

3 **OPEN SYNCHRONOUS CELLULAR
LEARNING AUTOMATA**

5 HAMID BEIGY

7 *Computer Engineering Department,
Sharif University of Technology, Tehran, Iran*

9 *Institute for Studies in Theoretical Physics and Mathematics (IPM),
School of Computer Science, Tehran, Iran
beigy@ce.sharif.edu*

11 M. R. MEYBODI

13 *Computer Engineering Department,
Amirkabir University of Technology, Tehran, Iran*

15 *Institute for Studies in Theoretical Physics and Mathematics (IPM),
School of Computer Science, Tehran, Iran
mmeybodi@aut.ac.ir*

17 Received

Revised

19 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
21 and also is superior to single learning automaton because it is a collection of learning
23 automata which can interact together. In some applications such as image processing, a
25 type of cellular learning automata in which the action of each cell in the next stage of
27 its evolution not only depends on the local environment (actions of its neighbors) but it
29 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 a 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.

31 *Keywords:* Cellular automata; learning automata; cellular learning automata; dynamical
systems.

33 **1. Introduction**

35 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
37 topics, such as periodic evolution [1], the development of pigment patterns in mol-
lusks [2], and growth of clonal plants [3], to mention a few. Cellular automata is a
39 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

2 H. Beigy and M. R. Meybodi

1 cell. The environment of a cell is usually taken to be a small number of neighboring
 3 cells. The dynamics of cellular automata is generated by repeatedly applying
 5 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, homoge-
 7 nously applied at each step. Cellular automata perform complex computation with
 a high degree of efficiency and robustness.

7 On the other hand, learning automata (LA) are simple agents for doing simple
 9 things. The learning automata have a finite set of actions and at each stage choose
 11 one of them. The choice of an action depends on the state of automaton which
 13 is usually represented by an action probability vector. For each action chosen by
 15 the automaton, the environment gives a reinforcement signal with fixed unknown
 17 probability distribution, which specified the goodness of the applied action. Then
 19 upon receiving the reinforcement signal, the learning automaton updates its action
 21 probability vector by employing a learning algorithm. The interaction of the learn-
 ing automaton and its environment is shown in Fig. 1. The learning algorithm is a
 recurrence relation and is used to modify the action probability vector \underline{p} . Various
 learning algorithms have been reported in the literature. Below, a learning algo-
 rithm, 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
 $p(k)$. In L_{R-I} algorithm, the action probability vector is updated according to the
following rule.

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

23 When $\beta(k) = 0$ i.e. the environment rewards the chosen action of learning automa-
 25 ton and the action probability vector remains unchanged when $\beta(k) = 1$, i.e. the
 27 environment penalizes the chosen action of learning automaton. b ($0 < b < 1$) rep-
 29 presents *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 rout-
 ing [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.

31 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

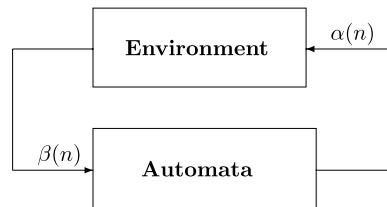


Fig. 1. The interaction of a learning automaton and its environment.

1 on the problem that needs to be solved, one of these structures for interaction may
 3 be chosen. In most applications, full interaction between all LAs is not necessary
 5 and is not natural. Local interaction of LAs, which can be defined in a form of graph
 7 such as tree, mesh, or array, is natural in many applications. On the other hand, CA
 9 are mathematical models for systems consisting of large numbers of simple identical
 components with local interactions. In Ref. 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.

11 CLA is a mathematical model for dynamical complex systems that consists of
 13 a large number of simple components [11]. The simple components, which have
 15 learning capability, act together to produce complicated behavioral patterns. A
 17 CLA is a CA in which a LA will be assigned to its every cell. The learning automaton
 19 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 non-stationary
 because of the fact that it changes as the action probability vectors of neighboring
 LAs vary.

21 The operation of cellular learning automata could be described as follows: At
 23 the first step, the internal state of every cell is specified. The state of every cell
 25 is determined on the basis of action probability vectors of the learning automata
 27 residing in that cell. The initial value of this state may be chosen on the basis of
 29 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.

31 **Definition 1 (Cellular Learning Automata).** A d -dimensional cellular learning
 automata is a structure $\mathcal{A} = (Z^d, \Phi, A, N, \mathcal{F})$, where

- (i) Z^d is a lattice of d -tuples of integer numbers.
- (ii) Φ is a finite set of states.
- (iii) A is the set of LAs each of which is assigned to each cell of the CA.
- (iv) $N = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{\bar{m}}\}$ 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 a set of cells
 $\{u + \bar{x}_i | i = 1, 2, \dots, \bar{m}\}$. We assume that, there exists a neighborhood function
 $\bar{N}(u)$ mapping a cell u to the set of its neighbors, that is

41
$$\bar{N}(u) = (u + \bar{x}_1, u + \bar{x}_2, \dots, u + \bar{x}_{\bar{m}}). \quad (2)$$

4 H. Beigy and M. R. Meybodi

1 For the sake of simplicity, we assume that the first element of neighborhood
 2 vector (i.e. \bar{x}_1) is equal to d -tuple $(0, 0, \dots, 0)$. The neighborhood function
 3 $\bar{N}(u)$ must satisfy in the two following conditions:

- $u \in \bar{N}(u)$ for all $u \in Z^d$.
- 5 — $u_1 \in \bar{N}(u_2) \iff u_2 \in \bar{N}(u_1)$ for all $u_1, u_2 \in Z^d$.

7 (v) $\mathcal{F} : \Phi^{\bar{m}} \rightarrow \underline{\beta}$ is the local rule of the cellular learning automata, where $\underline{\beta}$ is the
 7 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.

9 A number of applications for CLA have been developed recently such as rumor
 11 diffusion [12], image processing [13–18], modeling of commerce networks [19], fixed
 13 channel assignment in cellular networks [20], and VLSI Placement [21] to mention a
 15 few. The CLA can be classified into *synchronous* and *asynchronous*. In synchronous
 17 CLA, all cells are synchronized with a global clock and executed at the same time.
 In Ref. 22, 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.

19 In some applications such as image processing, a type of cellular learning
 21 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
 23 on the external environments. We call such a CLA as *open synchronous cellular*
 25 *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 application of OSCLA to image
 processing has been presented.

