Vol. 7 • Nos. 3 & 4 • September & December 2004

ADVANCES IN
COMPLEX SYSTEMS
A Multidisciplinary Journal

A Mathematical Framework for Cellular Learning Automata

H. Beigy and M. R. Meybodi

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Advances in Complex Systems, Vol. 7, Nos. 3&4 (2004) 295–319

A MATHEMATICAL FRAMEWORK FOR 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
meybodi@ce.aut.ac.ir

Received 14 November 2003
Revised 15 July 2004

The cellular learning automata, which is a combination of cellular automata, and learning automata, is a new recently introduced model. This model is superior to cellular automata because of its ability to learn and is also superior to a single learning automaton because it is a collection of learning automata which can interact with each other. The basic idea of cellular learning automata, which is a subclass of stochastic cellular learning automata, is to use the learning automata to adjust the state transition probability of stochastic cellular automata. In this paper, we first provide a mathematical framework for cellular learning automata and then study its convergence behavior. It is shown that for a class of rules, called commutative rules, the cellular learning automata converges to a stable and compatible configuration. The numerical results also confirm the theoretical investigations.

Keywords: Cellular learning automata; cellular automata; learning automata; interconnected automata.

1. Introduction

Decentralization is a common feature of natural and man-made systems in which, due to large spatial separation of decision makers or limited bandwidth of communication channels, complete information exchange may not be feasible. The decision makers in such a system can gather limited information about each other and the overall system. Hence, the decisions must be made by individual decision makers that have access to partial information regarding the state of the system. Decentralization, by nature, introduces uncertainty into the decision process.

In addition to spatial separation of the system and incomplete information exchange, uncertainties regarding system parameters, control actions taken by other decision makers and external events increase the complexity of decentralized systems. Even in the absence of these uncertainties it is well known that the coordination of decentralized decision makers is a formidable problem.

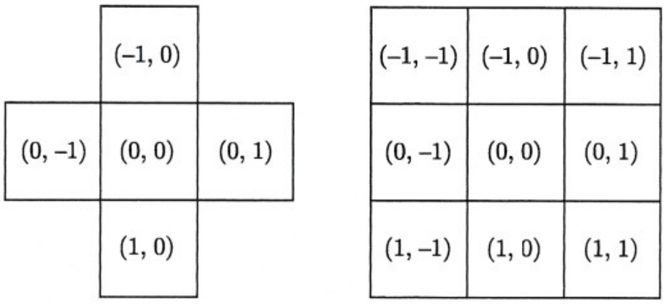
Adaptation (learning) in the decision process overcomes the introduced uncertainty. By using learning, the different decentralized decision makers used in the system attempt to converge to their optimal strategies by improving their performance online, based upon the response of the overall system. Hence, learning can be considered as a critical part for decision makers that have access to the partial information. A subclass of such systems, which are modeled using cellular automata (CA), use information exchange with neighborhood decision makers.

Cellular automata are mathematical models for systems consisting of large numbers of simple identical components with local interactions. CA are non-linear dynamical systems in which space and time are discrete. It is called *cellular* because it is made up cells like points in a lattice or like squares of checker boards, and it is called *automata* because it follows a simple rule [6]. The simple components act together to produce complicated patterns of behavior. Cellular automata perform complex computations with a high degree of efficiency and robustness. They are especially suitable for modeling natural systems that can be described as massive collections of simple objects interacting locally with each other [19, 25]. Informally, a d-dimensional CA consists of an infinite d-dimensional lattice of identical cells. Each cell can assume a state from a finite set of states. The cells update their states synchronously on discrete steps according to a local rule. The new state of each cell depends on the previous states of a set of cells, including the cell itself, and constitutes its neighborhood [7]. The state of all cells in the lattice are described by a configuration. A configuration can be described as the state of the whole lattice. The rule and the initial configuration of the CA specifies the evolution of CA that tells how each configuration is changed in one step. Formally, a CA can be defined as follows:

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

(i) Z^d is a lattice of d-tuples of integer numbers. Each cell in the d-dimensional lattice, Z^d, is represented by a d-tuple $(z_1, z_2, \ldots, z_d)$.
(ii) $\Phi = \{1, \ldots, m\}$ is a finite set of states.
(iii) $N = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{\bar{m}}\}$ is a finite subset of Z^d called the neighborhood vector, where $\bar{x}_i \in Z^d$. The neighborhood vector determines the relative position of the neighboring lattice cells from any given cell u in the lattice Z^d. The neighbors of a particular cell u are the set of cells $\{u + \bar{x}_i | i = 1, 2, \ldots, \bar{m}\}$. We assume that there exists a neighborhood function $\bar{N}(u)$ that maps a cell u to the set of its neighbors, that is

$$\bar{N}(u) = (u + \bar{x}_1, u + \bar{x}_2, \ldots, u + \bar{x}_{\bar{m}}). \tag{1}$$



(a) The von Neuman neighborhood (b) The Moore neighborhood

Fig. 1. The von Neuman and Moore neighborhoods.

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

— $u \in \bar{N}(u)$ for all $u \in Z^d$.

— $u_1 \in \bar{N}(u_2) \Leftrightarrow u_2 \in \bar{N}(u_1)$ for all $u_1, u_2 \in Z^d$.

For example, the neighborhood vectors $N_{vN} = \{(0,0), (1,0), (0,1), (-1,0), (0,-1)\}$ as shown in Fig. 1(a) and $N_M = \{(0,0), (1,0), (0,1), (-1,0), (0,-1), (-1,-1), (-1,1), (1,-1), (1,1)\}$ as shown in Fig. 1(b) are called *von Neuman* and *Moore* neighborhoods, respectively.

(iv) $\mathcal{F}$: $\Phi^{\bar{m}} \to \Phi$ is the local rule of the cellular automata. It computes the new state for each cell from the current states of its neighbors.

