Open Synchronous Cellular Learning Automata

Hamid Beigy
Computer Engineering Department, Sharif University of Technology, Tehran, Iran
Institute for Studies in Theoretical Physics and Mathematics (IPM), School of Computer Science, Tehran, Iran
beigy@ce.sharif.edu

M. R. Meybodi
Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
Institute for Studies in Theoretical Physics and Mathematics (IPM), School of Computer Science, Tehran, Iran
mmeybodi@aut.ac.ir

Abstract. Cellular learning automata is a combination of learning automata and cellular automata. This model is superior to cellular learning automata because of its ability to learn and also is superior to single learning automaton because it is a collection of learning automata which can interact together. In some applications such as image processing, a type of cellular learning automata in which the action 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 cellular learning automata. In this paper, we introduce open cellular learning automata and then study its steady state behavior. It is shown that for class of rules called commutative rules, the open cellular learning automata in stationary external environments converges to a stable and compatible configuration. Then the application of this new model to image segmentation has been presented.

Keywords. Cellular Automata, Learning Automata, Cellular Learning Automata, Dynamical Systems.

1 Introduction

In recent years cellular automata (CA) have frequently been used to model the dynamics of spatially extended physical systems. Examples include a wide range of topics, such as periotic evolution [1], the development of pigment patterns in mollusks [2], and growth of clonal plants [3], to mention a few. Cellular automata is a collection of cells that each adapts one of a finite number of states. CA updates the state of each cell by employing 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 evolves 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 (LA) are simple agents for doing simple things. The learning automata have finite set of actions and at 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 L_{R-I}, for updating the action probability vector is given. Let α_i be the action chosen at time k as a sample realization from probability distribution $\underline{p}(k)$. In L_{R-I} algorithm, the action probability vector is updated according to the following rule.

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

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. b $(0 < b < 1)$ represents *learning parameter* and r is the number of actions for LA [4]. LA have been used successfully in many applications such as telephone and data network routing [5], solving NP-Complete problems [6], capacity assignment [7], neural network engineering [8, 9] and call admission in cellular networks [10] to mention a few.

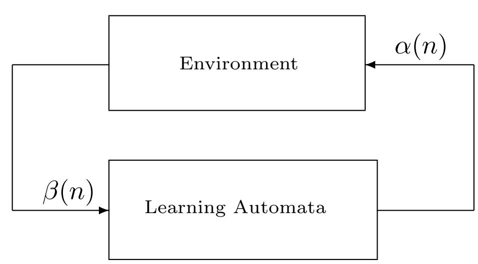


Fig. 1. The interaction of a learning automaton and its environment.

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 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], 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.

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 LA 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 operates 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 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 CLA 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 (Cellular Learning Automata). *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 $\bar{m}$ represents the number of neighboring cells and $\bar{x}_i \in Z^d$. The neighborhood vector determines the relative position*

of the neighboring 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 $\bar{N}(u)$ mapping a cell u to the set of its neighbors, that is

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

For the sake of simplicity, we assume that the first element of neighborhood vector (i.e. $\bar{x}_1$) is equal to d-tuple $(0, 0, \cdots, 0)$. The neighborhood function $\bar{N}(u)$ must satisfy in the two following conditions:
- $u \in \bar{N}(u)$ *for all* $u \in Z^d$.
- $u_1 \in \bar{N}(u_2) \Longleftrightarrow u_2 \in \bar{N}(u_1)$ *for all* $u_1, u_2 \in Z^d$.

5. $\mathcal{F} : \underline{\Phi}^{\bar{m}} \rightarrow \underline{\beta}$ *is the local rule of the cellular learning automata, where $\underline{\beta}$ is the set of values that the reinforcement signal can take. It gives the 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 [13–17], modeling of commerce networks [18], fixed channel assignment in cellular networks [19], and VLSI Placement [20] to mention a few. 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 [21], a mathematical methodology to study the steady state 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 image processing, a type of cellular learning automata in which the action 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 synchronous cellular learning automata* (OCLA). In this paper, we introduce OSCLA in which all cells are updated synchronously and study its steady state behavior. It is shown that for a class of rules called commutative rules, the OSCLA converges to a globally stable state in stationary external environments. Then an applications of OSCLA to image processing has been presented.

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

2 Open Synchronous Cellular Learning Automata

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 synchronous cellular learning automata* (OSCLA) is introduced, in which the evolution of CLA is influenced by the external environments. Two types of environments can be considered in the OSCLA: global environment and exclusive environment. Each CLA has one global environment that influences all cells and an exclusive environment for each particular cell influencing the evolution of that cell. The interconnection of a typical cell in the OSCLA and its various types of environments is shown in figure 2.

Formally, a $d-$dimensional open synchronous cellular learning automata is defined as follows.

Definition 2 (Open Synchronous Cellular Learning Automata). *A $d-$dimensional open synchronous cellular learning automata is a structure $\mathcal{A} = (Z^d, \Phi, A, E^G, E^E, N, \mathcal{F})$, 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. E^G *is the global environment.*
5. $E^E = \{E_1^E, E_2^E, \ldots, E_n^E\}$ *is the set of exclusive environments, where E_i^E is the exclusive environment for learning automaton residing in cell i.*
6. $N = \{\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_{\bar{m}}\}$ *is neighborhood vector.*
7. $\mathcal{F} : \underline{\Phi}^{\bar{m}} \times \underline{O}(G) \times \underline{O}(E) \rightarrow \underline{\beta}$ *is the local rule of the cellular automata, where $\underline{O}(G)$ and $\underline{O}(E)$ is the set of reinforcement signals of global and exclusive environments, respectively.*

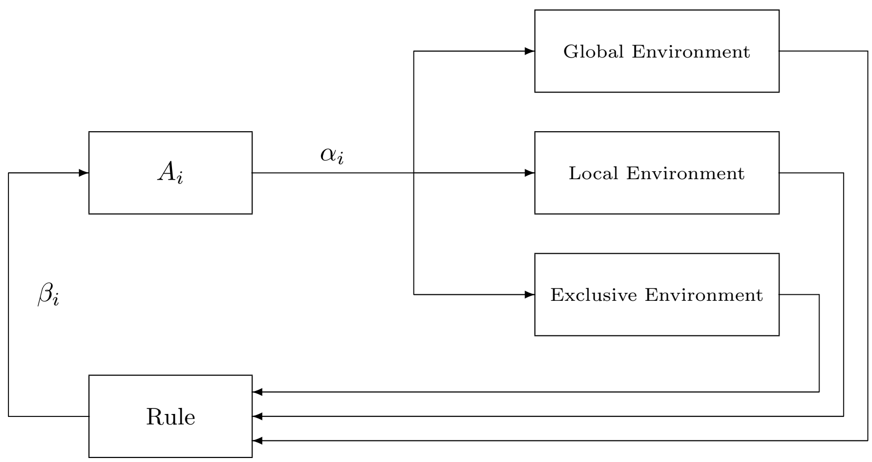


Fig. 2. The interconnection of a typical cell in OSCLA with its various environments.