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

31 2. Open Synchronous Cellular Learning Automata

33 CLA studied so far are closed, because they do not take into account the interaction
 between the CLA and the external environments. In this section, a new class of
 35 CLA called *open synchronous cellular learning automata* (OSCLA) is introduced,
 in which the evolution of CLA is influenced by local and external environments. In
 37 CLA, the neighboring LAs of any cell constitute its local environment. Two types of
 environments can be considered in the OSCLA: global environment and exclusive
 39 environment. Each CLA has one global environment that influences all cells and
 an exclusive environment for each particular cell influencing the evolution of that

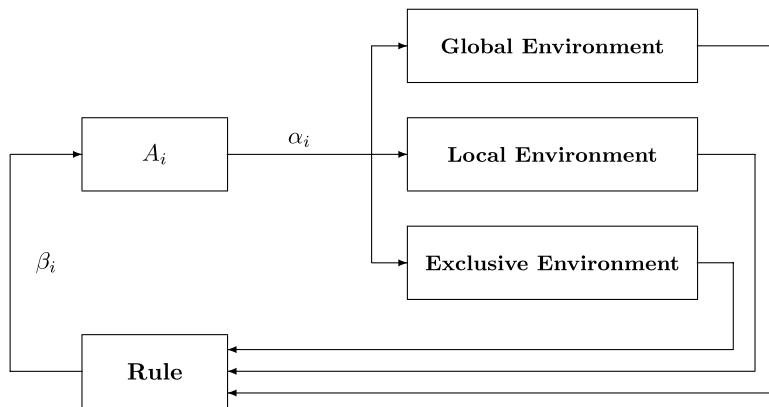


Fig. 2. The interconnection of a typical cell in OSCLA with its various environments.

1 cell. The interconnection of a typical cell in the OSCLA and its various types of
2 environments is shown in Fig. 2.

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

5 **Definition 2 (Open Synchronous Cellular Learning Automata).** A d -
6 dimensional open synchronous cellular learning automata is a structure $\mathcal{A} = (Z^d, \Phi,$
7 $A, E^G, E^E, N, \mathcal{F})$, where

- (i) Z^d is a lattice of d -tuples of integer numbers.
- 9 (ii) Φ is a finite set of states.
- (iii) A is the set of LAs each of which is assigned to each cell of the CA.
- 11 (iv) E^G is the global environment.
- 13 (v) $E^E = \{E_1^E, E_2^E, \dots, E_n^E\}$ is the set of exclusive environments, where E_i^E is
the exclusive environment for learning automaton residing in cell i .
- 15 (vi) $N = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{\bar{m}}\}$ is neighborhood vector.
- 17 (vii) $\mathcal{F} : \underline{\Phi}^{\bar{m}} \times \underline{\mathcal{Q}}(G) \times \underline{\mathcal{Q}}(E) \rightarrow \underline{\beta}$ is the local rule of the cellular automata, where
 $\underline{\mathcal{Q}}(G)$ and $\underline{\mathcal{Q}}(E)$ is the set of reinforcement signals of global and exclusive
environments, respectively.

19 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
21 to cell i (for $i = 1, \dots, n$) of the OSCLA. Let cardinality of $\underline{\alpha}_i$ be m_i and the
23 state of the OSCLA 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 automaton A_i .

25 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

6 H. Beigy and M. R. Meybodi

1 environment produces a signal, which are used by the local rule to generate a
 3 reinforcement signal to the learning automaton residing in every cell. The higher
 5 values of β_i means that the chosen action of automaton A_i is more rewarded,
 7 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 non-stationary
 while global and exclusive environments may be stationary or non-stationary. In
 this paper, we assume that the global and exclusive environments are stationary.

9 In the following sections, we give some definitions and notations, which will be
 used later in Sec. 3 to analysize the behavior of OSCLA.

11 2.1. Definitions and notations

13 In this section, 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.

15 **Definition 3.** A configuration of OSCLA is a mapping $\mathcal{K} : Z^d \rightarrow \underline{p}$ that asso-
 17 ciates 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} .

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

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

23 and

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

25 respectively.

27 Note that, the set of probabilistic configurations \mathcal{K} is the convex hull of the set
 of all deterministic configurations \mathcal{K}^* [22]. The application of the local rule to every
 cell allows transforming a configuration to a new one.

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

31 **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 $\underline{p}(k+1) = \mathcal{G}(\underline{p}(k))$.

1 **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

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

5 where z and w are the signals produced by the global and the exclusive environments
 7 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

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

9 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 .

11 **Definition 8.** A configuration $\underline{p} \in \mathcal{K}$ is compatible provided

$$13 \quad \sum_r d_{ir}(\underline{p}) p_{ir} \geq \sum_r d_{ir}(\underline{p}) q_{ir} \quad (5)$$

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

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

21 **Definition 9.** A configuration $\underline{p} \in \mathcal{K}$ is admissible provided

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

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

25 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.

27 **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.

29 **Proof.** The proof is trivial from Definitions 8 and 9. \square

31 **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

$$29 \quad \mathcal{D}(\underline{p}) = \sum_i D_i(\underline{p}). \quad (7)$$

31 **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 $\underline{q} \in \mathcal{K}$.*

33 **Proof.** The proof is trivial by looking Definitions 9 and 10. \square

8 H. Beigy and M. R. Meybodi

1 **Remark 1.** With the approach given in this paper, the probability of different
 3 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.

5 **Lemma 1.** OSCLA has at least one compatible configuration.

7 **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} . Then
 we introduce the following transformation

$$9 \quad \bar{p}_{ir} = \frac{p_{ir} + \phi_{ir}}{1 + \sum_{j=1}^m \phi_{ij}}, \quad (8)$$

for $i = 1, \dots, n$ and $r = 1, \dots, m_i$ and denote it by mapping $T : \mathcal{K} \rightarrow \mathcal{K}$ given by

$$11 \quad \bar{\underline{p}} = T(\underline{p}). \quad (9)$$

13 It is evident that T is a continuous mapping. Since \mathcal{K} is closed, bounded and convex,
 15 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
 17 compatible configuration of OSCLA and conversely every compatible configuration
 19 is a fixed point of T , thereby concluding the proof of the theorem. We first verify
 21 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 $i = 1, \dots, n$. Configuration
 \underline{q} also includes $\underline{q} = (\underline{p}'_1, \dots, \underline{e}'_{r_i}, \dots, \underline{p}'_n)'$ for fixed i ($i = 1, \dots, n$). Thus we obtain
 $\psi_{ir_i}(\underline{p}) \leq 0$. Hence, $\phi_{ir_i} = 0$ for all $i = 1, \dots, n$ and $r_i = 1, \dots, m_i$ and we have
 $\underline{p} = T(\underline{p})$, which concluding that \underline{p} is a fixed point of T .

23 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 $\tilde{\underline{p}}_i$ such that $\tilde{\underline{p}}_i = (\underline{p}'_1, \dots, \tilde{\underline{p}}'_i, \dots, \underline{p}'_n)'$ and

$$25 \quad \sum_r d_{ir}(\underline{p})p_{ir} < \sum_r d_{ir}(\underline{p})\tilde{p}_{ir}. \quad (10)$$

27 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 $D_i(\tilde{\underline{p}}_i)$ can be bounded from above by $d_{iy_i}(\underline{p})$, thus
 29 implies $\psi_{iy_i}(\underline{p}) > 0$, which implies $\phi_{iy_i}(\underline{p}) > 0$. But since $\phi_{ir_i}(\underline{p})$ is non-negative 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
 31 reward for automaton A_i attains its minimum value. Then by using inequality (10),
 it can be shown that $D_i(\underline{p})$ is bounded below by $d_{ir_i}(\underline{p})$. This implies $\psi_{ir_i}(\underline{p}) < 0$,
 33 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 . \square

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

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

35 for all i and r .