Learning in the learning automata (LA) has been studied using the paradigm of an automaton operating in an unknown random environment. In a simple form, the automaton has a finite set of actions to choose from and at each stage, its choice (action) depends upon its action probability vector. For each action chosen by the automaton, the environment gives a reinforcement signal with fixed unknown probability distribution. The automaton then updates its action probability vector depending upon the reinforcement signal at that stage, and evolves to the some final desired behavior. A class of learning automata is called *variable structure learning automata* and are represented by triple $\langle \beta, \alpha, T \rangle$, where β is a set of inputs, α is a set of actions, and T is a learning algorithm [21]. 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. In what follows, two learning algorithms for updating the action probability vector are given. Let α_i be the action chosen at time k as a sample realization from probability distribution $p(k)$. In the linear reward-ϵpenalty algorithm ($L_{R-\epsilon P}$) scheme the recurrence equation for updating p is defined as

$$p_j(k+1) = \begin{cases} p_j(k) + a \times [1 - p_j(k)] & \text{if } i = j, \\ p_j(k) - a \times p_j(k) & \text{if } i \neq j, \end{cases} \tag{2}$$

when $\beta(k) = 0$ and

$$p_j(k+1) = \begin{cases} p_j(k) \times (1-b) & \text{if } i = j, \\ \frac{b}{r-1} + p_j(k)(1-b) & \text{if } i \neq j, \end{cases} \tag{3}$$

when $\beta(k) = 1$. Parameters $0 < b \ll a < 1$ represent *step lengths* and r is the number of actions for LA. The $a(b)$ determines the amount of increase (decrease) of the action probabilities. If $a = b$, then the recurrence equations (2) and (3) are called the *linear reward penalty* (L_{R-P}) algorithm and if $b = 0$, then the recurrence equations (2) and (3) are called the *linear reward inaction* (L_{R-I}) algorithm. LA have been used successfully in many applications such as telephone and data network routing [26], solving NP-Complete problems [23], capacity assignment [24], neural network engineering [12, 13], and cellular networks [1, 2, 4, 5], to mention a few.

Automata are, by design, "simple agents for doing simple things." The full potential of an LA is realized when multiple automata interact with each other. Interaction may assume different forms such as a tree, mesh, array, 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 LA is not necessary and is not natural. Local interaction of LA, which can be defined in the form of a graph such as a tree, mesh, or array, is natural in many applications. On the other hand, CA are mathematical models for systems consisting of large numbers of simple identical components with local interactions. In this paper, we combine the CA and LA 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 LA which can interact with each other. The basic idea of CLA, which is a subclass of stochastic CA, is to use learning automata to adjust the state transition probability of stochastic CA. 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 Ref. 10, an asynchronous CLA with several LA in each cell is given and used as an adaptive controller. In this model, the state space of the system under control is uniformly discretized into cells. The actions of each LA correspond to discritized values of the corresponding control variable. Based on the state of the system (S_0) one cell in the CLA is activated. Every LA of the activated cell chooses an action based on its action probability vector. These actions are applied to the system and the state of the system is changed from S_0 to S_1. The environment then passes a reinforcement signal to the LA of the activated cell. Depending on this signal, LA in the activated cell and its neighboring cells update their action probability vectors. This process continues until the termination state is reached. In Ref. 14, a model of synchronous CLA has been proposed in which each cell can hold one LA. The CLA have been used in many applications such as image processing [8, 9, 14, 16], rumor diffusion [18], modeling of commerce networks [16], channel assignment in cellular networks [3], and VLSI placement [1].

Since the introduction of CLA, it has been used in a number of applications but no mathematical framework for studying its behavior has been developed yet. Having a mathematical framework for CLA enables us to investigate the characteristics of this model deeper, which may help us to find more applications. Having such a mathematical framework also makes it possible to study the previous applications more rigorously and develop better CLA based algorithms for these applications. In this paper, we develop a mathematical framework to study the behavior of the CLA and investigate its convergence properties. It is shown that for class rules, which will be called commutative rules, the CLA converges to a globally stable state.

The rest of this paper is organized as follows. In Sec. 2, the CLA is presented. Section 4 studies the behavior of the cellular learning automata and Sec. 5 studies the behavior of cellular learning automata when commutative rules are used. Section 5 presents a numerical example, and Sec. 6 concludes the paper.

2. Cellular Learning Automata

Cellular learning automata (CLA) (Fig. 2) is a mathematical model for dynamical complex systems that consists of a large number of simple components. The simple components, which have learning capability, act together to produce complicated behavioral patterns. A CLA is a CA in which a learning automaton is assigned to every cell. The learning automaton residing in a particular cell determines its

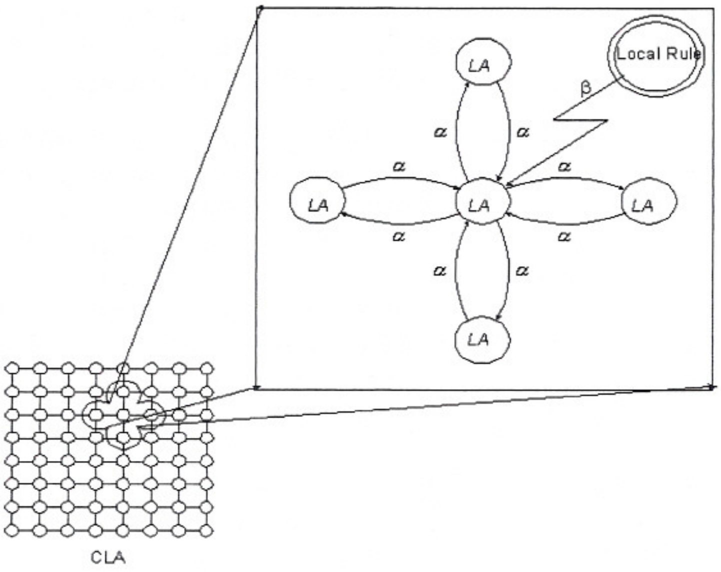


Fig. 2. Cellular learning automata.

state (action) on the basis of its action probability vector. Like CA, there is a rule that the CLA operate under. The rule of the CLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to the LA residing in a cell. The neighboring LAs of any particular LA constitute the local environment of that cell. The local environment of a cell is nonstationary because the action probability vectors of the neighboring LAs vary during evolution of the CLA.

The operation of the CLA can be described as follows: At the first step, the internal state of a cell is specified. The state of every cell is determined on the basis of the action probability vectors of the LA residing in that cell. The initial value of this state may be chosen on the basis of the past experience or at random. In the second step, the rule of the CLA determines the reinforcement signal to the LA residing in the cell. Finally, each LA updates its action probability vector on the basis of the supplied reinforcement signal and the chosen action by the cell. This process continues until the desired result is obtained. A d-dimensional CLA is formally defined below.

Definition 2. 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) $\underline{\Phi}$ is a finite set of states.
(iii) A is the set of LA each of which is assigned to one cell of the CLA.
(iv) $N = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{\bar{m}}\}$ is a finite subset of Z^d called neighborhood vector, where $\bar{x}_i \in Z^d$.
(v) $\mathcal{F}$: $\underline{\Phi}^{\bar{m}} \to \underline{\beta}$ is the local rule of the cellular learning automata, where $\underline{\beta}$ is the set of values that the reinforcement signal can take. It computes the reinforcement signal for each LA based on the actions selected by the neighboring LA.