In what follows, we consider an OSCLA 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 $i = 1, \ldots, n$) of the OSCLA. Let cardinality of $\underline{\alpha}_i$ be m_i and the state of the OSCLA represented by $\underline{p} = (\underline{p}_1', \underline{p}_2', \ldots, \underline{p}_n')'$, where $\underline{p}_i = (p_{i1}, \ldots, p_{im_i})'$ is the action probability vector of automaton A_i.

The operation of OSCLA 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 environment (neighboring learning automata) as well as the global environment and their corresponding exclusive environments. Each environment produces a signal, which are used by the local rule to generate a reinforcement signal to the learning automaton residing in every cell. The higher values of β_i means that the chosen action of automaton A_i is more rewarded, where β_i is the reinforcement signal of automaton A_i. 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 this paper, we assume that the global and exclusive environments are stationary.

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 to study the steady state behavior of OSCLA.

Definition 3. *A configuration of OSCLA is a mapping $\mathcal{K} : Z^d \rightarrow \underline{p}$ 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 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 OSCLA are*

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

and

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

respectively.

Note that, the set of probabilistic configurations $\mathcal{K}$ is the convex hull of the set of all deterministic configurations $\mathcal{K}^*$ [21]. 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 $\mathcal{G} : \mathcal{K} \to \mathcal{K}$ that describes the dynamics of OSCLA.*

Definition 6. *The evolution of 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 in a OSCLA with configuration $\underline{p} \in \mathcal{K}$ is defined as*

$$d_{ir}(\underline{p}) = \sum_{y_2} \cdots \sum_{y_{\bar{m}}} \sum_{z\in O(G)} \sum_{w\in O(E_i)} \mathcal{F}^i(r, y_2, \ldots, y_{\bar{m}}, z, w) d_z^G d_w^E \prod_{\substack{l\in\bar{N}(i)\\ l\neq i}} p_{ly_l}, \tag{3}$$

where z and w 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

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

The above definition implies that if 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 providing*

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

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. *A configuration $\underline{p} \in \mathcal{K}$ is admissible providing*

$$D_i(\underline{p}) \geq D_i(\underline{q}) \tag{6}$$

for all configurations $\underline{q} \in \mathcal{K}$ and all cells i.

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

Corollary 1. *Admissibility implies compatibility but the converse is not true, i.e, every admissible configuration is compatible but every compatible configuration is not necessarily admissible.*

Proof: The proof is trivial from definitions 8 and 9. ∎

Definition 10. *Total average reward for the OSCLA, which is the sum of the average reward for all the learning automata in the OSCLA, for configuration $\underline{p} \in \mathcal{K}$ is defined as*

$$\mathcal{D}(\underline{p}) = \sum_i D_i(\underline{p}). \tag{7}$$

Corollary 2. *A configuration $\underline{p} \in \mathcal{K}$ is admissible if and only if $\mathcal{D}(\underline{p}) \geq \mathcal{D}(\underline{q})$ for all $\underline{q} \in \mathcal{K}$.*

Proof: The proof is trivial by looking definitions 9 and 10. ■

Remark 1. With the approach given in this paper, the probability of different configurations are updated according to a learning algorithm that can be considered as hill-climbing in probability space. At every stage, the change in probabilities are such that total average reward is improved monotonically in expected sense.

Lemma 1. *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 $i = 1, \ldots, n$ and $r = 1, \ldots, m_i$. Note that $\psi_{ir}(\underline{p})$ and $\phi_{ir}(\underline{p})$ are continuous functions on $\mathcal{K}$. Then introducing the following transformation

$$\bar{p}_{ir} = \frac{p_{ir} + \phi_{ir}}{1 + \sum_{j=1}^{m} \phi_{ij}} \tag{8}$$

for $i = 1, \ldots, n$ and $r = 1, \ldots, m_i$ and denoting it by mapping $T : \mathcal{K} \to \mathcal{K}$ given by

$$\underline{\bar{p}} = T(\underline{p}). \tag{9}$$

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 OSCLA and conversely every compatible configuration is a fixed point of T, thereby concluding the proof of the theorem. We first verify the latter assertion: if $\underline{p} \in \mathcal{K}$ is a compatible configuration, then by definition 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 ($i = 1, \ldots, n$). Thus we obtain $\psi_{ir_i}(\underline{p}) \leq 0$. Hence, $\phi_{ir_i} = 0$ for all $i = 1, \ldots, n$ and $r_i = 1, \ldots, m_i$ 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

$$\sum_r d_{ir}(\underline{p}) p_{ir} < \sum_r d_{ir}(\underline{p}) \tilde{p}_{ir} \tag{10}$$

Let y_i ($1 \leq i \leq m_i$) be an action for which the average reward for automaton A_i attains its maximum value. Then $D_i(\underline{\tilde{p}})$ can be bounded from above by $d_{iy_i}(\underline{p})$, thus implies $\psi_{iy_i}(\underline{p}) > 0$, which implies $\phi_{iy_i}(\underline{p}) > 0$. 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 the average reward for automaton A_i attains its minimum value. Then by using inequality (10), it can be shown that $D_i(\underline{p})$ bounded below by $d_{ir_i}(\underline{p})$. This implies $\psi_{ir_i}(\underline{p}) < 0$, which implies $\phi_{ir_i}(\underline{p}) = 0$, which when used in (8) yield the conclusion $\bar{p}_{ir_i} < \tilde{p}_{ir_i}$, because $\sum_j \phi_{ij}(\underline{p}) > 0$. But this 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 $D_i(\underline{p}) \geq D_i(\underline{q})$. Since, $\underline{q}$ includes $\underline{q} = (\underline{p}'_1, \ldots, \underline{e}'_{r_i}, \ldots, \underline{p}'_n)'$ for fixed i ($i = 1, \ldots, n$), 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 $r_i = 1, \ldots, m_i$) but $\underline{p}$ is not compatible. Then for some learning automaton A_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 $d_{iy_i}(\underline{p})$ 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 $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$ (for all $r_i = 1, \ldots, m_i$), 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

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

The above contradiction completes the proof of the lemma. ■

This lemma provides a means of finding compatible configurations in the OSCLA.

Theorem 1. *A configuration $\underline{p} \in \mathcal{K}$ is compatible if and only if $\sum_i \sum_y d_{iy}(\underline{p}) \left[p_{iy} - q_{iy}\right] \geq 0$ 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 $\underline{q} = (\underline{p}_1', \ldots, \underline{q}_l', \ldots, \underline{p}_n')'$, we obtain

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

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 learning automata except the specific A_i are held fixed, then the configuration reached by the OSCLA at the point, where the average reward of A_i is maximum, is compatible.

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

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

for all $z \in \underline{O}(G), w \in \underline{O}(E^i)$, $i = 1, \ldots, n$ and $r_i \neq t_i$.