1 **Proof.** If $\underline{p} \in \mathcal{K}$ is a compatible configuration, then from (5), for every $\underline{q} \in \mathcal{K}$ and
 2 $1 \leq i \leq n$, we have $D_i(\underline{p}) \geq D_i(\underline{q})$. Since, \underline{q} includes $\underline{q} = (\underline{p}'_1, \dots, \underline{e}'_{r_i}, \dots, \underline{p}'_n)'$ for
 3 fixed i ($i = 1, \dots, n$), then we obtain $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$.

4 Conversely, suppose that $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$ ($i = 1, \dots, n$ and $r_i = 1, \dots, m_i$) but
 5 \underline{p} is not compatible. Then for some learning automaton A_i with action probability
 6 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 $d_{iy_i}(\underline{p}) >$
 7 $D_i(\underline{q})$. Action y_i denotes the action for which $d_{iy_i}(\underline{p})$ attains its maximum value.
 8 Since \underline{q}_i is a probability vector, then $D_i(\underline{q})$ is bounded from the above with $d_{iy_i}(\underline{p})$
 9 and arrives at strict inequality $D_i(\underline{p}) < D_i(\underline{q}) < d_{iy_i}(\underline{p})$. But this contradicts the
 10 hypothesis that $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$ (for all $r_i = 1, \dots, m_i$), which concludes that \underline{p} is a
 11 compatible configuration. \square

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 Eq. (4), we obtain

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

13 The above contradiction completes the proof of the lemma. \square

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

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

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

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

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

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

10 *H. Beigy and M. R. Meybodi*

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 $\underline{q} = (\underline{p}'_1, \dots, \underline{q}'_l, \dots, \underline{p}'_n)'$, we obtain

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

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

3 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 $\underline{p} = (\underline{e}'_{t_1}, \underline{e}'_{t_2}, \dots, \underline{e}'_{t_n})'$ is compatible if and only if*

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

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

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

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

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

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

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

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

13 which concludes this assertion.

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

$$\begin{aligned} &\sum_{t_2} \dots \sum_{t_{\bar{m}}} \sum_z \sum_w \mathcal{F}^i(t^i, t_2, \dots, t_{\bar{m}}, z, w) d_z^G d_w^E \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l} \\ &\geq \sum_{t_2} \dots \sum_{t_{\bar{m}}} \sum_z \sum_w \mathcal{F}^i(r^i, t_2, \dots, t_{\bar{m}}, z, w) d_z^G d_w^E \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{lt_l}. \end{aligned}$$

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 doesnot change the value of the above inequality thus we obtain

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

1 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. \square

3 **Corollary 3.** A corner $\underline{p} = (\underline{e}'_{t_1}, \underline{e}'_{t_2}, \dots, \underline{e}'_{t_n})'$ is fully compatible if and only if
5 $\mathcal{F}^i(t_1, t_2, \dots, t_{\bar{m}}, z, w) > \mathcal{F}^i(r, t_2, \dots, t_{\bar{m}}, z, w)$ for all $z \in \underline{\mathcal{Q}}(G), w \in \underline{\mathcal{Q}}(E^i)$, $i = 1, \dots, n$ and $r \neq t_i$.

Proof. The proof is trivial from the proof of Theorem 2. \square

7 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 operate under stationary global and exclusive environments. Using the L_{R-I} learning algorithm, process $\{\underline{p}(k)\}_{k \geq 0}$ is Markovian and can be described by the following difference equation:

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

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 $M \times M$ 15 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 17 learning parameter for learning automaton A_i . Let $\underline{B} = (b_1, \dots, b_n)'$ denote the learning parameter of learning automata in OSCLA. Now, define

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

19 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

$$21 \Delta\underline{p}(k) = \underline{\Lambda}\underline{f}(\underline{p}(k)). \quad (15)$$

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

$$\begin{aligned} \Delta p_{iy}(k) &= b_i p_{iy}(k) [1 - p_{iy}(k)] E[\beta_{iy}(k)] - b_i \sum_{r \neq y} p_{ir}(k) p_{iy}(k) E[\beta_{ir}(k)] \\ &= b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) E[\beta_{iy}(k)] - b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) E[\beta_{ir}(k)] \\ &= b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) \{E[\beta_{iy}(k)] - E[\beta_{ir}(k)]\} \\ &= b_i p_{iy}(k) \sum_{r \neq y} p_{ir}(k) [d_{iy}(\underline{p}) - d_{ir}(\underline{p})] \\ &= b_i f_{iy}(\underline{p}), \end{aligned} \quad (16)$$

12 *H. Beigy and M. R. Meybodi*

where

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

1 For different values of \underline{B} , Eq. (13) generates different process and we shall use
 2 $\underline{p}^B(k)$ to denote this process whenever the value of \underline{B} is to be specified explicitly.
 3 Define a sequence of continuous time interpolation of $\underline{p}^B(k)$, denoted by $\tilde{\underline{p}}^B(t)$ and
 4 called the *interpolated process*, whose components are defined by

$$5 \quad \tilde{\underline{p}}_i^B(t) = \underline{p}_i(k), \quad t \in [kb_i, (k+1)b_i]. \quad (18)$$

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

$$6 \quad \dot{\underline{p}} = \underline{f}(\underline{p}), \quad (19)$$

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

$$13 \quad \frac{dp_{iy}}{dt} = p_{iy} [d_{iy}(\underline{p}) - D_i(\underline{p})]. \quad (20)$$

15 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 $\tilde{\underline{p}}^B$ as $\max\{b_1, b_2, \dots, b_n\}$ is sufficiently small. We show that the sequence of interpolated process $\{\tilde{\underline{p}}^B(t)\}$ converges weakly to the solution of ODE (19) with initial configuration $\underline{p}(0)$. This implies that the asymptotic behavior of $\underline{p}(k)$ can be obtained from the solution of ODE (19).

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

$$25 \quad \frac{d\underline{X}}{dt} = \underline{f}(\underline{X}) \quad (21)$$

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

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

- 27 (i) $\{\underline{p}(k), (\underline{\alpha}(k-1), \underline{\beta}(k-1))\}_{k \geq 0}$ is a Markov process.
- (ii) $(\underline{\alpha}(k), \underline{\beta}(k))$ takes values in a compact metric space.