In what follows, we consider CLA with n cells and the neighborhood function $\bar{N}(i)$. The learning automaton, A_i, which has a finite action set $\underline{\alpha}_i$, is associated to cell i (for $i = 1, \ldots, n$) of the CLA. Let the cardinality of $\underline{\alpha}_i$ be m_i. The state of the CLA 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 A_i.

The operation of the CLA takes place as the following iterations. At iteration k, each learning automaton chooses an action. Let $\alpha_i \in \underline{\alpha}_i$ be the action chosen by A_i. Then all learning automata receive a reinforcement signal. Let $\beta_i \in \underline{\beta}$ be the reinforcement signal received by A_i. This reinforcement signal is produced by the application of local rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) \to \underline{\beta}$. The higher value of β_i means that the chosen action of A_i will receive a higher reward. Since each set $\underline{\alpha}_i$ is finite, rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) \to \underline{\beta}$ can be represented by a hyper matrix of dimensions $m_1 \times m_2 \times \cdots \times m_{\bar{m}}$. These n hyper matrices constitute what we call the rule of the CLA. When all of these n hyper matrices are equal, the CLA is called uniform; otherwise it is called nonuniform. For the sake of simplicity in presentation, the rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}})$ is denoted by $\mathcal{F}^i(\alpha_1, \alpha_2, \ldots, \alpha_{\bar{m}})$. Based on the nature of the set $\underline{\beta}$, the CLA can be classified into three groups: P-model, Q-model, and S-model CLA. When $\underline{\beta} = \{0, 1\}$, we refer to CLA as *P-model CLA*, when $\underline{\beta} = \{b_1, \ldots, b_l\}$, (for $l < \infty$), we refer to CLA as *Q-model CLA*, and when $\underline{\beta} = [b_1, b_2]$, we refer to CLA as *S-model CLA*. If learning automaton A_i uses learning algorithm L_i, we denote CLA by $CLA(L_1, \ldots, L_n)$. If $L_i = L$ for all $i = 1, \ldots, n$, then we denote the CLA by $CLA(L)$.

2.1. *Definitions and notations*

In this section, we first give some definitions and derive some preliminary results used later in this paper for the analysis of the CLA.

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

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

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

and

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

respectively.

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

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

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

$$\underline{p} = \sum_{i_1=1}^{m_{i_1}} \sum_{i_2=1}^{m_{i_2}} \cdots \sum_{i_n=1}^{m_{i_n}} \left((p_{1i_1} \underline{e}_{i_1}), (p_{2i_2} \underline{e}_{i_1}), \ldots, (p_{ni_n} \underline{e}_{i_n}) \right). \tag{4}$$

Since each M-vector $(\underline{e}_{i_1}, \underline{e}_{i_2}, \ldots, \underline{e}_{i_n})$ is in $\mathcal{K}^*$, then the above sum can be interpreted as a convex combination of the elements of $\mathcal{K}^*$. $\square$

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

Definition 5. The global behavior of a CLA is a mapping $\mathcal{G}\colon \mathcal{K} \to \mathcal{K}$ that describes the dynamics of the CLA.

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

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

$$d_{ir}(\underline{p}) = \sum_{y_2} \cdots \sum_{y_{\bar{m}}} \mathcal{F}^i(r, y_2, \ldots, y_{\bar{m}}) \prod_{\substack{l \in \bar{N}(i) \\ l \neq i}} p_{l y_l}, \tag{5}$$

and the average reward for learning automaton A_i is defined as

$$D_i(\underline{p}) = \sum_r d_{ir}(\underline{p}) p_{ir}. \tag{6}$$

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

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

$$\sum_r d_{ir}(\underline{p}) p_{ir} \geq \sum_r d_{ir}(\underline{p}) q_{ir} \tag{7}$$

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 CLA have any reason to change its action.

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

$$\mathcal{D}(\underline{p}) = \sum_i D_i(\underline{p}). \tag{8}$$

Lemma 2. *The CLA has at least one compatible configuration.*

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

$$\bar{p}_{ir} = \frac{p_{ir} + \phi_{ir}}{1 + \sum_{j=1}^{m} \phi_{ij}} \tag{9}$$

for $i = 1, \ldots, n$ and $r = 1, \ldots, m_i$. It is evident that T is a continuous mapping. Since $\mathcal{K}$ is closed, bounded and convex, we can use the *Brouwer's fixed point theorem* to show that every mapping T has at least one fixed point. We now show that every fixed point of T is necessarily a compatible configuration of the CLA and conversely every compatible configuration of the CLA is a fixed point of T, that is, $\underline{p} = T(\underline{p})$ thereby concluding the proof of the lemma. We first verify the latter assertion: if $\underline{p} \in \mathcal{K}$ is a compatible configuration, then for every $\underline{q} \in \mathcal{K}$, we have $\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir}$ for all $i = 1, \ldots, n$. Configuration $\underline{q}$ also includes $\underline{q} = (\underline{p}'_1, \ldots, \underline{e}_{r_i}, \ldots, \underline{p}'_n)'$ for fixed i $(i = 1, \ldots, n)$. Since $d_{ir_i}(\underline{p})$ is independent of $\underline{p}_i$, we obtain $\psi_{ir_i}(\underline{p}) \leq 0$. Hence, $\phi_{ir_i} = 0$ for all $i = 1, \ldots, n$ and $r_i = 1, \ldots, m$ and we have $\underline{p} = T(\underline{p})$, from which we conclude 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 $d_{ir}(\underline{p})$ attains its maximum value. Then $D_i(\underline{\tilde{p}})$ can be bounded from above by $d_{iy_i}(\underline{p})$, thus implying that $\psi_{ir_i}(\underline{p}) > 0$, which implies $\phi_{ir_i}(\underline{p}) > 0$. But since $\phi_{ir_i}(\underline{p})$ is nonnegative for all r_i, then $\sum_j \phi_{ij}(\underline{p}) > 0$. Let r_i $(1 \leq i \leq m_i)$ be an action for which $d_{ir}(\underline{p})$ attains its minimum value. Then by using inequality (10), it can be shown that $D_i(\underline{p})$ is bounded below by $d_{ir_i}(\underline{p})$. This implies $\psi_{iy_i}(\underline{p}) < 0$, which implies $\phi_{iy_i}(\underline{p}) = 0$, which when used in (9) yields the conclusion $\bar{p}_{iy_i} < \tilde{p}_{iy_i}$, because $\sum_j \phi_{ij}(\underline{p}) > 0$, contradicting the hypothesis that $\underline{q}$ is a fixed point of T. □

Lemma 3. *Configuration $\underline{p} \in \mathcal{K}$ is compatible if and only if*

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

for all i and r.