Proof: We first show that if $\underline{p}$ is compatible then $\mathcal{F}^i(t^i, t_2, \ldots, t_{\bar{m}}, z, w) \geq \mathcal{F}^i(r^i, t_2, \ldots, t_{\bar{m}}, z, w)$. In order to show this assertion, we assume that $\underline{q} = \left(\underline{e}_{t^1}', \underline{e}_{t^2}', \ldots, \underline{e}_{r^i}', \ldots, \underline{e}_{t^n}'\right)'$ for $r^i \neq t^i$ is not a compatible corner. From definition 8, we have

$$\sum_r d_{ir}(\underline{p}) p_{ir} \geq \sum_r d_{ir}(\underline{p}) q_{ir}. \tag{11}$$

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

$$d_{it^i}(\underline{p}) \geq d_{ir^i}(\underline{p}). \tag{12}$$

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

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

which concluding this assertion.

Conversely, assume that $\mathcal{F}^i(t^i, t_2, \ldots, t_{\bar{m}}, z, w) \geq \mathcal{F}^i(r^i, t_2, \ldots, t_{\bar{m}}, z, w)$ but $\underline{p}$ is not compatible. Multiplying both sides of the above inequality by $d_z^G d_w^E \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l}$ and summing over all actions of neighboring automata of cell i, we obtain

$$\sum_{t_2} \cdots \sum_{t_{\bar{m}}} \sum_{z} \sum_{w} \mathcal{F}^i(t^i, t_2, \ldots, t_{\bar{m}}, z, w) d_z^G d_w^E \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l} \geq \sum_{t_2} \cdots \sum_{t_{\bar{m}}} \sum_{z} \sum_{w} \mathcal{F}^i(r^i, t_2, \ldots, t_{\bar{m}}, z, w) d_z^G d_w^E \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l}.$$

Using the above inequality and the definition 7, we obtain $d_{it^i}(\underline{p}) \geq d_{ir^i}(\underline{p})$. Since $\underline{p}$ and $\underline{q}$ are two corners, hence multiplying both sides of the above inequality by one and adding zero doesn't change the value of the above inequality, thus we obtain

$$d_{it^i}(\underline{p}) p_{it^i} + \sum_{r \neq t^i} d_{ir}(\underline{p}) p_{ir} \geq d_{ir^i}(\underline{p}) q_{ir^i} + \sum_{r \neq r^i} d_{ir}(\underline{p}) q_{ir}.$$

Simplifying the above inequality, we obtain $\sum_r d_{ir}(\underline{p}) p_{ir} \geq \sum_r d_{ir}(\underline{p}) q_{ir}$, which contradicts the assumption that $\underline{p}$ is not compatible. ■

Corollary 3. *A corner $\underline{p} = \left(\underline{e}'_{t_1}, \underline{e}'_{t_2}, \ldots, \underline{e}'_{t_n}\right)'$ is fully compatible if and only if $\mathcal{F}^i(t_1, t_2, \ldots, t_{\bar{m}}, z, w) > \mathcal{F}^i(r, t_2, \ldots, t_{\bar{m}}, z, w)$ for all $z \in \underline{O}(G), w \in \underline{O}(E^i)$, $i = 1, \ldots, n$ and $r \neq t_i$.*

Proof: The proof is trivial from the proof of Theorem 2. ■

3 Behavior of Open Synchronous Cellular Learning Automata

In this section, we analyze the OSCLA in which all learning automata use the L_{R-I} learning algorithm and operates under stationary global and exclusive environments. Using L_{R-I} learning algorithm, process $\{\underline{p}(k)\}_{k \geq 0}$ is Markovian and can be described by the following difference equation.

$$\underline{p}(k+1) = \underline{p}(k) + \underline{\Lambda} \underline{g}(\underline{p}(k), \underline{\beta}(k)), \tag{13}$$

where $\underline{\beta}(k)$ is composed of components $\beta_{iy}(k)$ (for $1 \leq i \leq n$ and $1 \leq y \leq m_i$), which are dependent on $\underline{p}$. $\underline{g}$ represents the learning algorithm, $\underline{\Lambda}$ is a $M \times M$ diagonal matrix with $\lambda_{jj} = b_i$ for $\sum_{l=1}^{i-1} m_l < i \leq \sum_{l=1}^{i} m_l$, and b_i represents the learning parameter for learning automaton A_i. Let $\underline{B} = (b_1, \ldots, b_n)'$ to denote the learning parameter of learning automata in OSCLA. Now, define

$$\Delta \underline{p}(k) = \mathrm{E}\left[\underline{p}(k+1) | \underline{p}(k)\right] - \underline{p}(k). \tag{14}$$

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

$$\Delta \underline{p}(k) = \underline{\Lambda} \underline{f}\left(\underline{p}(k)\right). \tag{15}$$

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

$$\Delta p_{iy}(k) = b_i p_{iy}(k) \left[1 - p_{iy}(k)\right] \mathrm{E}\left[\beta_{iy}(k)\right] - b_i \sum_{r \neq y} p_{ir}(k) p_{iy}(k) \mathrm{E}\left[\beta_{ir}(k)\right],$$

$$
\begin{aligned}
&= b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) \mathrm{E}\left[\beta_{iy}(k)\right] - b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) \mathrm{E}\left[\beta_{ir}(k)\right], \\
&= b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) \left\{ \mathrm{E}\left[\beta_{iy}(k)\right] - \mathrm{E}\left[\beta_{ir}(k)\right] \right\}, \\
&= b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right], \\
&= b_i f_{iy}(\underline{p}),
\end{aligned} \tag{16}
$$

where

$$
\begin{aligned}
f_{iy}(\underline{p}) &= p_{iy}(k) \sum_{r \neq y} p_{ir}(k) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right], \\
&= p_{iy}(k) \sum_{r} p_{ir}(k) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right], \\
&= p_{iy}(k) \left[d_{iy}(\underline{p}) - D_i(\underline{p}) \right].
\end{aligned} \tag{17}
$$

For different values of $\underline{B}$, equation (13) generates different process and we shall use $\underline{p}^B(k)$ to denote this process whenever the value of $\underline{B}$ is to be specified explicitly. Define a sequence of continuous-time interpolation of $\underline{p}^B(k)$, denoted by $\underline{\tilde{p}}^B(t)$ and called *interpolated process*, whose components are defined by

$$
\underline{\tilde{p}}_i^{\,B}(t) = \underline{p}_i(k) \qquad\qquad t \in [kb_i, (k+1)b_i]. \tag{18}
$$

The objective is to study the limit of sequence $\{\underline{\tilde{p}}^a(t)\}_{t \geq 0}$ as $\max\{b_1, \ldots, b_n\} \to 0$, which will be a good approximation to the asymptotic behavior of (18). When learning parameter b_i is sufficiently small for all $i = 1, 2, \ldots, n$, then equation (15) can be written as the following ordinary differential equation (ODE).

$$
\underline{\dot{p}} = \underline{f}(\underline{p}), \tag{19}
$$

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

$$
\frac{dp_{iy}}{dt} = p_{iy} \left[d_{iy}(\underline{p}) - D_i(\underline{p}) \right]. \tag{20}
$$

We are interested in characterizing the long term behavior of $\underline{p}(k)$ and hence the asymptotic behavior of ODE (19). The analysis of process $\{\underline{p}(k)\}_{k \geq 0}$ is done in two stages. In the first stage, we solve ODE (19) and in the second stage, we characterize the solution of this ODE. The solution of ODE (19) approximates the asymptotic behavior of $\underline{p}(k)$ and the characteristics of this solution specify the long term behavior of $\underline{p}(k)$. The following theorem gives the asymptotic behavior of $\underline{\tilde{p}}^B$ as $\max\{b_1, b_2, \ldots, b_n\}$ is sufficiently small. We show that the sequence of interpolated process $\{\underline{\tilde{p}}^B(t)\}$ converges weakly to the solution of ODE (19) with initial configuration $\underline{p}(0)$. This implies that asymptotic behavior of $\underline{p}(k)$ can be obtained from the solution of ODE (19).

