Journal of Computational Science 24 (2018) 358–370

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Dynamic irregular cellular learning automata

Mehdi Esnaashari [a,*], Mohammad Reza Meybodi [b]

[a] Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran
[b] Soft Computing Laboratory, Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 14 March 2017
Received in revised form 9 July 2017
Accepted 17 August 2017
Available online 11 September 2017

Keywords:
Learning automata
Cellular learning automata
Irregular cellular learning automata
Dynamic irregular cellular learning automata
Expediency

ABSTRACT

Irregular Cellular learning automaton (*ICLA*), which is recently introduced, is a cellular learning automaton (*CLA*) with irregular structure. ICLA is suitable for modeling problems which are not regular in nature, such as problems in the area of sensor networks, web mining, and grid computing. In some areas such as mobile ad hoc and sensor networks, where the structure of the environment changes over the time, an ICLA with a dynamic structure is required. For this reason, in this paper, we have extended ICLA in such a way that the structure of the extended model, called dynamic ICLA (*DICLA*), can change over time. For the newly introduced model, we have proposed the concept of expediency and then, discussed sufficient conditions under which a DICLA becomes expedient.

1. Introduction

Cellular learning automaton (*CLA*) [1,4–7] is a combination of cellular automaton (*CA*) [2] and learning automaton (*LA*) [3]. The basic idea of CLA is to use LA to adjust the state transition probability of stochastic CA. This model, which opens a new learning paradigm, is superior to CA because of its ability to learn and also is superior to single LA because it consists of a collection of LAs interacting with each other [4]. In [4], authors have introduced a mathematical framework for studying the behavior of the CLA. They have shown that, for a class of rules called commutative rules, different models of CLA converge to a globally stable state [4–7].

Irregular Cellular learning automaton (*ICLA*), which is recently introduced, is a cellular learning automaton (*CLA*) with irregular structure [8]. ICLA is suitable for modeling problems which are not regular in nature, such as problems in the area of sensor networks, web mining, and grid computing. ICLA has been used in many different applications including: clustering of sensor networks [9], sleep/wake scheduling of sensor nodes [10], graph coloring [11,12], finding the shortest path in stochastic graphs [13], and channel assignment in mobile ad hoc networks [14].

In many research domains, such as web mining, computer networks, social networks, and so on, problems can be modeled by graphs, and in some of such domains, the structure of the graph may change in time. For example, in a computer network, especially mobile networks, new links may become available between nodes or some existing links may be removed. In a social network, people may find new friends or lose their ex-friends. To be able to cope with such changes within the environment of the problem, the model must be able to change its structure. Therefore, in this paper, we introduce a new version of ICLA, called dynamic irregular cellular learning automaton (*DICLA*), in which, unlike ICLA, the structure may change over the course of time. Underlying an ICLA model, there is a graph which represents the neighborhood relationship between cells of the ICLA. The structure of this graph is fixed for the ICLA, but in the proposed DICLA model, the graph may change in time, resulting in dynamicity in the neighbors of each cell; that is, during the passage of time, two cells may become neighbors or two neighboring cells may not be further neighbors. We have successfully applied the DICLA to the deployment problem in mobile wireless sensor networks [15].

We have also studied the steady-state behavior of the proposed DICLA. For the first, we introduce the concept of expediency for DICLA and discuss conditions under which a DICLA becomes expedient. Informally, a DICLA is said to be expedient if, in the long run, all of its constituting learning automata perform better than pure-chance automata. A pure-chance automaton is an automaton which chooses each of its available actions by pure chance. Expediency is a notion of learning in that any automaton that is said to learn must then do at least better than a pure-chance automaton [16].

* Corresponding author.
E-mail addresses: esnaashari@kntu.ac.ir (M. Esnaashari), mmeybodi@aut.ac.ir (M.R. Meybodi).

https://doi.org/10.1016/j.jocs.2017.08.012
1877-7503/

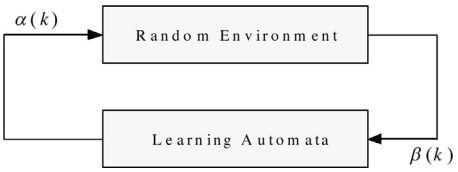


Fig. 1. Relationship between learning automata and its environment.