Proof. If $\underline{p} \in \mathcal{K}$ is a compatible configuration, then from (7), for every $\underline{q} \in \mathcal{K}$ and $1 \leq i \leq n$, we have $\sum_r d_{ir}(\underline{p})p_{ir} \geq \sum_r d_{ir}(\underline{p})q_{ir}$. Since, $\underline{q}$ includes $\underline{q} = (\underline{p}'_1, \ldots, \underline{e}_{r_i}, \ldots, \underline{p}'_n)'$ for fixed i $(i = 1, \ldots, n)$ and $d_{ir_i}(\underline{p})$ is independent of $\underline{p}_i$, then we obtain $d_{ir_i}(\underline{p}) \leq D_i(\underline{p})$.

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

Lemma 4. *Let $\underline{p} \in \mathcal{K}$ be a compatible configuration. Then for each i, we have*

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

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

Proof. From Lemma 3, we have

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

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

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

From the above inequality and Eq. (6), 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. □

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 (7), 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 (7) is solved by $\underline{p}$, then for any $\underline{q} \in \mathcal{K}$, fixed l, $1 \leq l \leq n$, and set $\underline{q} = (\underline{p}_1', \ldots, \underline{q}_l', \ldots, \underline{p}_n')'$, we have

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

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 the learning automata except the specific learning automaton A_i are held fixed for some i, then the configuration reached by the CLA 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*

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

for all $r \neq t_i$.

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

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

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 Eq. (6), we obtain

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

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

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

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

Corollary 1. *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* $\mathcal{F}^i(t_1, t_2, \ldots, t_{\bar{m}}) > \mathcal{F}^i(r, t_2, \ldots, t_{\bar{m}})$ *for all* $r \neq t_i$.

Proof. The proof is trivial given the proof of Theorem 2. □

3. Behavior of Cellular Learning Automata

In this section, we analyze the CLA in which all the learning automata use the L_{R-I} learning algorithm. The process $\{\underline{p}(k)\}_{k\geq 0}$ which evolves according to the L_{R-I} learning algorithm is Markovian and can be described by the following difference equation:

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

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 $\underline{p}(k)$. $\underline{g}$ represents the learning algorithm, $\underline{a}$ is a $M \times M$ diagonal matrix with $a_{jj} = a_i$ for $\sum_{l=1}^{i-1} m_l < i \leq \sum_{l=1}^{i} m_l$, and a_i represents the learning parameter for learning automaton A_i. Now, define

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

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 expressed as a function of $\underline{p}(k)$. Hence, we can write

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

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

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

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

$$\underline{\tilde{p}}_i^a(t) = \underline{p}_i(k), \qquad t \in [ka_i, (k+1)a_i], \tag{18}$$

where a_i is the learning parameter of the L_{R-I} algorithm for learning automaton A_i. The interpolated process $\{\underline{\tilde{p}}^a(t)\}_{t \geq 0}$ is a sequence of random variables that takes values from $\mathcal{R}^{m_1 \times \cdots \times m_n}$, where $\mathcal{R}^{m_1 \times \cdots \times m_n}$ is the space of all functions that, at each point, are continuous on the right and have a limit on the left over $[0, \infty)$ and take values in $\mathcal{K}$, which is a bounded subset of $\mathcal{R}^{m_1 \times \cdots \times m_n}$. The objective is to study the limit of sequence $\{\underline{\tilde{p}}^a(t)\}_{t \geq 0}$ as $\max\{\underline{a}\} \to 0$, which will be a good approximation to the asymptotic behavior of (18). When learning parameter a_i is sufficiently small for all $i = 1, 2, \ldots, n$, then Eq. (15) can be written as the following ordinary differential equation (ODE):

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

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

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

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 $\tilde{\underline{p}}^a$ as $\max\{\underline{a}\}$ becomes sufficiently small. We show that the sequence of interpolated processes $\{\tilde{\underline{p}}^a(t)\}$ 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).

Theorem 3. *Sequence $\{\tilde{\underline{p}}^a(.)\}$ 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 $a \to 0$, where $X_0 = \tilde{\underline{p}}^a(0)$ and $a = \max\{\underline{a}\}$.

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

(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;
(iii) $\underline{g}$ is bounded, continuous and independent of a;
(iv) ODE (21) has a unique solution for each initial condition $\underline{X}(0)$;
(v) for a specific configuration, $\underline{p}(k) = \bar{\underline{p}}$, $\{(\underline{\alpha}(k), \underline{\beta}(k))\}_{k\geq 0}$ is an independent identically distributed sequence.

Therefore, using the weak convergence theorem [11], sequence $\{\tilde{\underline{p}}^a(\cdot)\}$ converges weakly, as $\max\{\underline{a}\} \to 0$ 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 $M^{\bar{p}}$ and $M^{\bar{p}}$ is the distribution of process $\{(\underline{\alpha}(k), \underline{\beta}(k))\}_{k\geq 0}$. Since for $\underline{p}(k) = \hat{\underline{p}}$, $(\underline{\alpha}(k), \underline{\beta}(k))$ is an independent identically distributed sequence whose distribution depends only on $\hat{\underline{p}}$ and the rule of the CLA, then we have

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

and hence the theorem is proven. □

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

Lemma 5. *For large k and small enough value of* $\max\{\underline{a}\}$, *the asymptotic behavior of* $\underline{p}(k)$ *generated by the CLA can be approximated by the solution to ODE* (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 $E[h_T(\tilde{\underline{p}})^a] \to E[h_T(\underline{X})] = 0$ as $\max\{\underline{a}\} \to 0$. The limit is zero since the value of $h_T(\underline{X})$ on the paths of limit process is zero with probability one. Thus, the sup 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 a_0 such that

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

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

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 theorem about the convergence of the CLA.

3.1. *Equilibrium points*

The equilibrium points of Eq. (17) are those points that satisfy the set of equations $\Delta p_{ij}(k) = 0$ for all i, j, where the expected changes in the probabilities are zero. In other words, the equilibrium points are zeros of $\underline{f}\,(\underline{p})$, which are studied in the following two lemmas.