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

$$
\frac{d\underline{X}}{dt} = \underline{f}(\underline{X}) \tag{21}
$$

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

Proof: The following conditions are satisfied by the learning algorithm (13).

1. $\{\underline{p}(k), (\underline{\alpha}(k-1), \underline{\beta}(k-1))\}_{k \geq 0}$ is a Markov process.
2. $(\underline{\alpha}(k), \underline{\beta}(k))$ takes values in a compact metric space.
3. $\underline{g}$ is bounded, continuous and independent of $\underline{B}$.
4. ODE (21) has a unique solution for each initial condition $\underline{X}(0)$.

5. If $\underline{p}(k) = \underline{\bar{p}}$ is a constant, then $\{(\underline{\alpha}(k), \underline{\beta}(k))\}_{k\geq 0}$ is an independent identically distributed sequence. Let $M^{\bar{p}}$ be the distribution of process $\{(\underline{\alpha}(k), \underline{\beta}(k))\}_{k\geq 0}$.

Then using the weak convergence theorem [22], sequence $\{\underline{\tilde{p}}^B(.)\}$ converges weakly, as $\max\{b_1, \ldots, b_n\} \to 0$ to the solution of

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

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

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

and hence the theorem. ■

Theorem 3 enables us to understand the long term behavior of $\underline{p}(k)$. The weak convergence in this theorem implies that path $\underline{p}^B(t)$ will closely follow the solution to the ODE on any finite interval with an arbitrarily high probability as $\max\{b_1, \ldots, b_n\} \to 0$. As the length of the time interval increases and $\max\{b_1, \ldots, b_n\} \to 0$, the fraction of time that the path of the ODE must eventually spend in a small neighborhood of $\underline{p}^o$, the solution of the ODE, goes to one. Thus, $\underline{p}^B(.)$ 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 OSCLA follows the path of the ODE goes to infinity as $\max\{b_1, \ldots, b_n\} \to 0$. Although the speed of convergence depends on the specific value of $\underline{B}$. The above point is summarized in the following lemma.

Lemma 4. *For large k and small enough value of* $\max\{b_1, \ldots, b_n\}$*, the asymptotic behavior of $\underline{p}(k)$ generated by the OSCLA can be well approximated by the solution to ODE (21) with the same initial configuration.*

Proof: Let $\underline{X}(.)$ be the solution of ODE (21) 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 $\mathrm{E}\left[h_T(\underline{\tilde{p}})^B\right] \to \mathrm{E}\left[h_T(\underline{X})\right] = 0$ as $\max\{b_1, \ldots, b_n\} \to 0$. The limit is zero since the value of $h_T(\underline{X})$ on the paths of limit process is zero with probability one. Thus, the sup over $t \in [0, T]$ of the distance between the original sequence $\underline{p}(t)$ and $\underline{X}(t)$ goes to zero in probability as $k \to \infty$. With particular initial condition used, let $\underline{p}^o$ be the equilibrium configuration to which the solution of the ODE converges. Using this and the nature of interpolation, given in (18), it is implied that for the given initial configuration and any $\epsilon > 0$ and integers k_1 and k_2 $(0 < k_1 < k_2 < \infty)$, there exists a b_0 such that

$$\mathrm{Prob}\left[\sup_{k_1\leq k\leq k_2} \left|\left|\underline{p}(k) - \underline{p}^o\right|\right| > \epsilon\right] = 0 \qquad \forall \max\{b_1, \ldots, b_n\} < b_0.$$

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 ACLA converges to $\underline{p}^o$. ■

In the following subsections, we first find the equilibrium points of ODE (19), then study the stability property of equilibrium points of ODE (19), 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 $\underline{f}\left(\underline{p}\right)$, 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*

$$d_{iy}(\underline{p}) = d_{ir}(\underline{p}), \tag{22}$$

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

Proof: From equation (16), it is obvious that f_{iy} (for $i = 1, 2, \ldots, n$ and $y = 1, 2, \ldots, m_i$) is zero if $\underline{p}_i$ is a unit vector and hence all corners of $\mathcal{K}$ are equilibrium points of $\underline{f}(.)$. To find other zeros of $\underline{f}(.)$, it is obvious from (16) that $f_{iy} = 0$ if $p_{iy} = 0$. But $\underline{p}_i$ is a probability vector, and all components of $\underline{p}_i$ can not be zero at the same time. Hence, when $p_{iy} \neq 0$, we must have, for f_{iy} to be zero

$$\sum_{r \neq y} p_{ir}(k) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right] = 0 \tag{23}$$

The above equation can be rewritten as

$$\begin{aligned}
\sum_{r \neq y} p_{ir}(k) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right] &= \sum_{r \neq y} p_{ir}(k) d_{iy}(\underline{p}) - \sum_{r \neq y} p_{ir}(k) d_{ir}(\underline{p}), \\
&= d_{iy}(\underline{p}) \left[1 - p_{iy}(k) \right] - \sum_{r \neq y} p_{ir}(k) d_{ir}(\underline{p}), \\
&= d_{iy}(\underline{p}) - \sum_{r} p_{ir}(k) d_{ir}(\underline{p}), \\
&= d_{iy}(\underline{p}) - \sum_{r \neq q} p_{ir}(k) d_{ir}(\underline{p}) - p_{iq}(k) d_{iq}(\underline{p}), \\
&= d_{iy}(\underline{p}) - \sum_{r \neq q} p_{ir}(k) d_{ir}(\underline{p}) - d_{iq}(\underline{p}) \left[1 - \sum_{r \neq q} p_{ir}(k) \right], \\
&= d_{iy}(\underline{p}) - d_{iq}(\underline{p}) + \sum_{r \neq q} \left[d_{iq}(\underline{p}) - d_{ir}(\underline{p}) \right] p_{ir}(k). \qquad (24)
\end{aligned}$$

Thus, we obtain

$$\sum_{r \neq q} \left[d_{iq}(\underline{p}) - d_{ir}(\underline{p}) \right] p_{ir}(k) = d_{iq}(\underline{p}) - d_{iy}(\underline{p}), \tag{25}$$

for all $y = 1, \ldots, m_i$ and all $y \neq q$. 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

$$d_{iq}(\underline{p}) - d_{i1}(\underline{p}) = d_{iq}(\underline{p}) - d_{i2}(\underline{p}) = d_{iq}(\underline{p}) - d_{i3}(\underline{p}) = \ldots = d_{iq}(\underline{p}) - d_{im_i}(\underline{p}) = d_0.$$

When $d_0 \neq 0$, equation (25) 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 (23) 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. ■

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 (19). From lemmas 5 and 6, all the equilibrium points of (19) are known. In order to study the stability of the equilibrium points, 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 (19).