The rest of this paper is organized as follows. Section 2 briefly introduces the ICLA model. In Sections 3 and 4, we present DICLA and its steady state behavior, respectively. A set of numerical examples will be given in Section 5 for illustrating the theoretical results. Section 6 is the conclusion.

2. Irregular cellular learning automata: an extension to the CLA model

Before presenting the ICLA model in this section, we first give a brief introduction to the CA, LA, and CLA models.

2.1. Cellular automata

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. The states of all cells in the lattice are described by a configuration. A configuration can be described as the state of the whole lattice. The local rule and the initial configuration of the CA specify the evolution of CA, that is, how the configuration of the CA evolves in time.

2.2. Learning automata

Learning automata (*LA*) is an abstract model which randomly selects one action out of its finite set of actions and performs it on a random environment. Environment then evaluates the selected action and responses to the LA with a reinforcement signal. Based on the selected action, and the received signal, the LA updates its internal state and selects its next action. Fig. 1 depicts the relationship between an LA and its environment.

Environment can be defined by the triple $E = \{\alpha, \beta, c\}$where $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ represents a finite input set, $\beta = \{\beta_1, \beta_2, ..., \beta_r\}$ represents the output set, and $c = \{c_1, c_2, ..., c_r\}$ is a set of penalty probabilities, where each element $c_i = E[\beta \mid \alpha = \alpha_i]$ of c corresponds to one input α_i. An environment in which β can take only binary values 0 or 1 is referred to as P-model environment. A further generalization of the environment allows finite output sets with more than two elements that take values in the interval [0,1]. Such an environment is referred to as Q-model. Finally, when the output of the environment is a continuous random variable which assumes values in the interval [0,1], it is referred to as an S-model. LAs are classified into fixed-structure stochastic, and variable-structure stochastic. In the following, we consider only variable-structure LAs.

A variable-structure LA is defined by the quadruple $\{\alpha, \beta, p, T\}$ in which $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ represents the action set of the LA, $\beta = \{\beta_1, \beta_2, ..., \beta_r\}$ represents the input set, $p = \{p_1, p_2, ..., p_r\}$ represents the action probability set, and finally $p(k+1) = T\left[\alpha(k), \beta(k), p(k)\right]$ represents the learning algorithm. This LA operates as follows. Based on the action probability set p, LA randomly selects an action $\alpha(k)$, and performs it on the environment. After receiving the environment's reinforcement signal ($\beta(k)$), LA updates its action probability set based on Eq. (1) or (2) according to the selected action $\alpha(k)$.

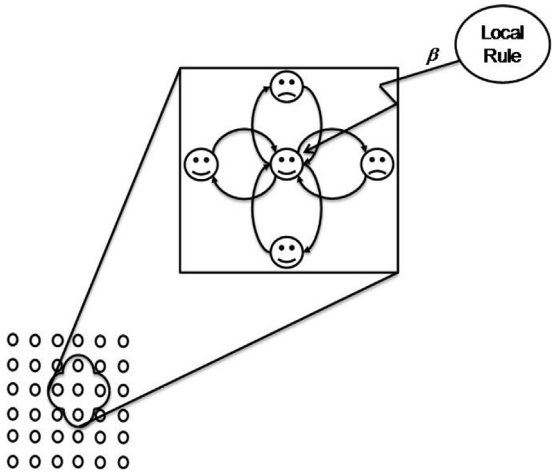


Fig. 2. Operation of the cellular learning automaton.

$$p_i(k+1) = p_i(k) + \beta(k)\left[\frac{b}{r-1} - b \cdot p_i(k)\right] - \left[1 - \beta(k)\right] \cdot a \cdot p_i(k), \quad when\, \alpha(k) \neq \alpha_i. \tag{1}$$

$$p_i(k+1) = p_i(k) - \beta(k) \cdot b \cdot p_i(k) + \left[1 - \beta(k)\right] \cdot a \cdot (1 - p_i(k)), \quad when\, \alpha(k) = \alpha_i. \tag{2}$$

In the above equations, a and b are reward and penalty parameters respectively. For $a = b$, learning algorithm is called L_{RP},[1] for $b \ll a$, it is called $L_{R\varepsilon P}$,[2] and for $b = 0$, it is called L_{RI}.[3]