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

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

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

Proof. From Eq. (16), it is obvious that $f_{iy} = 0$ (for $i = 1, 2, \ldots, n$) if $\underline{p}_i$ is a unit vector and hence all corners of $\mathcal{K}$ are equilibrium points of $\underline{f}(\cdot)$. In order to find other equilibrium points of $\underline{f}(\cdot)$, from (16) it is obvious that $f_{iy} = 0$ if $p_{iy} = 0$. Since $\underline{p}_i$ is a probability vector, then all components of $\underline{p}_i$ cannot be at the same time zero. Hence, when $p_{iy} \neq 0$, the following equation must hold:

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

The above equation can be rewritten as

$$\begin{aligned}
\sum_{r \neq y} p_{ir}(k) \left[d_{iy}(\underline{p}) - d_{ir}(\underline{p})\right] &= \sum_{r \neq y} p_{ir}(k) d_{iy}(\underline{p}) - \sum_{r \neq y} p_{ir}(k) d_{ir}(\underline{p}) \\
&= d_{iy}(\underline{p}) \left[1 - p_{iy}(k)\right] - \sum_{r \neq y} p_{ir}(k) d_{ir}(\underline{p}) \\
&= d_{iy}(\underline{p}) - \sum_{r} p_{ir}(k) d_{ir}(\underline{p}) \\
&= d_{iy}(\underline{p}) - \sum_{r \neq q} p_{ir}(k) d_{ir}(\underline{p}) - p_{iq}(k) d_{iq}(\underline{p}) \\
&= d_{iy}(\underline{p}) - \sum_{r \neq q} p_{ir}(k) d_{ir}(\underline{p}) - d_{iq}(\underline{p}) \left[1 - \sum_{r \neq q} p_{ir}(k)\right] \\
&= d_{iy}(\underline{p}) - d_{iq}(\underline{p}) + \sum_{r \neq q} \left[d_{iq}(\underline{p}) - d_{ir}(\underline{p})\right] p_{ir}(k) \\
&= 0.
\end{aligned} \tag{24}$$

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

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

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 makes $\underline{f}(\underline{p})$ 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 p_{iy} are zero, for $\underline{f}$ to be zero, Eq. (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 7. *All compatible configurations are equilibrium points of* $\underline{f}(\cdot)$.

Proof. Let $\underline{p}$ be a compatible configuration. Then by Lemma 4, for each i, either $p_{ir} = 0$ or $d_{ir}(\underline{p}) = D_i(\underline{p})$. Hence, $f_{ir}(\underline{p}) = 0$ for all i and r. □

3.2. *The stability property*

In this subsection we characterize the stability of equilibrium configurations of CLA, that is the equilibrium points of the ODE (19). From Lemmas 6 and 7, all the equilibrium points of (19) are known. In order to study the stability of the equilibrium points of (19), 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 8. *A corner $\underline{p}^o \in \mathcal{K}^*$ is a fully compatible configuration if and only if it is uniformly asymptotically stable.*

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

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

the origin is translated to $\underline{p}^0$. Since $\underline{p}_i$ $(1 \le i \le n)$ is a probability vector, then only $\sum_i (m_i - 1)$ components of $\underline{p}^o$ are independent. Suppose that p_{ir} for $r \neq t_i$ (for $1 \le i \le n$) be the independent components. Using a Taylor expansion, f_{iy} can be expressed as[a]

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

We consider the following positive definite Lyapunov function $V(\underline{\tilde{p}}) = \sum_i \sum_{y \neq t_i} \tilde{p}_{iy}$, where $V(\underline{\tilde{p}}) \ge 0$ and is zero when $\tilde{p}_{iy} = 0$ for all i, y, and its derivative is equal to $\dot{V}(\underline{\tilde{p}}) = \sum_i \sum_{y \neq t_i} f_{iy}$. Since corner $\underline{p}^o$ is a fully compatible configuration, then from Theorem 2 we have $\mathcal{F}^i(y, t_2, \ldots, t_{\bar{m}}) - \mathcal{F}^i(t_i, t_2, \ldots, t_{\bar{m}}) < 0$ for $i = 1, 2, \ldots, n$. Thus, Eq. (26) implies that there is a neighborhood around $\underline{p}^o$ such that the linear terms dominate the high order terms. Hence, $\dot{V}(\underline{\tilde{p}}) < 0$ 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 $\dot{\underline{\tilde{p}}} = A\underline{\tilde{p}}$, where $A = \text{diag}(\tilde{f}_{iy})$ and $\tilde{f}_{iy} = \mathcal{F}^i(y, t_2, \ldots, t_{\bar{m}}) - \mathcal{F}^i(t_i, t_2, \ldots, t_{\bar{m}})$ 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. □

[a]The details for derivation of the above equation for linear CLA is given in Appendix.

Lemma 9. *Non-compatible equilibrium points of $\underline{f}(\cdot)$ are unstable.*

Proof. Let $\underline{p}^o$ be an equilibrium point of $\underline{f}(\cdot)$ which is not compatible. Then from Lemma 4, there is a learning automaton A_j and an action y such that $d_{jy}(\underline{p}) > D_j(\underline{p})$. Since $d_{jy}(\underline{p})$ and $D_j(\underline{p})$ are continuous, then inequality $d_{jy}(\underline{p}) > D_j(\underline{p})$ will hold in a 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 1. In Lemmas 8 and 9, the solution of ODE (19) is 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.

Remark 2. An almost sure convergence method [22] can be used to show the convergence of CLA. Using this method, it can be shown that the evolution of CLA essentially follows the solution to the ODE (for large k) and also if the CLA enters the domain of attraction of an asymptotically stable configuration $\underline{p}^o$ infinitely often, it will eventually converges to $\underline{p}^o$. Therefore, if we use almost sure methods then the stability analysis performed above is not needed.

3.3. *Convergence results*

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

(i) $\underline{p}(0)$ is close to a compatible corner $\underline{p}^o$. From Lemma 8, there is a neighborhood around $\underline{p}^o$ entering which, the CLA will be absorbed by that corner. Thus, the CLA converges to a compatible configuration.

(ii) $\underline{p}(0)$ is close to a non-compatible corner $\underline{p}^o$. From Lemma 9, no matter how small the neighborhood we take around $\underline{p}^o$ is, 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 (iv) below.

(iii) $\underline{p}(0) \in \mathcal{K}^*$. Using the convergence properties of the L_{R-I} learning algorithm [21], no matter whether $\underline{p}(0)$ is compatible or not, the CLA will be absorbed to $\underline{p}(0)$.