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

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

the origin is translated to $\underline{p}^0$. Since $\underline{p}_i$ $(1 \leq i \leq n)$ 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

$$f_{iy} = \tilde{p}_{iy}[\mathcal{F}^i(y, t_2, \ldots, t_{\bar{m}}, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \ldots, t_{\bar{m}}, t_g, t_e)] + \text{high order terms}, \tag{26}$$

where the t_g and t_w are the response of the global and exclusive environments. 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 $\mathcal{F}^i(y, t_2, \ldots, t_{\bar{m}}, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \ldots, t_{\bar{m}}, t_g, t_e) < 0$ for $i = 1, 2, \ldots, n$. Thus, equation (26) 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 (19) can be written as $\dot{\underline{\tilde{p}}} = A\underline{\tilde{p}}$, where $A = \text{diag}\left(\tilde{f}_{iy}\right)$ and $\tilde{f}_{iy} = \mathcal{F}^i(y, t_2, \ldots, t_{\bar{m}}, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \ldots, t_{\bar{m}}, t_g, t_e)$ for $i = 1, 2, \ldots, n$. 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. *Incompatible equilibrium points of $\underline{f}(.)$ are unstable.*

Proof: Let $\underline{p}^o$ be an equilibrium point of $\underline{f}(.)$ which is not compatible. Then by lemma 2, 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 (20), 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 2. In lemmas 7 and 8, the solution of ODE (19) 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 $\mathcal{K}$.

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

2. $\underline{p}(0)$ is close to an incompatible corner $\underline{p}^o$. By lemma 8, no matter how small neighborhood we take around $\underline{p}^o$, the solution of (19) 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 [4], 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 is stated in theorem 4.

Theorem 4. *Suppose there is a bounded differential function* $\mathcal{D} : \mathcal{R}^{m_1+\cdots+m_{\bar{m}}+m_g+m_e} \to \mathcal{R}$ *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, *where* m_g *and* m_e *are the cardinality of the outputs of the global and exclusive environments, respectively. Then OSCLA for any initial configuration in* $\mathcal{K}-\mathcal{K}^*$ *and with sufficiently small value of learning parameter* $(\max\{b_1, b_2, \ldots, b_n\} \to 0)$, *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 CLA with n cells and neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. For the global environment, we add an extra row to this CLA, 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 CLA 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 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 OSCLA, we use the zero value for the learning parameter of learning automata representing the global and exclusive environments.

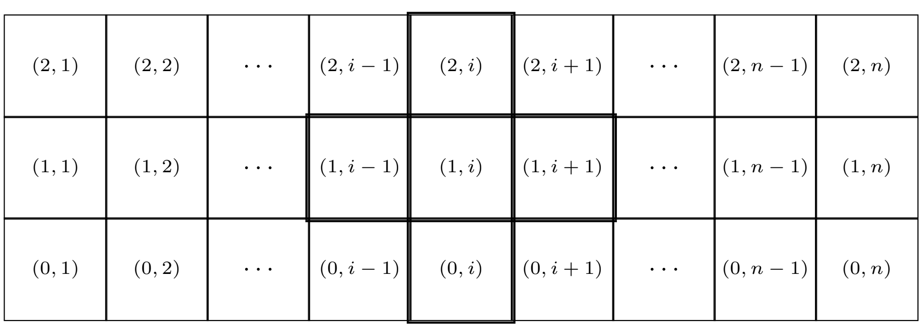


Fig. 3. Equivalent representation of a linear open cellular learning automata.

Now consider the variation of $\mathcal{D}$ along the solution paths of ODE (19), $\mathcal{D}$ is nondecreasing because

$$\begin{aligned}\frac{d\mathcal{D}}{dt} &= \sum_i \sum_y \frac{\partial \mathcal{D}}{\partial p_{iy}} \frac{\partial p_{iy}}{\partial t}, \\ &= \sum_i \sum_y \frac{\partial \mathcal{D}}{\partial p_{iy}} p_{iy} \sum_r p_{ir} \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right],\end{aligned}$$

$$
\begin{aligned}
&= c \sum_i \sum_y \sum_r p_{iy} p_{ir} d_{iy}(\underline{p}) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right], \\
&= c \sum_i \sum_y \left(\sum_{r>y} p_{iy} p_{ir} d_{iy}(\underline{p}) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right] + \sum_{r<y} p_{iy} p_{ir} d_{iy}(\underline{p}) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right] \right), \\
&= c \sum_i \sum_y \left(\sum_{r>y} p_{iy} p_{ir} d_{iy}(\underline{p}) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right] + \sum_{r>y} p_{ir} p_{iy} d_{ir}(\underline{p}) \left[d_{ir}(\underline{p}) - d_{iy}(\underline{p}) \right] \right), \\
&= c \sum_i \sum_y \sum_{r>y} p_{iy} p_{ir} \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p}) \right]^2 . \\
&\geq 0.
\end{aligned}
\tag{27}
$$

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 $\mathcal{R}^{m_1+\cdots+m_{\bar{m}}+m_g+m_e}$, asymptotically all solutions of ODE (19) will be in $\mathcal{K}$. Inequality (27) 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 (27), the derivative of $\mathcal{D}$ is zero if and only if for all i, y, r, we have $p_{ir} p_{iy} = 0$ or $d_{iy}(\underline{p}) = d_{ir}(\underline{p})$. From lemmas 5 and 6, these configurations are equilibrium points of $f_{iy}(\underline{p})$. Thus the solution to ODE (19) for any initial configuration in $\mathcal{K} - \mathcal{K}^*$ will converge to a set containing only equilibrium points of the ODE (19). Since all equilibrium configurations that are not compatible are unstable, the theorem follows. ■

Remark 3. If the OSCLA satisfies the sufficiency conditions needed for Theorem 4, then OSCLA will converge to a compatible configuration. When the OSCLA doesn't satisfy this sufficiency condition, convergence to compatible configurations cannot be guaranteed and the OSCLA may exhibit a limit cycle behavior [23].