- 1 (iii) \underline{g} is bounded, continuous and independent of \underline{B} .
 (iv) ODE (21) has a unique solution for each initial condition $\underline{X}(0)$.
 3 (v) 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 [23], sequence $\{\tilde{\underline{p}}^B(\cdot)\}$ converges weakly, as $\max\{b_1, \dots, b_n\} \rightarrow 0$ to the solution of

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

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 $\underline{p}(k) = \hat{\underline{p}}$, $(\underline{\alpha}(k), \underline{\beta}(k))$ is an independent identically distributed sequence whose distribution depends only on $\hat{\underline{p}}$ and the rule of OSCLA, then we have

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

5 and hence the theorem. \square

7 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 $\max\{b_1, \dots, b_n\} \rightarrow 0$. As the length of the time interval increases and
 $\max\{b_1, \dots, b_n\} \rightarrow 0$, 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,
 $\underline{p}^B(\cdot)$ 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 $\max\{b_1, \dots, b_n\} \rightarrow 0$.
 15 Although the speed of convergence depends on the specific value of \underline{B} . The above
 point is summarized in the following lemma.

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

Proof. Let $\underline{X}(\cdot)$ 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 $h_T(\cdot)$ is continuous on \mathcal{K} . Then Theorem 3 says that $E[h_T(\tilde{\underline{p}})] \rightarrow E[h_T(\underline{X})] = 0$ as $\max\{b_1, \dots, b_n\} \rightarrow 0$. 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 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 $k \rightarrow \infty$. With the particular initial condition used, let \underline{p}^o be the equilibrium

14 *H. Beigy and M. R. Meybodi*

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

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

1 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 . \square

3 In the following subsections, we first find the equilibrium points of ODE (19), then
5 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

7 The equilibrium points of Eq. (15) are those points that satisfy the set of equations
9 $\Delta p_{ij}(k) = 0$ 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.

11 **Lemma 5.** 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

$$13 \quad d_{iy}(\underline{p}) = d_{ir}(\underline{p}), \quad (22)$$

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

15 **Proof.** From Eq. (16), it is obvious that f_{iy} (for $i = 1, 2, \dots, n$ and $y = 1, 2, \dots, m_i$)
is zero if \underline{p}_i is a unit vector and hence all corners of \mathcal{K} are equilibrium points of $\underline{f}(\cdot)$.
17 To find other zeros of $\underline{f}(\cdot)$, 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,

$$19 \quad \sum_{r \neq y} p_{ir}(k) [d_{iy}(\underline{p}) - d_{ir}(\underline{p})] = 0. \quad (23)$$

The above equation can be rewritten as

$$\begin{aligned} \sum_{r \neq y} p_{ir}(k) [d_{iy}(\underline{p}) - d_{ir}(\underline{p})] &= \sum_{r \neq y} p_{ir}(k) d_{iy}(\underline{p}) - \sum_{r \neq y} p_{ir}(k) d_{ir}(\underline{p}) \\ &= d_{iy}(\underline{p}) [1 - p_{iy}(k)] - \sum_{r \neq y} p_{ir}(k) d_{ir}(\underline{p}) \\ &= d_{iy}(\underline{p}) - \sum_r p_{ir}(k) d_{ir}(\underline{p}) \end{aligned}$$

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

Thus, we obtain

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

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

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

When $d_0 \neq 0$, Eq. (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

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

- 1 for $\forall i = 1, 2, \dots, n$ and $\forall y \neq q$. When some p_{iy} are zero, for \underline{f} to be zero, Eq. (23)
 3 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. \square

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

- 5 **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 . \square

7 3.2. The stability property

- In this section, we characterize the stability of equilibrium configurations of OSCLA,
 9 that is the equilibrium points of the ODE (19). From Lemmas 5 and 6, all the equi-
 librium points of (19) are known. In order to study the stability of the equilibrium
 11 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
 13 are concerned with the stability properties of the equilibrium points of ODE (19).

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

16 H. Beigy and M. R. Meybodi

Proof. Let configuration $\underline{p}^o = (\underline{e}'_{t_1}, \dots, \underline{e}'_{t_n})'$ 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

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

where the t_g and t_w are the response of the global and exclusive environments. We consider the following positive definite Lyapunov function $V(\tilde{\underline{p}}) = \sum_i \sum_{y \neq t_i} \tilde{p}_{iy}$, where $V(\tilde{\underline{p}}) \geq 0$ and is zero when $\tilde{p}_{iy} = 0$ for all i, y , and its derivative is equal to $\dot{V}(\tilde{\underline{p}}) = \sum_i \sum_{y \neq t_i} f_{iy}$. Since corner \underline{p}^o is a fully compatible configuration, then from Theorem 2 we have $\mathcal{F}^i(y, t_2, \dots, t_{\bar{m}}, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \dots, t_{\bar{m}}, t_g, t_e) < 0$ for $i = 1, 2, \dots, n$. Thus, Eq. (26) implies that there is a neighborhood around \underline{p}^o such that the linear terms dominate the high order terms. Hence, $\dot{V}(\tilde{\underline{p}}) < 0$ 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 (19) can be written as $\dot{\tilde{\underline{p}}} = A\tilde{\underline{p}}$, where $A = \text{diag}(\tilde{f}_{iy})$ and $\tilde{f}_{iy} = \mathcal{F}^i(y, t_2, \dots, t_{\bar{m}}, t_g, t_e) - \mathcal{F}^i(t_i, t_2, \dots, t_{\bar{m}}, t_g, t_e)$ for $i = 1, 2, \dots, n$. Since \underline{p}^o is uniformly asymptotical stable, A should have eigenvalues with negative real parts and hence $\tilde{f}_{iy} < 0$. Using Theorem 2, this implies that \underline{p}^o is a fully compatible configuration. This completes the proof of this lemma. \square

15 **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 $\frac{d\underline{p}_{jy}}{dt} > 0$ 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. \square

23 **Remark 2.** In Lemmas 7 and 8, the solution of ODE (19) well-characterized and
25 it is shown that full compatibility implies uniformly asymptotic stability of the cor-
26 ners. In order to obtain necessary and sufficient conditions for uniformly asymptotic
27 stability, it is essential to consider in detail the nonlinear terms in the differential
equation, which appears to be a difficult problem.

1 **3.3. Convergence results**

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

- 5 (i) $\underline{p}(0)$ 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.
- 7 (ii) $\underline{p}(0)$ 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 (iv) below.
- 9 (iii) $\underline{p}(0) \in \mathcal{K}^*$. 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 $\underline{p}(0)$.
- 11 (iv) $\underline{p}(0) \in \mathcal{K} - \mathcal{K}^*$. The convergence results of the OSCLA for these initial configurations is stated in Theorem 4.