(iv) $\underline{p}(0) \in \mathcal{K}-\mathcal{K}^*$. The convergence of the CLA for these initial configurations is stated in Theorem 4.

Theorem 4. *Suppose there is a bounded differential function $\mathcal{D}$: $\mathcal{R}^{m_1+\cdots+m_{\bar{m}}} \to \mathcal{R}$ 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. Then the CLA for any initial configuration in $\mathcal{K}-\mathcal{K}^*$ and with a sufficiently small value of the* *learning parameter* ($\max\{\underline{a}\} \to 0$), *always converges to a configuration that is stable and compatible.*

Proof. Consider the variation of $\mathcal{D}$ along the solution paths of ODE (19); $\mathcal{D}$ is non-decreasing because

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

The CLA 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 $\mathcal{R}^{m_1+\cdots+m_{\bar{m}}}$, asymptotically all solutions of ODE (19) will be in $\mathcal{K}$. Inequality (27) shows that CLA 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 $p_{iy} = p_{ir}$. From Lemmas 6 and 7, 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 CLA satisfies the sufficiency conditions needed for Theorem 4, then the CLA will converge to a compatible configuration; otherwise the convergence of the CLA to a compatible configurations cannot be guaranteed and it may exhibit a limit cyclic behavior [20].

4. Cellular Learning Automata Using Commutative Rules

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

Definition 10. A rule $\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}})$ is called commutative if and only if

$$\begin{aligned}
&\mathcal{F}^i(\alpha_{i+\bar{x}_1}, \alpha_{i+\bar{x}_2}, \ldots, \alpha_{i+\bar{x}_{\bar{m}}}) \\
&\quad = \mathcal{F}^i(\alpha_{i+\bar{x}_{\bar{m}}}, \alpha_{i+\bar{x}_1}, \ldots, \alpha_{i+\bar{x}_{\bar{m}-1}}) = \cdots = \mathcal{F}^i(\alpha_{i+\bar{x}_2}, \alpha_{i+\bar{x}_3}, \ldots, \alpha_{i+\bar{x}_1}).
\end{aligned} \tag{28}$$

In order to simplify the algebraic manipulations, we give the analysis for the linear CLA, as shown in Fig. 3, which uses the neighborhood function $\bar{N}(i) = \{i-1, i, i+1\}$. The following theorem is an additional property for compatible configurations when the CLA use a commutative rule.

Theorem 5. *If a CLA uses a commutative rule, then a configuration* $\underline{p}$ *at which* $\mathcal{D}(\underline{p})$ *is a local maximum, is compatible.*

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

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

Thus using chain rule, we obtain $\nabla\mathcal{D}(\underline{p})(\underline{q} - \underline{p}) \le 0$. $\nabla\mathcal{D}(\underline{q})$ has M elements in which (l, r)th component of $\nabla F(\underline{q})$ is denoted by q_{lr} and calculated by the following equation:

$$\begin{aligned}
q_{lr} &= \frac{\partial}{\partial p_{lr}} \sum_i \sum_y \sum_x \sum_z \mathcal{F}^i(y, x, z) p_{jx} p_{iy} p_{kz} \\
&= \sum_i \sum_x \sum_y \sum_z [\mathcal{F}^i(y, x, z)\delta_{lj}\delta_{rx} p_{iy} p_{kz} + \mathcal{F}^i(y, x, z)\delta_{li}\delta_{ry} p_{jx} p_{kz} \\
&\quad + \mathcal{F}^i(y, x, z)\delta_{lk}\delta_{rz} p_{jx} p_{iy}] \\
&= \sum_y \sum_z \mathcal{F}^i(y, r, z) p_{iy} p_{kz} + \sum_x \sum_z \mathcal{F}^i(r, x, z) p_{jx} p_{kz} + \sum_x \sum_y \mathcal{F}^i(y, x, r) p_{jx} p_{iy} \\
&= \sum_x \sum_z \mathcal{F}^i(y, y, z) p_{jx} p_{kz} + \sum_x \sum_z \mathcal{F}^i(r, x, z) p_{jx} p_{kz} + \sum_x \sum_z \mathcal{F}^i(z, x, r) p_{jx} p_{kz} \\
&= 3 \sum_x \sum_z \mathcal{F}^i(r, x, z) p_{jx} p_{kz} \\
&= 3 d_{lr}(\underline{p}),
\end{aligned}$$

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

Fig. 3. The linear CLA.

where $y = \alpha_{i+\bar{x}_1}$, $x = \alpha_{i+\bar{x}_2}$, $z = \alpha_{i+\bar{x}_3}$, $j = i-1$, and $k = i+1$. Using the above result and $\nabla\mathcal{D}(\underline{p})(\underline{q}-\underline{p}) \leq 0$, we have

$$\begin{aligned}\nabla\mathcal{D}(\underline{p})(\underline{q}-\underline{p}) &= 3\sum_i\sum_y\sum_x\sum_z \mathcal{F}^i(y,x,z)p_{jx}p_{kz}[q_{iy}-p_{iy}]\\ &= 3\sum_i\sum_y d_{iy}(\underline{p})[q_{iy}-p_{iy}]\\ &\leq 0,\end{aligned}$$

for all $\underline{q} \in \mathcal{K}$. So, $\underline{p}$ satisfies the condition of Theorem 1, and hence $\underline{p}$ is a compatible configuration. □

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

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

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

Remark 4. From the proof of the Theorem 6, we can conclude that the CLA converges to one of its compatible configurations, if any. If the CLA has one compatible configuration, then CLA converges to this configuration for which $D(\underline{p})$ is the maximum. If there are more than one compatible configurations, then the CLA depending on the initial configuration $\underline{p}(0)$ converges to one of its compatible configurations for which $D(\underline{p})$ is a local maximum.

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

5. Numerical Examples

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

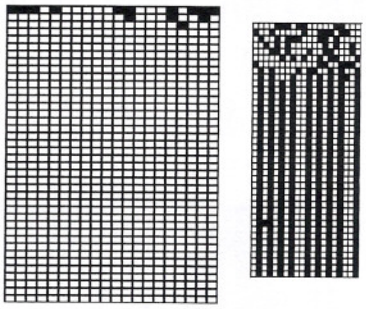
Figures 4 through 6 show the time-space diagram evolution of CLA using commutative rules with 20 cells and a two-action L_{R-I} learning automaton in each cell.

Figure 7 shows the time-space diagram evolution of CLA using noncommutative rules with 20 cells and a two-action L_{R-I} learning automaton in each cell.

Figure 8 shows the time-space diagram evolution of CLA with eight cells and a three-action L_{R-I} learning automaton in each cell.