4 Open Synchronous Cellular Learning Automata 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 11 (Commutative Rule). *A rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}})$ is called commutative if and only if*

$$
\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) = \mathcal{F}^i(\alpha_{i+\bar{x}_{\bar{m}}}, \alpha_{i+\bar{x}_1}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) = \ldots = \mathcal{F}^i(\alpha_{i+\bar{x}_2}, \alpha_{i+\bar{x}_3}, \ldots, \alpha_{i+\bar{x}_1}). \tag{28}
$$

In order to simplify the algebraic manipulations, we give the analysis for 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	$\cdots$	$i-1$	i	$i+1$	$\cdots$	$n-1$	n

Fig. 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, then $\underline{p}$ is compatible.*

Proof: Since $\mathcal{K}$ is convex [21], 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 that $\mathcal{D}(\underline{p})$ is local maximum, then $\mathcal{D}(\underline{p})$ doesn't increase as one moves away from $\underline{p}$. Thus we have

$$\left.\frac{d\mathcal{D}(\lambda \underline{q} + (1-\lambda)\underline{p})}{d\lambda}\right|_{\lambda=0} \leq 0. \tag{29}$$

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 $\nabla D(\underline{q})$ is denoted by q_{lr} and calculated by the following equation. Let $y = \alpha_{i+\bar{x}_1}$, $x = \alpha_{i+\bar{x}_2}$, $z = \alpha_{i+\bar{x}_3}$, $j = i-1$, $k = i+1$, $u \in \underline{O(G)}$, $v \in \underline{O(E^i)}$, a be the index for global environment and b be the index for the exclusive environment E^i.

$$\begin{aligned}
q_{lr} &= \frac{\partial}{\partial p_{lr}} \sum_i \sum_y \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(y,x,z,u,v) d_u^G d_v^E p_{jx} p_{iy} p_{kz}, \\
&= \sum_i \sum_x \sum_y \sum_z \sum_u \sum_v \left[\mathcal{F}^i(y,x,z,u,v)\delta_{lj}\delta_{rx} d_u^G d_v^E p_{iy} p_{kz} + \mathcal{F}^i(y,x,z,u,v)\delta_{li}\delta_{ry} d_u^G d_v^E p_{jx} p_{kz} \right. \\
&\quad \left. + \mathcal{F}^i(y,x,z,u,v)\delta_{lk}\delta_{rz} d_u^G d_v^E p_{jx} p_{iy} + \mathcal{F}^i(y,x,z,u,v)\delta_{la}\delta_{ru} d_u^G d_v^E p_{iy} p_{kz} + \mathcal{F}^i(y,x,z,u,v)\delta_{lb}\delta_{rv} d_u^G d_v^E p_{jx} p_{kz} \right], \\
&= \sum_y \sum_z \sum_u \sum_v \mathcal{F}^i(y,r,z,u,v) d_u^G d_v^E p_{iy} p_{kz} + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r,x,z,u,v) d_u^G d_v^E p_{jx} p_{kz} \\
&\quad + + \sum_x \sum_y \sum_u \sum_v \mathcal{F}^i(y,x,r,u,v) d_u^G d_v^E p_{jx} p_{iy} + \sum_x \sum_y \sum_z \sum_v \mathcal{F}^i(y,x,z,r,v) d_v^E p_{iy} p_{jx} p_{kz} \\
&\quad + \sum_x \sum_y \sum_z \sum_u \mathcal{F}^i(y,x,z,u,r) d_u^G p_{iy} p_{jx} p_{kz}, \\
&= \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r,x,z,u,v) d_u^G d_v^E p_{jx} p_{kz} + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r,x,z,u,v) d_u^G d_v^E p_{jx} p_{kz} \\
&\quad + + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r,z,x,u,v) d_u^G d_v^E p_{jx} p_{kz} + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r,x,z,u,v) d_u^G d_v^E p_{jx} p_{kz} \\
&\quad + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r,x,z,u,v) d_u^G d_v^E p_{jx} p_{kz}, \\
&= 5 \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r,x,z,u,v) d_u^G d_v^E p_{jx} p_{kz}, \\
&= 5 d_{lr}(\underline{p}),
\end{aligned}$$

when the cell l represents a cell of cellular learning automata and $q_{lr} = 0$, when the cell l represents global or exclusive environments. Thus, $\nabla\mathcal{D}(\underline{p})(\underline{q}-\underline{p}) \leq 0$ implies that

$$\begin{aligned}
\nabla\mathcal{D}(\underline{p})(\underline{q}-\underline{p}) &= 5 \sum_i \sum_y \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(y,x,z,u,v) d_u^G d_v^E p_{jx} p_{kz} \left[q_{iy} - p_{iy}\right], \\
&= 5 \sum_i \sum_y d_{iy}(\underline{p}) \left[q_{iy} - p_{iy}\right], \\
&\leq 0.
\end{aligned}$$

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

Remark 4. In general, when rules of OSCLA are 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 OSCLA, which uses uniform and commutative rule, starting from* $\underline{p}(0) \in \mathcal{K} - \mathcal{K}^*$ *and with sufficiently small value of learning parameter,* $(\max\{b_1, \ldots, b_n\} \to 0)$*, always converges to a deterministic configuration, that is stable and also compatible.*

Proof: Let function $\mathcal{D}: \mathcal{R}^{m_1+\cdots+m_{\bar{m}}+m_g+m_e} \rightarrow \mathcal{R}$ be the total average reward for the OSCLA. Hence, we have $\frac{\partial \mathcal{D}}{\partial p_{ir}}(\underline{p}) = 5d_{ir}(\underline{p})$ for all i and r. Using theorem 4 convergence of OSCLA can be concluded. ■

Remark 5. From the proof of the Theorem 6, we can conclude that the OSCLA converges to one of its compatible configurations, if any. If this compatible configuration is unique, 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 $\underline{p}(0)$ may converge to one of its compatible configurations for which $\mathcal{D}(\underline{p})$ is a local maximum, .

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

5 Computer Experiments

In this section, we give two computer experiments: 1) patterns formed by the evolution of CLA from random initial configuration, and 2) image segmentation.

5.1 Numerical Examples

This section discusses patterns formed by the evolution of cellular learning automata from random initial configuration, chosen by the learning automata in cells. Different cellular learning automata rules are found to yield different configurations. For the sake of simplicity in our presentation, we use the following notation to specify the rules for OSCLA for which each cell has a learning automaton with m actions. The actions of each learning automaton are represented by integers in interval $[0, m-1]$. Hence, the configuration of each cell and its neighbors forms a $\bar{m}$ digits number in interval $[0, m^{\bar{m}}-1]$ with $m^{\bar{m}}$ possible values. The value of reinforcement signal for all of the above $m^{\bar{m}}$ configurations constitute an $m^{\bar{m}}$ bit number. Then the rule identified by decimal representation of this $m^{\bar{m}}$-bit number. For the sake of simplicity in our presentation, we use notation $(j)_m$ to specify the rules in the OSCLA, where j is a decimal representing the rule and m is the number of actions for each learning automaton. For example, the following table represents the rule 22 for a linear OSCLA with two-actions learning automata and represented by $(22)_2$. Each of the eight possible sets configuration for a cell and its neighbors appear on the upper row, while the lower row gives the value of the reinforcement signal to be taken to the central cell on the next time step.

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

Configuration	111	110	101	100	011	010	001	000
Reinforcement Signal	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.

The simulation results show that the OSCLA converges to a configuration in $\mathcal{K}^*$ rather than to a configuration in $\mathcal{K} - \mathcal{K}^*$.

5.2 Image Segmentation

The ultimate aim in a large number of image processing applications, like medical image analysis, optical character recognition, industrial automation, and photo editing, is to extract important features from rule $(32768)_2$

Fig. 5. Time-space diagram for OSCLA

image data from which a description, interpretation, or understanding of the scene can be provided by machine. In such applications, image analysis basically involves the study of feature extraction, segmentation, and classification techniques. In machine analysis of image systems, the image is first preprocessed and then certain features are extracted to decompose the image into its components and finally assign the components or segmented parts into one of several objects, each identified by a label. Image segmentation refers to the decomposition of an image X into its component parts. We assume that we have m different class of objects $Q = \{1, 2, \ldots, m\}$.