2.3. Cellular learning automata

A CLA is a CA in which a number of LAs is assigned to every cell. Each LA residing in a particular cell determines its action (state) on the basis of its action probability vector. Like CA, there is a local rule that the CLA operates under. The local rule of the CLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to that LA. The neighboring LAs (cells) of any particular LA (cell) constitute the local environment of that LA (cell). The local environment of an LA (cell) is non-stationary due to the fact that the action probability vectors of the neighboring LAs vary during the evolution of the CLA. The operation of a CLA could be described as the following steps (Fig. 2): At the first step, the internal state of every cell is determined on the basis of the action probability vectors of the LAs residing in that cell. In the second step, the local rule of the CLA determines the reinforcement signal to each LA residing in that cell. Finally, each LA updates its action probability vector based on the supplied reinforcement signal and the chosen action. This process continues until the desired result is obtained. Some dynamic models of the CLA have been recently introduced in the literature [17,18] and have been utilized for solving problems in peer-to-peer networks.

2.4. Irregular cellular learning automata

Irregular cellular learning automaton (Fig. 3) is a generalization of the CLA in which the restriction of regular structure is removed.

[1] Linear Reward-Penalty.
[2] Linear Reward epsilon Penalty.
[3] Linear Reward Inaction.

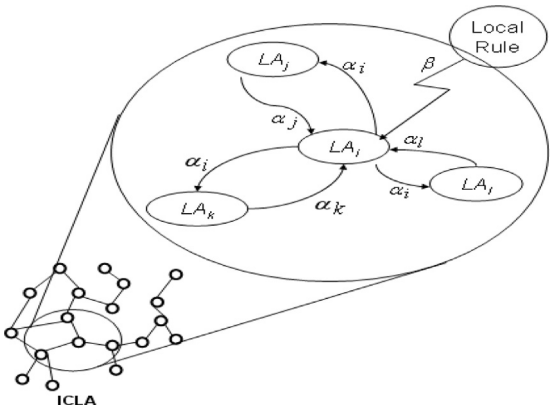


Fig. 3. Irregular cellular learning automaton.

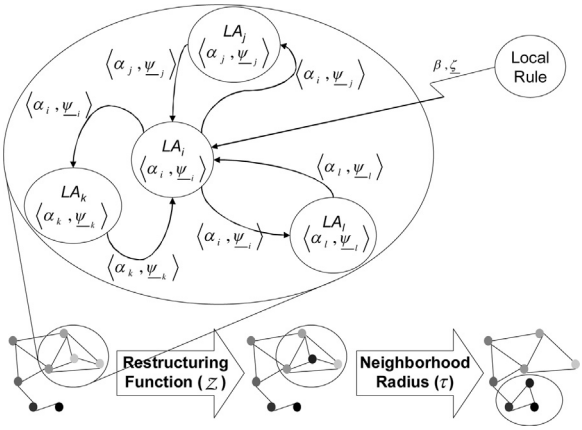


Fig. 4. Dynamic irregular cellular learning automaton.

An ICLA is defined as an undirected graph in which, each vertex represents a cell and is equipped with an LA, and each edge induces an adjacency relation between two cells (two LAs). The LA residing in a particular cell determines its state (action) according to its action probability vector. Like CLA, there is a rule that the ICLA operates under. The rule of the ICLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to that LA.

3. Dynamic irregular CLA

Dynamic irregular cellular learning automaton is a generalization of ICLA with a dynamic structure. A DICLA is defined as an undirected graph in which, each vertex represents a cell and a learning automaton is assigned to every cell (vertex). A finite set of interests is defined for DICLA. For each cell of DICLA a tendency vector is defined whose jth element shows the degree of tendency of that cell to the jth interest. In DICLA, the state of each cell consists of two parts; the action selected by the learning automaton and the tendency vector. Two cells are neighbors in DICLA if the distance between their tendency vectors is smaller than or equal to the neighborhood radius.

Like CLA, there is a local rule that DICLA operates under. The rule of DICLA and the actions selected by the neighboring learning automata of any particular learning automaton LA_i determine the followings: 1. the reinforcement signal to the learning automaton LA_i, and 2. the restructuring signal to the cell in which LA_i resides. Restructuring signal is used to update the tendency vector of the cell. Dynamicity of DICLA is the result of modifications made to the tendency vectors of its constituting cells. Fig. 4 gives a schematic of DICLA. A DICLA is formally defined below. Table 1 provides the definitions of major symbols.