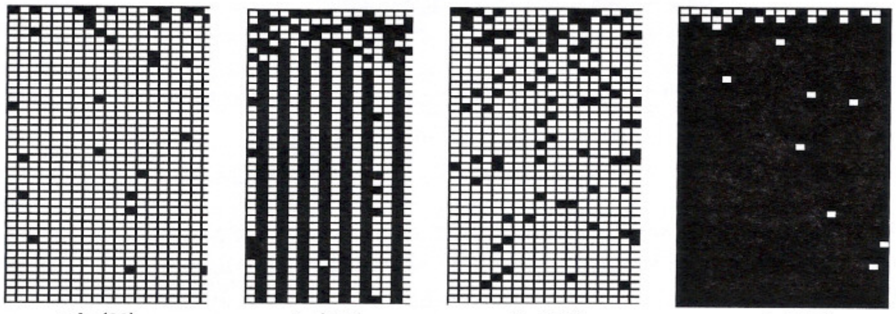
Table 1. The scheme for the rule numbering for two actions learning automata.

Configuration	111	110	101	100	011	010	001	000
Reinforcement signal	0	0	0	1	0	1	1	0



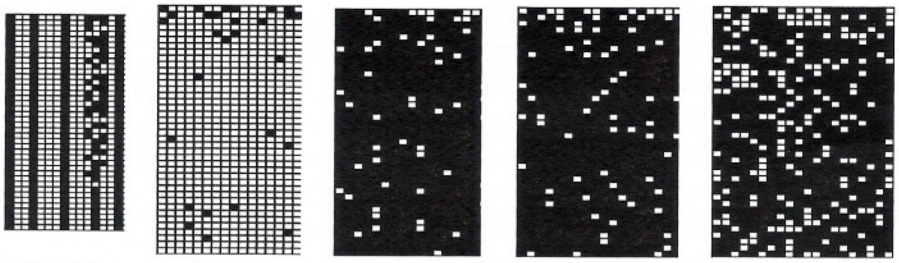
rule $(1)_2$ rule $(22)_2$

Fig. 4. Time-space diagram of synchronous CLA using commutative rules.



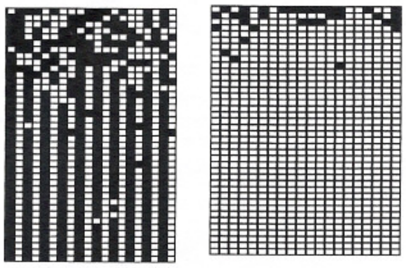
rule $(23)_2$ rule $(126)_2$ rule $(127)_2$ rule $(128)_2$

Fig. 5. Time-space diagram of synchronous CLA using commutative rules.



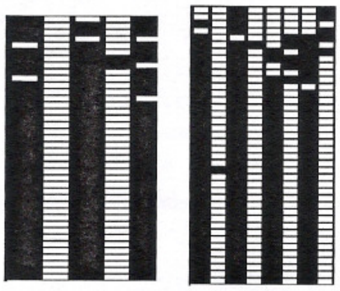
rule $(150)_2$ rule $(151)_2$ rule $(232)_2$ rule $(233)_2$ rule $(254)_2$

Fig. 6. Time-space diagram of synchronous CLA using commutative rules.



rule $(180)_2$ rule $(190)_2$

Fig. 7. Time-space diagram for CLA.



rule $(1562)_3$ rule $(9754)_3$

Fig. 8. Time-space diagram of synchronous CLA using commutative rules.

6. Conclusions

In this paper, a formal description of cellular learning automata has been given and its convergence behavior studied. It has been shown that for commutative rules, the cellular learning automata converges to a stable configuration for which the average reward for the CLA is maximum. The numerical results have also confirmed the theory.

Acknowledgment

This work is partially supported by a grant from School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (No. CS1382-4-05), Tehran, Iran.

Appendix

In this appendix, we give the derivation of the Taylor expansion of $f_{iy}(\underline{p})$ for the linear CLA; let $j = i-1$, $k = i+1$, $y = \alpha_{i+\bar{x}_1}$, $y_j = \alpha_{i+\bar{x}_2}$, and $y_k = \alpha_{i+\bar{x}_3}$.

$$
\begin{aligned}
f_{ir} &= p_{ir}(k)\sum_{y\neq r} p_{iy}(k)[d_{ir}(\underline{p}) - d_{iy}(\underline{p})] \\
&= p_{ir}\sum_{\substack{y_i\neq r\\ y_i\neq t_i}} p_{ir}\sum_{y_j\neq t_j}\Bigg[\sum_{y_k\neq t_k}\mathcal{F}^i(y_i,y_j,y_k)p_{ky_k} \\
&\quad + \mathcal{F}^i(y_i,y_j,t_k)\Bigg(1-\sum_{y_k\neq t_k}p_{ky_k}\Bigg)\Bigg]p_{jy_j} \\
&\quad - p_{ir}\sum_{\substack{y_i\neq r\\ y_i\neq t_i}} p_{ir}\sum_{y_j\neq t_j}\Bigg[\sum_{y_k\neq t_k}\mathcal{F}^i(r,y_j,y_k)p_{ky_k} \\
&\quad + \mathcal{F}^i(r,y_j,t_k)\Bigg(1-\sum_{y_k\neq t_k}p_{ky_k}\Bigg)\Bigg]p_{jy_j} \\
&\quad + p_{ir}\sum_{\substack{y_i\neq r\\ y_i\neq t_i}} p_{ir}\Bigg[\sum_{y_k\neq t_k}\mathcal{F}^i(y_i,t_j,y_k)p_{ky_k} \\
&\quad + \mathcal{F}^i(y_i,t_j,t_k)\Bigg(1-\sum_{y_k\neq t_k}p_{ky_k}\Bigg)\Bigg]\Bigg(1-\sum_{y_j\neq t_j}p_{jy_j}\Bigg) \\
&\quad - p_{ir}\sum_{\substack{y_i\neq r\\ y_i\neq t_i}} p_{ir}\Bigg[\sum_{y_k\neq t_k}\mathcal{F}^i(r,t_j,y_k)p_{ky_k} \\
&\quad + \mathcal{F}^i(r,t_j,t_k)\Bigg(1-\sum_{y_k\neq t_k}p_{ky_k}\Bigg)\Bigg]\Bigg(1-\sum_{y_j\neq t_j}p_{jy_j}\Bigg) \\
&\quad + p_{ir}\Bigg(1-\sum_{y_i\neq t_i}p_{iy_i}\Bigg)\sum_{y_j\neq t_j}\Bigg[\sum_{y_k\neq t_k}\mathcal{F}^i(t_i,y_j,y_k)p_{ky_k} \\
&\quad + \mathcal{F}^i(t_i,y_j,t_k)\Bigg(1-\sum_{y_k\neq t_k}p_{ky_k}\Bigg)\Bigg]p_{jy_j}
\end{aligned}
$$