17 **Theorem 4.** Suppose there is a bounded differential function $\mathcal{D}: \mathcal{R}^{m_1+\dots+m_m+m_g+m_e} \rightarrow \mathcal{R}$ such that for some constant $c > 0$, $\frac{\partial \mathcal{D}}{\partial p_{ir}}(\underline{p}) = cd_{ir}(\underline{p})$ 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 ($\max\{b_1, b_2, \dots, b_n\} \rightarrow 0$), always converges to a configuration, that is stable and compatible.

23 **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 be modified. The modified neighborhood function 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

(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)

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

1 synchronous CLA to prove the convergence of OSCLA, we use the zero value for
 2 the learning parameter of learning automata representing the global and exclusive
 3 environments.

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

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

1 OSCLA updates the action probabilities in a such a way that $\underline{p}(k) \in \mathcal{K}$ for
 3 all $\underline{p}(0) \in \mathcal{K}$ and $k > 0$. Since \mathcal{K} is a compact subset of $\mathcal{R}^{m_1+\dots+m_m+m_g+m_e}$,
 5 asymptotically all solutions of ODE (19) will be in \mathcal{K} . Inequality (27) shows that
 7 OSCLA updates the configuration probabilities in the gradient ascent manner and
 9 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. \square

11 **Remark 3.** If the OSCLA satisfies the sufficiency conditions needed for Theorem 4, then OSCLA will converge to a compatible configuration. When the OSCLA
 13 doesnot satisfy this sufficiency condition, convergence to compatible configurations
 cannot be guaranteed and the OSCLA may exhibit a limit cycle behavior [24].

15 **4. Open Synchronous Cellular Learning Automata Using
 Commutative Rules**

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

Definition 11 (Commutative Rule). A rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \dots, \alpha_{i+\bar{x}_{\bar{m}}})$ is
 called commutative if and only if

$$\begin{aligned}\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \dots, \alpha_{i+\bar{x}_{\bar{m}}}) &= \mathcal{F}^i(\alpha_{i+\bar{x}_{\bar{m}}}, \alpha_{i+\bar{x}_1}, \dots, \alpha_{i+\bar{x}_{\bar{m}}}) \\ &= \dots = \mathcal{F}^i(\alpha_{i+\bar{x}_2}, \alpha_{i+\bar{x}_3}, \dots, \alpha_{i+\bar{x}_1}).\end{aligned}\quad (28)$$

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

25 **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 [22], 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,

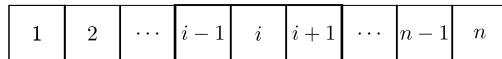


Fig. 4. The linear OSCLA.

1 then $\mathcal{D}(\underline{p})$ does not increase as one moves away from \underline{p} . Thus we have

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

Thus using the 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 the (l, r) th component of $\nabla\mathcal{D}(\underline{q})$ 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 .

$$\begin{aligned} q_{lr} &= \frac{\partial}{\partial p_{lr}} \sum_i \sum_y \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(y, x, z, u, v) d_u^G d_v^E p_{jx} p_{iy} p_{kz} \\ &= 5d_{lr}(\underline{p}), \end{aligned} \quad (30)$$

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^a

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

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

5 **Remark 4.** In general, when rules of OSCLA are not commutative, the local maxima for $\mathcal{D}(\underline{p})$ still exist, but they may not be compatible.

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

9 **Theorem 6.** An OSCLA, which uses uniform and commutative rules, starting
11 from $\underline{p}(0) \in \mathcal{K} - \mathcal{K}^*$ and with sufficiently small value of learning parameter,
 $(\max\{b_1, \dots, b_n\} \rightarrow 0)$, always converges to a deterministic configuration, that is
stable and also compatible.

13 **Proof.** Let function $\mathcal{D} : \mathcal{R}^{m_1 + \dots + m_m + m_g + m_e} \rightarrow \mathcal{R}$ be the total average reward for
15 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. \square

^aThe details for derivation of this equation is given in the Appendix.

1 **Remark 5.** From the proof of Theorem 6, we can conclude that the OSCLA converges to one of its compatible configurations, if any. If this compatible configuration
 3 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
 5 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.

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

9 **5. Computer Experiments**

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

13 **5.1. Numerical examples**

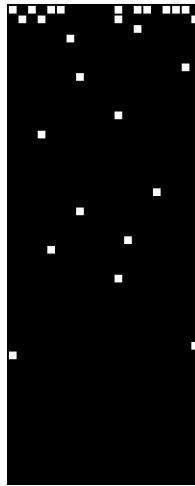
15 This section discusses patterns formed by the evolution of cellular learning automata from random initial configuration, chosen by the learning automata in cells. Different cellular learning automata rules are found to yield different configurations.
 17 For the sake of simplicity in our presentation, 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 the 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.
 19 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
 21 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
 23 signal to be taken to the central cell on the next timestep.

25 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
 27 of CLA with 20 cells and a two-actions L_{R-I} learning automaton in each cell. The
 29 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
 31 probability of 0.3 and 0.7, respectively.

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

22 *H. Beigy and M. R. Meybodi*



rule $(32768)_2$

Fig. 5. Time-space diagram for OSCLA.

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

3 **5.2. Image segmentation**

5 The ultimate aim in a large number of image processing applications, like medi-
 7 cal image analysis, optical character recognition, industrial automation, and photo
 9 editing, is to extract important features from image data from which a description,
 11 interpretation, or understanding of the scene can be provided by machine. In such
 13 applications, image analysis basically involves the study of feature extraction, seg-
 15 mentation, and classification techniques. In machine analysis of image systems, the
 17 image is first preprocessed and then certain features are extracted to decompose the
 19 image into its components and finally assign the components or segmented parts
 21 into one of several objects, each identified by a label. Image segmentation refers
 23 to the decomposition of an image X into its component parts. We assume that we
 have m different classes of objects $Q = \{1, 2, \dots, m\}$.

15 In this section, we present an application of OSCLA for image segmentation.
 17 First, we model an image using Gibbs random field and then present an algorithm
 19 based on OSCLA for image segmentation. Let $L_N = \{(i, j) | 1 \leq i, j \leq N\}$ denote
 21 the $N \times N$ integer lattice. We assume that the observation image is described by a
 23 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

1 in $Q = \{1, 2, \dots, m\}$ for each pixel $(i, j) \in L_N$ and $x_{ij} = l$ denotes the fact that the
 3 pixel (i, j) in realization X , belongs to component of type l . The realization X , of
 5 course, is not observed and cannot be obtained deterministically from Y . Hence 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,

$$7 \quad x^* = \arg \max Prob[X = x|Y = y]. \quad (31)$$

Using Bayes's rule, we have

$$9 \quad Prob[X = x|Y = y] = \frac{Prob[Y = y|X = x]Prob[X = x]}{Prob[Y = y]}. \quad (32)$$

11 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 the maximization of the following energy function:

$$13 \quad U(x) = \min_{x \in X} \left\{ \sum_{q \in Q} \frac{(\mu_q - x)^2}{2\sigma^2} + \sum_{c \in C} V_c(x) \right\}, \quad (33)$$

15 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
 17 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.

