEE Transactions on Computers* 49(6), pp. 608-620, 2000.
- [22] P. R. Srikantakumar and K. S. Narendra, "A Learning Model for Routing in Telephone Networks," *SIAM Journal of Control and Optimization* 20(1), pp. 34-57, 1982.



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

Email: beigy@ce.aut.ac.ir



Mohammad Reza Meybodi received the B.S. and M.S. degrees in Economics from Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M.S. and Ph.D. degree from 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 Western Michigan University, and from 1985 to 1991 as an associate professor at Ohio University, USA. His research interests include channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.

Email: Meybodi@ce.aut.ac.ir