$$-p_{ir}\left(1-\sum_{y_i\neq t_i}p_{iy_i}\right)\sum_{y_j\neq t_j}\left[\sum_{y_k\neq t_k}\mathcal{F}^i(t_i,t_j,y_k)p_{ky_k}\right.$$
$$\left.+\mathcal{F}^i(t_i,t_j,t_k)\left(1-\sum_{y_k\neq t_k}p_{ky_k}\right)\right]p_{jy_j}$$
$$+p_{ir}\left(1-\sum_{y_i\neq t_i}p_{iy_i}\right)\left[\sum_{y_k\neq t_k}\mathcal{F}^i(t_i,t_j,y_k)p_{ky_k}\right.$$
$$\left.+\mathcal{F}^i(t_i,t_j,t_k)\left(1-\sum_{y_k\neq t_k}p_{ky_k}\right)\right]\left(1-\sum_{y_j\neq t_j}p_{jy_j}\right)$$
$$-p_{ir}\left(1-\sum_{y_i\neq t_i}p_{iy_i}\right)\left[\sum_{y_k\neq t_k}\mathcal{F}^i(t_i,t_j,y_k)p_{ky_k}\right.$$
$$\left.+\mathcal{F}^i(t_i,t_j,t_k)\left(1-\sum_{y_k\neq t_k}p_{ky_k}\right)\right]\left(1-\sum_{y_j\neq t_j}p_{jy_j}\right).$$

Using the transformation

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

to change the origin to $\underline{p}^0$, f_{iy} can be approximated linearly:

$$f_{iy}=\tilde{p}_{iy}[\mathcal{F}^i(t_{i-1},y,t_{i+1})-\mathcal{F}^i(t_{i-1},t_i,t_{i+1})]+\text{higher order terms.} \quad (30)$$

References

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

[2] Beigy, H. and Meybodi, M. R., A Learning Automata Based Dynamic Guard Channel Scheme, Vol. 2510 of *Springer-Verlag Lecture Notes in Computer Science* (Springer-Verlag, 2002), pp. 643–650.

[3] Beigy, H. and Meybodi, M. R., A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach, *Springer-Verlag Lecture Notes in Computer Science*, Vol. 2690 (Springer-Verlag, 2003), pp. 119–126.

[4] Beigy, H. and Meybodi, M. R., An Adaptive Uniform Fractional Guard Channel Algorithm: A Learning Automata Approach, *Springer-Verlag Lecture Notes in Computer Science*, Vol. 2690 (Springer-Verlag, 2003), pp. 405–409.

[5] Beigy, H. and Meybodi, M. R., Adaptive uniform fractional channel algorithms, *Iranian J. Elec. Comp. Eng.* **3**, 47–53 (2004).

[6] Fredkin, E., Digital machine: A informational process based on reversible cellular automata, *Physica* **D45**, 254–270 (1990).

[7] Kari, J., Reversability of 2D cellular automata is undecidable, *Physica* **D45**, 379–385 (1990).

[8] Kharazmi, M. R. and Meybodi, M. R., Image Segmentation Using Cellular Learning Automata, in *Proc. 10th Iranian Conf. on Electrical Engineering, ICEE-96*, Tabriz, Iran, May 2001.

[9] Kharazmi, M. R. and Meybodi, M. R., Image Restoration Using Cellular Learning Automata, in *Proc. 2nd Iranian Conf. on Machine Vision, Image Processing and Applications*, Tehran, Iran, 2003, pp. 261–270.

[10] Krishna, K., Cellular learning automata: A stochastic model for adaptive controllers, Master's thesis, Department of Electrical Engineering, Indian Institute of Science, Banglore, India, June 1993.

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

[12] Meybodi, M. R. and Beigy, H., New learning automata based algorithms for adaptation of backpropagation algorithm parameters, *I. J. Neural Systems* **12**, 45–68 (2002).

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

[14] Meybodi, M. R., Beigy, H. and Taherkhani, M., Cellular Learning Automata and its Applications, *J. Science Tech. Sharif* (Sharif University of Technology, Tehran, Iran) **19**(25), 54–77 (2004).

[15] Meybodi, M. R. and Kharazmi, M. R., Image Restoration Using Cellular Learning Automata, in *Proc. 1st Iranian Conf. on Machine Vision, Image Processing and Applications*, Birjand, Iran, May 2000, pp. 244–254.

[16] Meybodi, M. R. and Khojasteh, M. R., Application of Cellular Learning Automata in Modelling of Commerce Networks, in *Proc. 6th Annual Int. Computer Society of Iran Computer Conf. CSICC-2001*, Isfehan, Iran, February 2001, pp. 284–295.

[17] Meybodi, M. R. and Mehdipour, F., VLSI Placement Using Cellular Learning Automata, in *Proc. 8th Annual Int. Computer Society of Iran Computer Conf. CSICC-2001*, Mashad, Iran, March 2003, pp. 195–203.

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

[19] Mitchell, M., Computation in cellular automata: A selected review, Technical report, Santa Fe Institute, Santa Fe, NM, USA, Sept. 1996.

[20] Narendra, K. S. and Annaswarmy, A. M., *Stable Adaptive Systems* (Prentice-Hall, New York, 1989).

[21] Narendra, K. S. and Thathachar, K. S., *Learning Automata: An Introduction* (Prentice-Hall, New York, 1989).

[22] Norman, M. F., *Markovian Process and Learning Models* (Academic Press, New York, 1972).

[23] Oommen, B. J. and de St. Croix, E. V., Graph partitioning using learning automata, *IEEE Trans. Computers* **45**, 195–208 (1996).

[24] Oommen, B. J. and Roberts, T. D., Continuous learning automata solutions to the capacity assignment problem, *IEEE Trans. Computers* **49**, 608–620 (2000).

[25] Packard, N. H. and Wolfram, S., Two-dimensional cellular automata, *J. Stat. Phys.* **38**, 901–946 (1985).

[26] Srikantakumar, P. R. and Narendra, K. S., A learning model for routing in telephone networks, *SIAM Journal of Control and Optimization* **20**, 34–57 (1982).