Definition 1. Dynamic irregular cellular learning automaton is a structure $\mathrm{A} = \left(\mathcal{G}\langle V, E\rangle, \Psi, A, \Phi, \varphi\langle\alpha, \underline{\psi}\rangle, \tau, \mathcal{F}, \mathrm{Z}\right)$, where

- $\mathcal{G}$ is an undirected graph, with V as the set of vertices (cells) and E as the set of edges (adjacency relations).
- Ψ is a finite set of interests. Cardinality of Ψ is denoted by $|\Psi|$.
- A is the set of learning automata each of which is assigned to one cell of DICLA.
- Φ is a finite set of actions.
- $\varphi < \alpha, \quad \underline{\psi} >$ is the cell state. State of a cell c_i (φ_i) consists of two parts; 1. $\underline{\alpha}_i$ which is the action selected by the learning automaton of that cell, and 2. A vector $\underline{\psi}_i = \left(\psi_{i1}, \psi_{i2}, \ldots, \psi_{i|\Psi|}\right)^T$ called the tendency vector of the cell. Each element $\psi_{ik} \in [0, \quad 1]$ in the tendency vector of the cell c_i shows the degree of tendency of c_i to the interest $\psi_k \in \Psi$.
- τ is the neighborhood radius. Two cells c_i and c_j of DICLA are neighbors if $\|\underline{\psi}_i - \underline{\psi}_j\| \leq \tau$. In other words, two cells of DICLA are neighbors if the distance between their tendency vectors is smaller than or equal to τ.
- $\mathcal{F}^i : \underline{\varphi}_i \rightarrow \langle\underline{\beta}, \quad [0, \quad 1]^{|\Psi|}\rangle$ is the local rule of DICLA in each cell c_i, where $\underline{\varphi}_i = \left\{\varphi_j | \|\underline{\psi}_i - \underline{\psi}_j\| \leq \tau\right\} \cup \left\{\varphi_i\right\}$ is the set of states of all neighbors of c_i, $\underline{\beta}$ is the set of values that the reinforcement signal can take, and $[0, \quad 1]^{|\Psi|}$ is a $|\Psi|$-dimensional unit hypercube. From the current states of the neighboring cells of each cell c_i, local rule performs the followings: 1. gives the reinforcement signal to the learning automaton LA_i resides in c_i, and 2. produces a restructuring signal ($\underline{\zeta}_i = \left(\zeta_{i1}, \zeta_{i2}, \ldots, \zeta_{i|\Psi|}\right)^T$) which is used to change the tendency vector of c_i. Each element $\zeta_{ij}$of the restructuring signal is a scalar value within the close interval $[-1, \quad 1]$.
- $\mathrm{Z} : [0, 1]^{|\Psi|} \times [-1, 1]^{|\Psi|} \rightarrow [0, \quad 1]^{|\Psi|}$ is the restructuring function which modifies the tendency vector of the cell c_i using the restructuring signal produced by the local rule of the cell.

In what follows, we consider DICLA with n cells. The learning automaton LA_i which has a finite action set $\underline{\alpha}_i$ is associated to the cell c_i (for $i = 1, 2, \ldots, N$) of DICLA. Let the cardinality of $\underline{\alpha}_i$ be m_i.

The operation of DICLA 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 LA_i. Then, each learning automaton receives a reinforcement signal. Let $\beta_i \in \underline{\beta}$ be the reinforcement signal received by LA_i. This reinforcement signal is produced by the application of the local rule $\mathcal{F}^i : \underline{\varphi}_i \rightarrow \langle\underline{\beta}, \quad [0, \quad 1]^{|\Psi|}\rangle$. Higher values of β_i mean that the selected action of LA_i will receive higher penalties. Each LA updates its action probability vector on the basis of the supplied reinforcement signal and the action chosen by the cell. Next, each cell c_i updates its tendency vector using the restructuring function Z (Eq. (3)).

$$\underline{\psi}_i(k+1) = \mathrm{Z}\left(\underline{\psi}_i(k), \underline{\zeta}_i(k)\right) \tag{3}$$

Like ICLA, DICLA can be either synchronous or asynchronous and an asynchronous DICLA can be either time-driven or step-driven.