In this section, we present an application of OSCLA for image segmentation. First, we model an image using Gibbs random field [24] and then present an algorithm based on OSCLA for image segmentation. Let $L_N = \{(i, j)|\ 1 \leq i, j \leq N\}$ denote the $N \times N$ integer lattice. We assume that the observation image is described by a random field $Y = \{y_{ij}|\ (i, j) \in L_N\}$, where $y_{ij} = F(x_{ij}) + n_{ij}$ represents the gray level of the observed image at pixel (i, j) and assumes a value from its domain of k discrete gray levels, and n_{ij} is a zero-mean white Gaussian noise with variance σ^2 and uncorrelated with the original image $X = \{x_{ij}|\ (i, j) \in L_N\}$. The random field Y is defined in terms of an underlying random field X, where X is the original image. The random field X is a discrete valued random field, where x_{ij} takes values in $Q = \{1, 2, \ldots, m\}$ for each pixel $(i, j) \in L_N$ and $x_{ij} = l$ denotes the fact that the pixel (i, j) in realization X, belongs to component of type l. The realization X, of course, is not observed and can not be obtained deterministically from Y. So the problem is to obtain an estimate $x^* = X^*(y)$ of the scene X based on realization $\{Y = y\}$. The MAP estimation involves determining x^*, an estimate for the scene, that minimizes the a posteriori distribution

$$x^* = \arg\max Prob[X = x|Y = y]. \tag{30}$$

Using the Bayes's rule we have

$$Prob[X = x|Y = y] = \frac{Prob[Y = y|X = x]Prob[X = x]}{Prob[Y = y]}. \tag{31}$$

Since $Prob[Y = y]$ does not affect the maximization process, it is equivalent to maximize the joint distribution $Prob[Y = y|X = x]Prob[X = x]$, which is equivalent to maximization of the following energy function

$$U(x) = \min_{x \in X} \left\{ \sum_{q \in Q} \frac{(\mu_q - x)^2}{2\sigma^2} + \sum_{c \in C} V_c(x) \right\}, \tag{32}$$

where μ_q corresponds to the mean of pixels associated to the class q, σ^2 is variance, and C is the set of all cliques. The function $V_c(x)$ is called the clique potential associated with clique c and its value is determined by x_{ij} with $(i, j) \in C$. In general, MAP estimation tries to maximize the a posteriori probability.

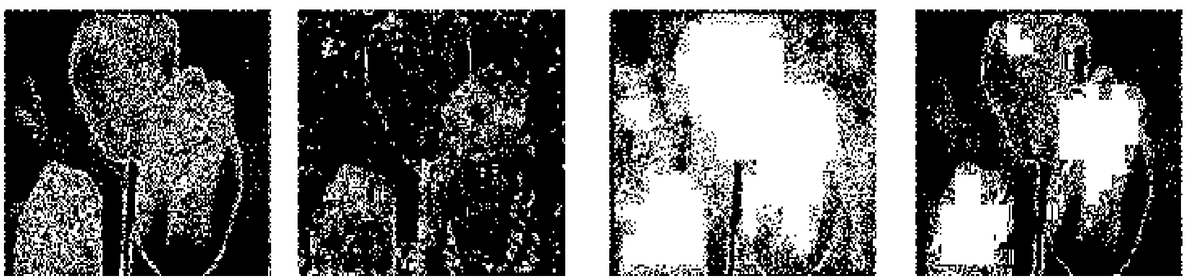
The proposed algorithm In the rest of this section, we present an algorithm based on OSCLA for image segmentation. In this algorithm, each cell of OSCLA has two environments : one local environment and one global environment. In the proposed algorithm, a stochastic learning automaton with m action is associated with each pixel with the possible $m - 1$ class of labels and the background at the pixel, which constituting the action-set of the automaton. Each of the m actions denotes the objects or background.

In each iteration, each LA chooses its action which corresponds to one class of labels. Then the global environment calculates the mean and the variance of the selected actions and the based on the similarity of the action of each cell and the actions selected its neighboring automata and its distance from the global mean and the global variance calculated by the global environment, the reward signal for each learning automata is produced. When each learning automaton chooses an action that results in a highest similarity, it receives reward and otherwise it receives penalty. This process repeated until in some consecutive iterations, the state of all automata remain fixed. The main steps of the proposed algorithm is given below.

Algorithm 1 Open synchronous cellular learning automata based algorithm for image segmentation.

repeat
  All LAs in OSCLA choose their actions.
  The global environment calculates the mean and variance of the chosen actions.
  The reinforcement signal is produced for each LA.
  All LAs update their action probability vector.
until (The configuration of OSCLA remains fixed for some iterations)

In order to evaluate the proposed algorithm, computer experiments are conducted on five 128×128 pixels 256 gray level images and compare the results obtained from the proposed algorithm with the results obtained from simulated annealing (SA) [25] and threshold based image segmentation algorithms [26]. The figures 6 through 10 show the results obtained from the proposed algorithm and the results obtained by other methods on images of figures 6 through 10, respectively.



a) The original image b) The results of simulated annealing based algorithm c) The results of Threshold based algorithm d) The results of OSCLA based algorithm

Fig. 6. The original image and the results after segmentation for image 1.

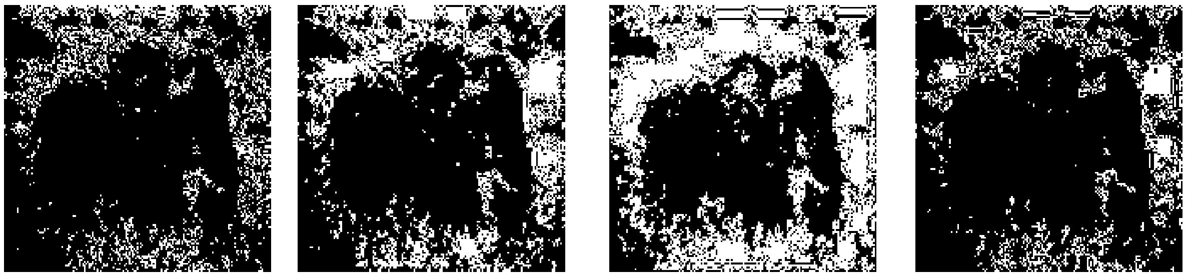
As the resulting images show, the OSCLA based image segmentation algorithms saves edges and their continuity and result in a better image segmentation.

6 Conclusions

In this paper, the open synchronous cellular learning automata is introduced and its steady state behavior is studied. It is shown that for commutative rules, the open cellular learning automata converges to a stable configuration for which the average rewards for the OSCLA is maximized. The results of computer experiments also confirms the theory.