19 **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
 21 has two environments: one local environment and one global environment. In the
 proposed algorithm, a stochastic learning automaton with m action is associated
 23 with each pixel with the possible $m - 1$ class of labels and the background at the
 pixel, which constitutes the action-set of the automaton. Each of the m actions
 denotes the objects or background.

25 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
 27 selected actions. 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
 29 global variance calculated by the global environment, the reward signal for each
 learning automata is produced. When each learning automaton chooses an action
 31 that results in a highest similarity, it receives a reward and otherwise it receives a
 penalty. This process is repeated until in some consecutive iterations, the state of all
 33 automata remain fixed. The main steps of the proposed algorithm is given below.

35 In order to evaluate the proposed algorithm, computer experiments are con-
 ducted on five 128×128 pixels 256 gray level images and we compare the results
 obtained from the proposed algorithm with the results obtained from simulated
 annealing (SA) and threshold-based image segmentation algorithms. Figures 6–10

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

repeat

 Each LA in OSCLA chooses its actions.

 The global environment calculates the mean and variance of the chosen actions.

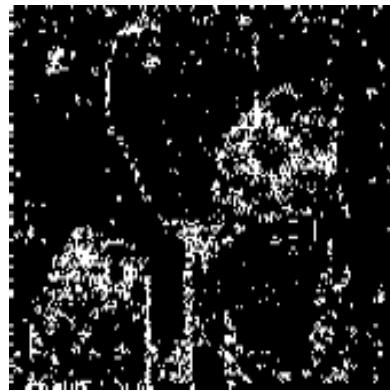
 The reinforcement signal is produced for each LA.

 Each LA updates its action probability vector.

until (The configuration of OSCLA remains fixed for some iterations.)



(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.

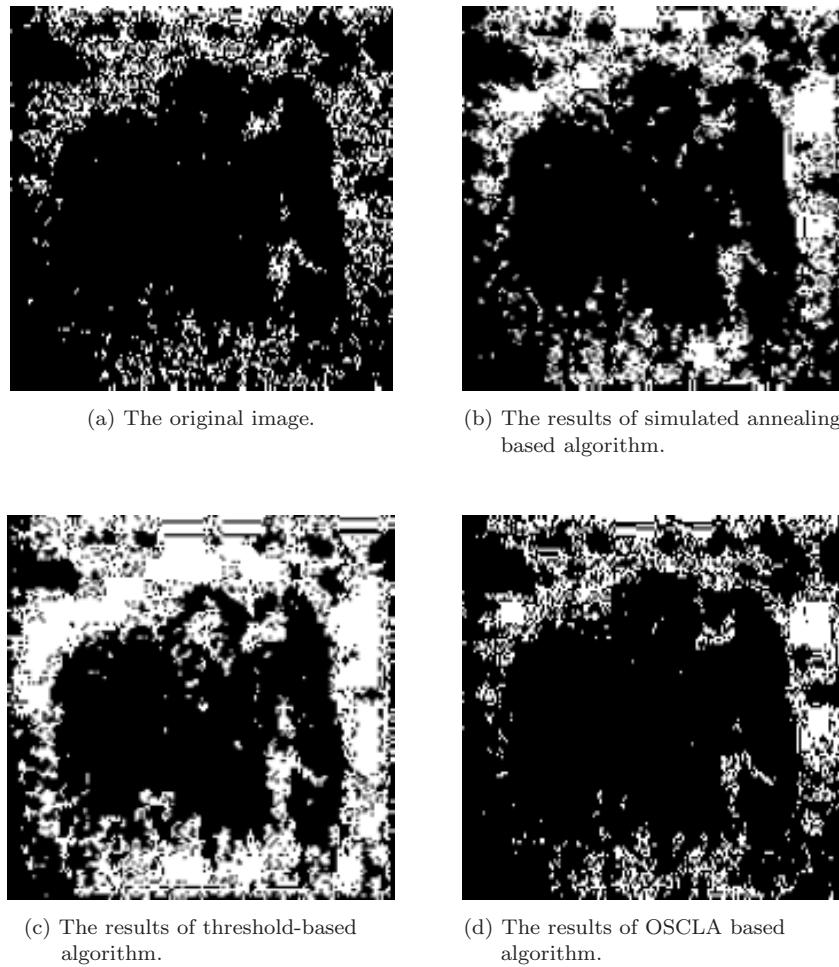


Fig. 7. The original image and the results after segmentation for image 2.

1 show the results obtained from the proposed algorithm and the results obtained by
2 other methods on images of Figs. 6–10, respectively.

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

5 6. Conclusion

7 In this paper, the open synchronous cellular learning automata is introduced and
8 its steady state behavior is studied. It is shown that for commutative rules, the
9 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.

26 H. Beigy and M. R. Meybodi

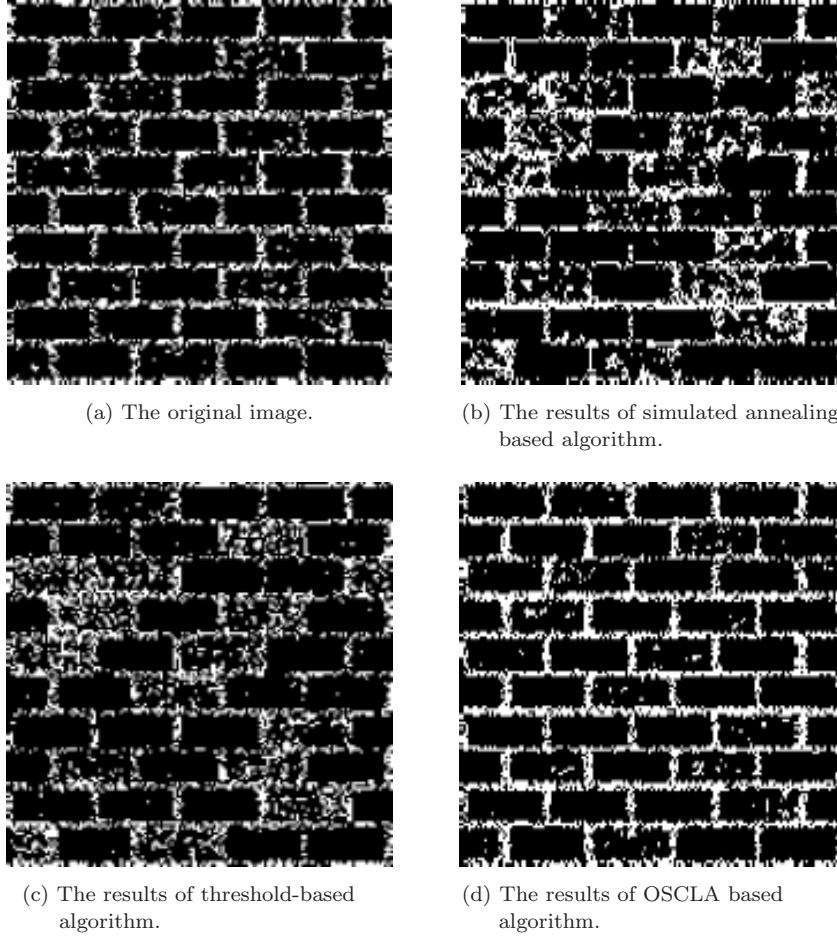


Fig. 8. The original image and the results after segmentation for image 3.