3.1. Evolution of the DICLA

Definition 2. A configuration of DICLA at step k is denoted by $\langle\underline{p}(k), \underline{\psi}(k)\rangle = \langle\left(\underline{p}_1, \ldots, \underline{p}_N\right)^T, \left(\underline{\psi}_1, \ldots, \underline{\psi}_N\right)^T\rangle$, where $\underline{p}_i$ is the action probability vector of the learning automaton LA_i and $\underline{\psi}_i$ is the tendency vector of the cell c_i.

Table 1
Table of symbols.

Symbol	Definition
K	The probability space
Y	The tendency space
$DICLA^{\underline{\psi}^*}$	DICLA, whose evolution within the tendency space Y is stopped at the point $\underline{\psi}^*$
c_i	Cell indexed by i in the DICLA
$\varphi < \alpha,\ \underline{\psi} >$	Cell state
α_i	Action selected by learning automaton LA_i
$\underline{\psi}_i = \left(\psi_{i1}, \psi_{i2}, \ldots, \psi_{i\lvert\Psi\rvert}\right)^T$	Tendency Vector
Ψ	Set of interests
τ	Neighborhood radius
$\underline{\varphi}_i = \left\{\varphi_j \mid \lVert\underline{\psi}_i - \underline{\psi}_j\rVert \le \tau\right\} \cup \left\{\varphi_i\right\}$	Set of states of all neighbors of c_i
$\beta(k)$	Reinforcement signal
$\underline{\beta}$	The set of values that the reinforcement signal can take
$\underline{\zeta}_i = \left(\zeta_{i1}, \zeta_{i2}, \ldots, \zeta_{i\lvert\psi\rvert}\right)^T$	Restructuring signal
$\mathcal{F}^i : \underline{\varphi}_i \to \langle \underline{\beta},\ [0,\ 1]^{\lvert\Psi\rvert}\rangle$	Local rule of the DICLA in each cell c_i
$\mathcal{F}_\beta$	Reinforcement signal generator part of the local rule
$\mathcal{F}_\zeta$	Restructuring signal generator part of the local rule
$\mathcal{Z} : [0,1]^{\lvert\Psi\rvert} \times [-1,1]^{\lvert\Psi\rvert} \to [0,\ 1]^{\lvert\Psi\rvert}$	The restructuring function
$\langle \mathcal{K}^*, \mathcal{Y}^* \rangle$	The set of all unit configurations
$\langle \underline{p}(k), \underline{\psi}(k)\rangle = \langle \left(\underline{p}_1, \ldots, \underline{p}_N\right)^T,$	Configuration of the DICLA at step k
$N^{\underline{\psi}}(i)$	Neighborhood set of any particular LA_i in configuration $\langle \underline{p}, \underline{\psi}\rangle$
$d_{ir}^{\beta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$	The average penalty for action r of the learning automaton LA_i for configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle \mathcal{K}, \mathcal{Y}\rangle$
$D_i\left(\langle p, \psi\rangle\right)$	The average penalty for the learning automaton LA_i at configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle \mathcal{K}, \mathcal{Y}\rangle$
$D\left(\langle p, \psi\rangle\right)$	The total average penalty for DICLA at configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle \mathcal{K}, \mathcal{Y}\rangle$
$d_{ij}^{\zeta}\left(\langle p, \psi\rangle\right)$	The average restructuring signal for interest j of the cell c_i for configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle \mathcal{K}, \mathcal{Y}\rangle$

Definition 3. A configuration $\langle \underline{p}, \underline{\psi}\rangle$ is called unit if the action probability vector of each learning automaton and the tendency vector of each cell are unit vectors, otherwise it is called general. Hence, the set of all unit configurations, $\langle K^*, Y^*\rangle$, and the set of all general configurations, $\langle K, Y\rangle$, in DICLA are

$$\langle K^*, Y^*\rangle = \left\{ \langle \underline{p}, \underline{\psi}\rangle \mid \begin{array}{c} \underline{p} = \left(\underline{p}_1, \ldots, \underline{p}_N\right)^T, \underline{p}_i = \left(p_{i1}, \ldots, p_{im_i}\right)^T, \forall i, y:\ p_{iy} \in \left\{0,1\right\}, \forall i: \sum_y p_{iy} = 1 \\ \underline{\psi} = \left(\underline{\psi}_1, \ldots, \underline{\psi}_N\right)^T, \underline{\psi}_i = \left(\psi_{i1}, \ldots, \psi_{i\lvert\Psi\rvert}\right)^T, \forall i, j:\ \psi_{ij} \in \left\{0,1\right\}, \forall i: \sum_j \psi_{ij} = 1 \end{array} \right\}, \quad (4)$$