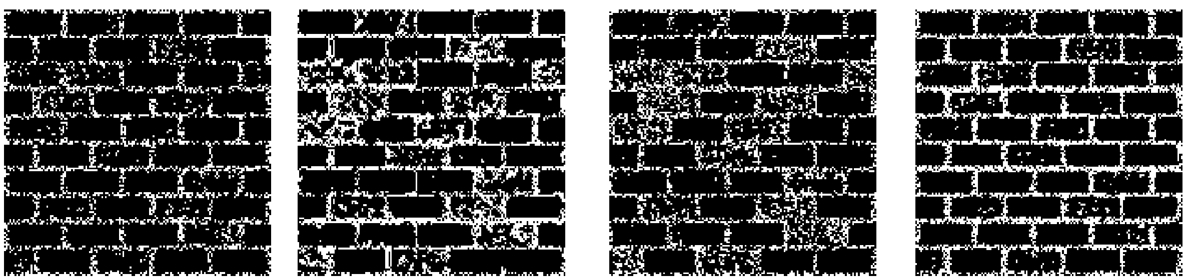
References

1. M. C. Borelijst and P. Hogeweg, "Spiral Wave Structure in Pre-Biotic Evolution: Hypercycles Stable Against Parasites," *Physica D*, vol. 48, pp. 17–28, 1991.



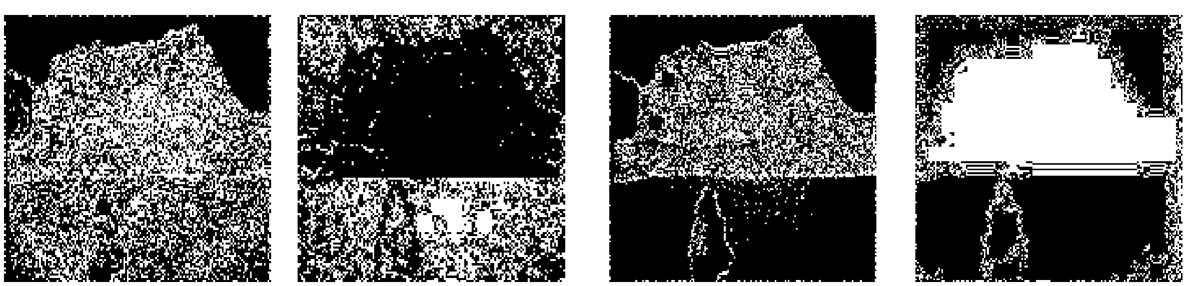
a) The original image
b) The results of simulated annealing based algorithm
c) The results of Threshold based algorithm
d) The results of OSCLA based algorithm

Fig. 7. The original image and the results after segmentation for image 2.



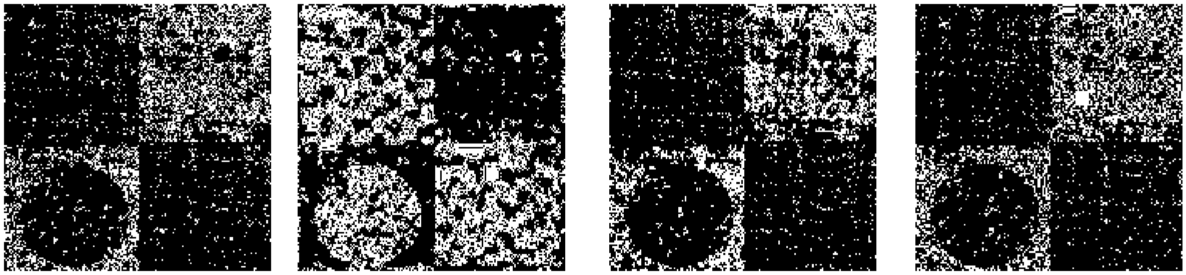
a) The original image
b) The results of simulated annealing based algorithm
c) The results of Threshold based algorithm
d) The results of OSCLA based algorithm

Fig. 8. The original image and the results after segmentation for image 3.



a) The original image
b) The results of simulated annealing based algorithm
c) The results of Threshold based algorithm
d) The results of OSCLA based algorithm

Fig. 9. The original image and the results after segmentation for image 4.



a) The original image b) The results of simulated annealing based algorithm c) The results of Threshold based algorithm d) The results of OSCLA based algorithm

Fig. 10. The original image and the results after segmentation for image 5.

2. Y. Gunji, "Pigment Color Patterns of Mulucs as an Autonomous Process Generated by Asynchronous Automata," *BioSystems*, vol. 23, pp. 317–334, 1990.
3. O. Inghe, "Genet and Ramet Survivorship Under Different Mortality Regimes- A Cellular Automata Model," *Journal of Theoritical Biology*, vol. 138, pp. 257–270, 1989.
4. K. S. Narendra and K. S. Thathachar, *Learning Automata: An Introduction*. New York: Printice-Hall, 1989.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. H. Beigy and M. R. Meybodi, "Adaptive Uniform Fractional Channel Algorithms," *Iranian Journal of Electrical and Computer Engineering*, vol. 3, pp. 47–53, Winter-Spring 2004.
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. Meybodi, H. Beigy, and M. Taherkhani, "Application of Cellular Learning Automata to Image Processing," in *Proceedings of First Conference in Mathematics and Communication, Iranian Telecommunication Research Center, Tehran, Iran*, pp. 23.1–23.10, Oct. 2000.
14. M. R. Meybodi and M. R. Kharazmi, "Image Restoration Using Cellular Learning Automata," in *Proceedings of the First Iranian Conf. on Machine Vision, Image Processeing and Applications, Birjand, Iran*, pp. 244–254, May 2000.
15. 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.
16. 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.
17. M. R. Meybodi and M. R. Kharazmi, "Application of Cellular Learning Automata to Image Processing," *Journal of Amirkabir, to be published*, 2003.
18. M. R. Meybodi and M. R. Khojasteh, "Application of Cellular Learning Automata in Modelling of Commerce Networks," in *Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC-2001, Isfehan, Iran*, pp. 284–295, Feb. 2001.
19. 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.

20. M. R. Meybodi and F. Mehdipour, "VLSI Placement Using Cellular Learning Automata," in *Proceedings of 8th Annual International Computer Society of Iran Computer Conference CSICC-2001, Mashad, Iran*, pp. 195–203, Mar. 2003.
21. M. R. Meybodi and H. Beigy, "A Mathematical Framework for Cellular Learning Automata," *Journal of Advances in Complex Systems, to be appreared.*
22. H. J. Kushner, *Approximation and Weak Convergence Methods for Random Processes.* Cambridge: MIT Press, 1984.
23. K. S. Narendra and A. M. Annaswarmy, *Stable Adaptive Systems.* New York: Printice-Hall International Inc., 1989.
24. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, November 1984.
25. H. K. Tan, S. B. Gelfand, and E. J. Delp, "A Cost Minimization Approach to Edge Detection Using Simulated Annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 3–18, Jan. 1992.
26. J. R. Beveridge, J. S. Griffith, R. R. Kohler, A. R. Hanson, and E. M. Riseman, "Segmenting Images Using Localized Histograms and Region Merging," *International Journal of Computer Vision*, vol. 2, pp. 311–352, Jan. 1989.