1 Appendix

In this appendix, we give the derivation of Eq. (30).

$$\begin{aligned}
 q_{lr} &= \frac{\partial}{\partial p_{lr}} \sum_i \sum_y \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(y, x, z, u, v) d_u^G d_v^E p_{jx} p_{iy} p_{kz} \\
 &= \sum_i \sum_x \sum_y \sum_z \sum_u \sum_v [\mathcal{F}^i(y, x, z, u, v) \delta_{lj} \delta_{rx} d_u^G d_v^E p_{iy} p_{kz} \\
 &\quad + \mathcal{F}^i(y, x, z, u, v) \delta_{li} \delta_{ry} d_u^G d_v^E p_{jx} p_{kz} + \mathcal{F}^i(y, x, z, u, v) \delta_{lk} \delta_{rz} d_u^G d_v^E p_{jx} p_{iy} \\
 &\quad + \mathcal{F}^i(y, x, z, u, v) \delta_{la} \delta_{ru} d_u^G d_v^E p_{iy} p_{kz} + \mathcal{F}^i(y, x, z, u, v) \delta_{lb} \delta_{rv} d_u^G d_v^E p_{jx} p_{kz}]
 \end{aligned} \tag{34}$$

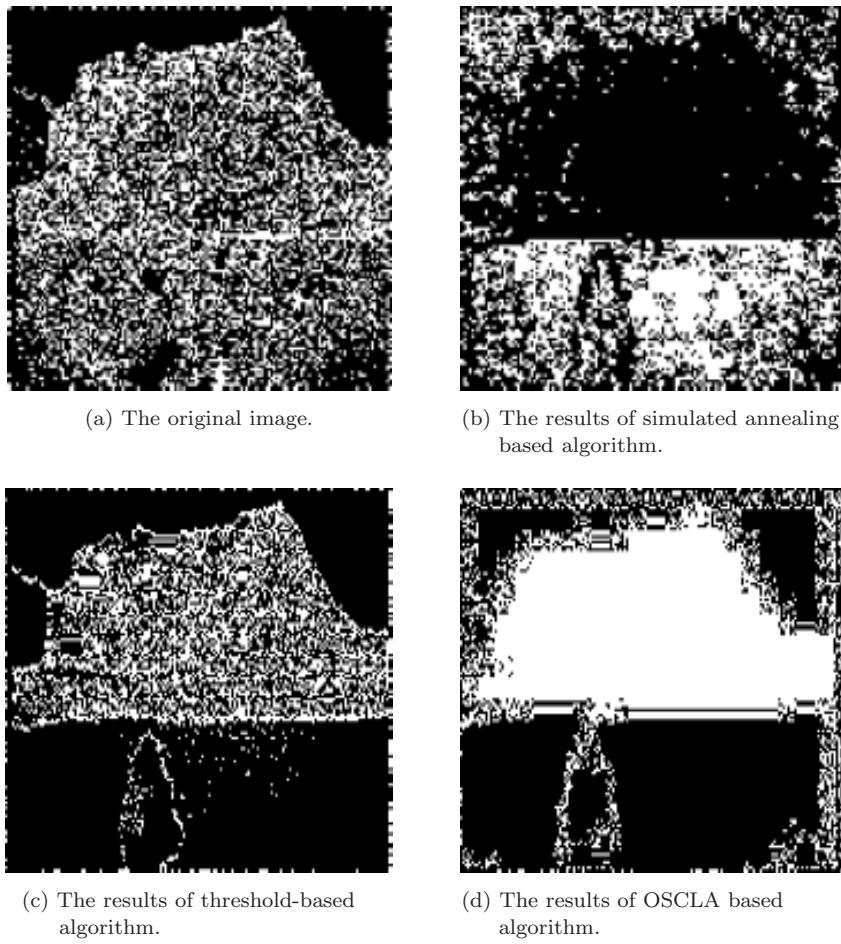


Fig. 9. The original image and the results after segmentation for image 4.

$$\begin{aligned}
 &= \sum_y \sum_z \sum_u \sum_v \mathcal{F}^i(y, r, z, u, v) d_u^G d_v^E p_{iy} p_{kz} \\
 &\quad + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r, x, z, u, v) d_u^G d_v^E p_{jx} p_{kz} \\
 &\quad + \sum_x \sum_y \sum_u \sum_v \mathcal{F}^i(y, x, r, u, v) d_u^G d_v^E p_{jx} p_{iy} \\
 &\quad + \sum_x \sum_y \sum_z \sum_v \mathcal{F}^i(y, x, z, r, v) d_v^E p_{iy} p_{jx} p_{kz} \\
 &\quad + \sum_x \sum_y \sum_z \sum_u \mathcal{F}^i(y, x, z, u, r) d_u^G p_{iy} p_{jx} p_{kz}
 \end{aligned}$$