and

$$\langle K, Y\rangle = \left\{ \langle \underline{p}, \underline{\psi}\rangle \mid \begin{array}{c} \underline{p} = \left(\underline{p}_1, \ldots, \underline{p}_N\right)^T, \underline{p}_i = \left(p_{i1}, \ldots, p_{im_i}\right)^T, \forall i, y:\ 0 \le p_{iy} \le 1, \forall i: \sum_y p_{iy} = 1 \\ \underline{\psi} = \left(\underline{\psi}_1, \ldots, \underline{\psi}_N\right)^T, \underline{\psi}_i = \left(\psi_{i1}, \ldots, \psi_{i\lvert\Psi\rvert}\right)^T, \forall i, j:\ 0 \le \psi_{ij} \le 1 \end{array} \right\} \quad (5)$$

respectively. We refer to K as the probability space and to Y as the tendency space of DICLA hereafter.

Lemma 1. $\langle K, Y\rangle$ is the convex hull of $\langle K^*, Y^*\rangle$.

Proof. Proof of this lemma is similar to the proof of Lemma 1 given in [4].■

The application of the local rule to every cell allows transforming a configuration to a new one. The local rule of DICLA ($\mathcal{F}$) consists of two parts; reinforcement signal generator ($\mathcal{F}_\beta$) and restructuring signal generator ($\mathcal{F}_\zeta$).

Definition 4. The global behavior of a DICLA is a mapping G: <K,Y> → <K,Y> that describes the dynamics of DICLA. The evolution of DICLA from a given initial configuration $\langle \underline{p}(0), \underline{\psi}(0)\rangle \in \langle K, Y\rangle$ is a sequence of configurations $\left\{\langle \underline{p}(k), \underline{\psi}(k)\rangle\right\}_{k\ge 0}$, such that $\langle \underline{p}(k+1), \underline{\psi}(k+1)\rangle = G\left(\langle \underline{p}(k), \underline{\psi}(k)\rangle\right)$.

Definition 5. Neighborhood set of any particular LA_i in configuration $\langle \underline{p}, \underline{\psi}\rangle$, denoted by $N^{\underline{\psi}}(i)$, is defined as the set of all learning automata residing in the adjacent cells of the cell c_i, that is,

$$N^{\underline{\psi}}(i) = \left\{ LA_j \mid \lVert\underline{\psi}_i - \underline{\psi}_j\rVert \le \tau \right\}. \quad (6)$$

Let $N_i^{\underline{\psi}}$ be the cardinality of $N^{\underline{\psi}}(i)$.

Definition 6. The average penalty for action r of the learning automaton LA_i for configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle K, Y\rangle$ is defined as

$$\begin{aligned} d_{ir}^{\beta}\left(\langle \underline{p}, \underline{\psi}\rangle\right) &= E\left[\beta_i \mid \langle \underline{p}, \underline{\psi}\rangle, \alpha_i = r\right] \\ &= \sum_{y_{h_1}^{\underline{\psi}}, \ldots, y_{h_{N_i^{\underline{\psi}}}}^{\underline{\psi}}} \left(F_\beta^i\left(y_{h_1}^{\underline{\psi}}, \ldots, y_{h_{N_i^{\underline{\psi}}}}^{\underline{\psi}}, r\right) \prod_{LA_l \in N^{\underline{\psi}}(i)} p_{l y_{h_l}^{\underline{\psi}}} \right), \end{aligned} \quad (7)$$

and the average penalty for the learning automaton LA_i is defined as

$$D_i\left(\langle \underline{p}, \underline{\psi}\rangle\right) = E\left[\beta_i | \langle \underline{p}, \underline{\psi}\rangle\right] = \sum_y d_{iy}^{\beta}\left(\langle \underline{p}, \underline{\psi}\rangle\right) p_{iy}. \quad (8)$$

The above definition implies that if the learning automaton LA_j is not a neighboring learning automaton for LA_i, then $d_{ir}^{\beta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$ does not depend on $\underline{p}_j$. We assume that $d_{ir}^{\beta}\left(\langle \underline{p}, \underline{\psi}\rangle\right) \neq 0$ for all i, r and $\langle \underline{p}, \underline{\psi}\rangle$, that is, in any configuration, any action has a non-zero chance of receiving penalty.

Definition 7. The total average penalty for DICLA at configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle \mathrm{K}, \mathrm{Y}\rangle$ is the sum of the average penalties for all learning automata in DICLA, that is,

$$D\left(\langle \underline{p}, \underline{\psi}\rangle\right) = \sum_i D_i\left(\langle \underline{p}, \underline{\psi}\rangle\right). \quad (9)$$

Definition 8. The average restructuring signal for interest j of the cell c_i for configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle \mathrm{K}, \mathrm{Y}\rangle$ is defined as

$$d_{ij}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right) = \sum_r \sum_{y_{h_1}^{\underline{\psi}}, \ldots, y_{h_{N_i^{\underline{\psi}}}}^{\underline{\psi}}} \left(\mathrm{F}_{\zeta}^{i}\left(y_{h_1}^{\underline{\psi}}, y_{h_2}^{\underline{\psi}}, \ldots, y_{h_{N_i^{\underline{\psi}}}}^{\underline{\psi}}, r, \underline{\psi}_{*j}\right) \prod_{LA_l \in N^{\underline{\psi}}(i)} p_{l y_{h_l}^{\underline{\psi}}}\right), \quad (10)$$

where $\underline{\psi}_{*j} = \left(\psi_{1c}, \psi_{1j}, \ldots, \psi_{Nj}\right)^T$.

The above definition implies that all interests are independent of each other, since the restructuring signal generator $(\mathcal{F}_{\zeta})$ for any interest of any cell depends only on the tendency values of the neighboring cells to that interest.

4. Dynamics of DICLA

Dynamics of DICLA within its environment is twofold; dynamics of the tendency vectors of its constituting cells and dynamics of the action probability vectors of its constituting learning automata. We refer to the former as the structural dynamics and to the latter as the behavioral dynamics of DICLA. In this section, we will first study the structural dynamics of DICLA and then analyzing its behavioral dynamics.

4.1. Structural dynamics of DICLA

Structural dynamics of DICLA can be described by the process $\left\{\underline{\psi}(k)\right\}_{k\geq 0}$ which evolves according to the following equation:

$$\underline{\psi}(k+1) = \underline{\mathrm{Z}}\left(\underline{\psi}(k), \underline{\zeta}(k)\right), \quad (11)$$

where $\underline{\mathrm{Z}}$ represents the restructuring function. In this paper, we study the structural dynamics of DICLA for two different restructuring functions, namely simple restructuring function (Eq. (12)) and vanishing restructuring function (Eq. (13)).

$$\underline{\psi}(k+1) = \underline{\psi}(k) + \underline{b} \cdot \underline{\zeta}(k). \quad (12)$$

$$\underline{\psi}(k+1) = \underline{\psi}(k) + \underline{b} \cdot \nu(k) . \underline{\zeta}(k). \quad (13)$$

In the above equations, $\underline{b}$ is an $n \times n$ diagonal matrix with $b_{ii} = b_i$ and $b_i > 0$ represents the restructuring rate for the cell c_i. $\nu(k)$ in Eq. (13), is a vanish at infinity function. A function ν is said to vanish at infinity if $\nu(k) \to 0$ as $k \to \infty$. Note that the above restructuring functions can be used only if for every i and j we have $-\psi_{ij} \leq b_i \cdot \zeta_{ij} \leq 1 - \psi_{ij}$.

4.1.1. Structural dynamics of DICLA using simple restructuring function

Components of $\underline{\mathrm{Z}}$, using simple restructuring function, can be obtained as follows:

$$\mathrm{Z}_{ij}\left(\psi_{ij}, \zeta_{ij}\right) = \psi_{ij} + b_i \cdot \zeta_{ij}. \quad (14)$$

From Eq. (11) it follows that $\left\{\underline{\psi}(k)\right\}_{k\geq 0}$ is a discrete-time Markov process [19] defined on the tendency space Y.

Now consider the following assumptions:

(A-1) There exists a set of fitness evaluation functions ($\mathcal{F}^i$, $i = 1, 2, \ldots, N$), each is able to evaluate the normalized fitness of the tendency vector of the cell c_i within the range [0,1], where 0 is the fitness of the optimum tendency vector of the cell.