28 H. Beigy and M. R. Meybodi

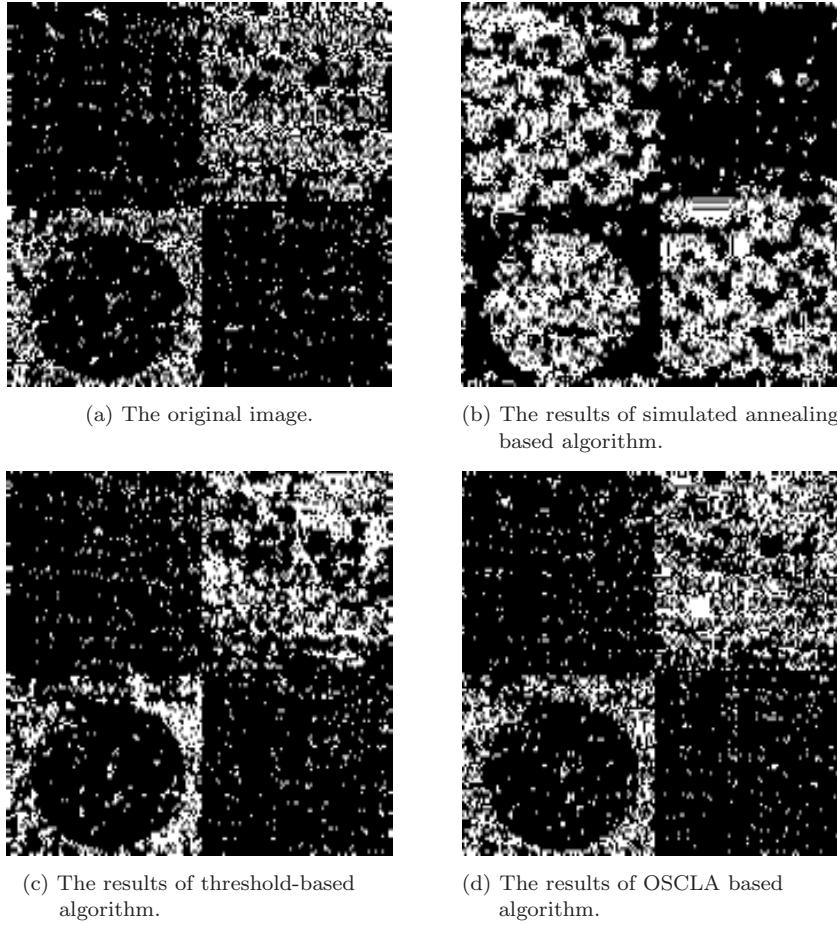


Fig. 10. The original image and the results after segmentation for image 5.

$$\begin{aligned}
&= \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r, x, z, u, v) d_u^G d_v^E p_{jx} p_{kz} \\
&\quad + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r, x, z, u, v) d_u^G d_v^E p_{jx} p_{kz} \\
&\quad + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r, z, x, u, v) d_u^G d_v^E p_{jx} p_{kz} \\
&\quad + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r, x, z, u, v) d_u^G d_v^E p_{jx} p_{kz} \\
&\quad + \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r, x, z, u, v) d_u^G d_v^E p_{jx} p_{kz} \\
&= 5 \sum_x \sum_z \sum_u \sum_v \mathcal{F}^i(r, x, z, u, v) d_u^G d_v^E p_{jx} p_{kz} \\
&= 5d_{lr}(p).
\end{aligned}$$

1 Acknowledgment

The authors thank the reviewers for their very helpful comments and suggestions.

3 References

- [1] Borelijst, M. C. and Hogeweg, P., Spiral wave structure in pre-biotic evolution: Hypercycles stable against parasites, *Physica D* **48** (1991) 17–28.
- [2] Gunji, Y., Pigment color patterns of mulucs as an autonomous process generated by asynchronous automata, *BioSystems* **23** (1990) 317–334.
- [3] Inghe, O., Genet and ramet survivorship under different mortality regimes — a cellular automata model, *Journal of Theoretical Biology* **138** (1989) 257–270.
- [4] Narendra, K. S. and Thathachar, K. S. *Learning Automata: An Introduction* (Prentice-Hall, New York, 1989).
- [5] Srikanthakumar, P. R. and Narendra, K. S., A learning model for routing in telephone networks, *SIAM Journal of Control and Optimization* **20** (1982) 34–57.
- [6] Oommen, B. J. and de St. Croix, E. V., Graph partitioning using learning automata, *IEEE Transactions on Computers* **45** (1996) 195–208.
- [7] Oommen, B. J. and Roberts, T. D., Continuous learning automata solutions to the capacity assignment problem, *IEEE Transactions on Computers* **49** (2000) 608–620.
- [8] Meybodi, M. R. and Beigy, H., A note on learning automata based schemes for adaptation of BP parameters, *Journal of Neurocomputing* **48** (2002) 957–974.
- [9] Meybodi, M. R. and Beigy, H., New learning automata based algorithms for adaptation of backpropagation algorithm parameters, *International Journal of Neural Systems* **12** (2002) 45–68.
- [10] Beigy, H. and Meybodi, M. R., Adaptive uniform fractional channel algorithms, *Iranian Journal of Electrical and Computer Engineering* **3** (2004) 47–53.
- [11] Meybodi, M. R. Beigy, H. and Taherkhani, M., Cellular learning automata and its applications, *Sharif Journal of Science and Technology* **19**(25) (2003) 54–77.
- [12] Meybodi, M. R. and Taherkhani, M., 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*, May 2001, pp. 102–110.
- [13] Meybodi, M. R. Beigy, H. and Taherkhani, M., Application of cellular learning automata to image processing,” in *Proceedings of First Conference in Mathematics and Communication, Iranian Telecommunication Research Center, Tehran, Iran*, Oct. 2000, pp. 23.1–23.10.
- [14] Meybodi, M. R. and Kharazmi, M. R., Image restoration using cellular learning automata, in *Proceedings of the First Iranian Conf. on Machine Vision, Image Processing and Applications, Birjand, Iran*, May 2000, pp. 244–254.
- [15] Kharazmi, M. R. and Meybodi, M. R., Image segmentation using cellular learning Automata, in *Proceedings of 10th Iranian Conference on Electrical Engineering, ICEE-96, Tabriz, Iran*, May 2001.
- [16] Kharazmi, M. R. and Meybodi, M. R., Image restoration using cellular learning automata, in *Proceedings of the Second Iranian Conf. on Machine Vision, Image Processing and Applications, Tehran, Iran*, 2003, pp. 261–270.
- [17] Meybodi, M. R. and Kharazmi, M. R., Application of cellular learning automata to image processing, *Journal of Amirkabir* **14**(564) (2004) 1101–1126.
- [18] Marchini, F. and Meybodi, M. R., Application of cellular learning automata to image processing: finding skeleton, in *Proceedings of the Third Iranian Conf. on Machine Vision, Image Processing and Applications, Iran*, Feb. 2005, pp. 271–280.
- [19] Meybodi, M. R. and Khojasteh, M. R., Application of Cellular Learning Automata in Modelling of Commerce Networks, in *Proceedings of 6th Annual International*

30 H. Beigy and M. R. Meybodi

- 1 Computer Society of Iran Computer Conference CSICC-2001, Isfahan, Iran, Feb.
2001, pp. 284–295.
- 3 [20] Beigy, H. and Meybodi, M. R. *A Self-Organizing Channel Assignment Algorithm: A
Cellular Learning Automata Approach*, Vol. 2690 *Lecture Notes in Computer Science*
(Springer-Verlag, 2003) pp. 119–126.
- 5 [21] Meybodi, M. R. and Mehdipour, F., VLSI placement using cellular learning
Automata,” in *Proceedings of 8th Annual International Computer Society of Iran
Computer Conference CSICC-2001, Mashad, Iran*, Mar. 2003, pp. 195–203.
- 7 [22] Meybodi, M. R. and Beigy, H., A Mathematical Framework for Cellular Learning
Automata,” *Journal of Advances in Complex Systems* **7** (2004) 295–320.
- 9 [23] Kushner, H. J. *Approximation and Weak Convergence Methods for Random Processes*
(MIT Press, Cambridge, 1984).
- 11 [24] Narendra, K. S. and Annaswamy, A. M. *Stable Adaptive Systems* (Prentice-Hall
International Inc., New York, 1989).
- 13