(A-2) The restructuring signal generator $(\mathcal{F}_{\zeta})$ can be described as the following equation:

$$\mathcal{F}_{\zeta}^{i}\left(y_{h_1}^{\underline{\psi}}, y_{h_2}^{\underline{\psi}}, \ldots, y_{h_{N_r^{\underline{\psi}}}}^{\underline{\psi}}, r, \underline{\psi}_{*j}\right) = \mathcal{F}^{i}\left(\underline{\psi}_i\right) \cdot \mathcal{G}^{i}\left(y_{h_1}^{\underline{\psi}}, y_{h_2}^{\underline{\psi}}, \ldots, y_{h_{N_r^{\underline{\psi}}}}^{\underline{\psi}}, r, \underline{\psi}_{*j}\right), \quad (15)$$

where $\mathcal{G}^i$ is a function within the range $[-1, 1]$.

(A-3) $\sum_j \left[\mathcal{G}^{i}\left(y_{h_1}^{\underline{\psi}}, y_{h_2}^{\underline{\psi}}, \ldots, y_{h_{N_r^{\underline{\psi}}}}^{\underline{\psi}}, r, \underline{\psi}_{*j}\right)\right]^2 \neq 0.$

Remark 1. Assumption (A-1) enables us to define a local optimum for the tendency vector of each cell of DICLA. The study of the structural dynamics of DICLA is then reduced to investigating whether the tendency vector of each cell converges to its local optimum or not. We refer to a point $\underline{\psi}^*$ within the tendency space of DICLA as the maximal tendency point, if for every cell c_i, $\underline{\psi}_i^*$ is the local optimum of the tendency vector of that cell.

Remark 2. Assumption (A-2) indicates that F_{ζ}^{i} is a multiplication of $\mathcal{F}^i$ and $\mathcal{G}^i$. Considering this assumption, $\mathcal{G}^i$ specifies the orientation and direction of the restructuring signal and $\mathcal{F}^i$ specifies its magnitude. When the tendency vector of a cell c_i is far from its optimal vector, $\mathcal{F}^i$ is large and as a result, the restructuring signal will be large. On the other hand, when the tendency vector of a cell c_i is near its optimal vector, $\mathcal{F}^i$ is small and hence, the restructuring signal will be small.

Remark 3. Assumption (A-3) indicates that the value of $\mathcal{G}^i$ must be non-zero for at least one of the interests. To explain the reason for this assumption, we have to note that $\mathcal{G}^i$ specifies the orientation and direction of the restructuring signal, and thus its value cannot be zero for all interests; otherwise it cannot specify any direction or orientation.

Using the above assumptions, the average restructuring signal for the interest j of the cell c_i for configuration $\langle \underline{p}, \underline{\psi}\rangle \in \langle \mathrm{K}, \mathrm{Y}\rangle$, defined previously through Eq. (10), can be rewritten as given below:

$$d_{ij}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right) = \mathcal{F}^{i}\left(\underline{\psi}_i\right) \cdot \sum_r \sum_{y_{h_1}^{\underline{\psi}}, \ldots, y_{h_{N_i^{\underline{\psi}}}}^{\underline{\psi}}} \left(\mathcal{G}^{i}\left(y_{h_1}^{\underline{\psi}}, y_{h_2}^{\underline{\psi}}, \ldots, y_{h_{N_i^{\underline{\psi}}}}^{\underline{\psi}}, r, \underline{\psi}_{*j}\right) \prod_{LA_l \in N^{\underline{\psi}}(i)} p_{l y_{h_l}^{\underline{\psi}}}\right) = \mathcal{F}^{i}\left(\underline{\psi}_i\right) \cdot \overline{\mathcal{G}}^{ij}\left(\langle \underline{p}, \underline{\psi}\rangle\right). \quad (16)$$

Now define

$$\Delta\underline{\psi}(k) = E\left[\underline{\psi}(k+1) | \underline{\psi}(k)\right] - \underline{\psi}(k). \quad (17)$$

Since $\left\{\underline{\psi}(k)\right\}_{k\geq 0}$ is Markovian and $\underline{\zeta}(k)$ depends only on $\langle \underline{p}(k), \underline{\psi}(k)\rangle$ and not on k explicitly, then $\Delta\underline{\psi}(k)$ can be expressed as a function of $\langle \underline{p}(k), \underline{\psi}(k)\rangle$. Hence we can write

$$\Delta\underline{\psi} = \underline{b}\, \underline{f}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right). \quad (18)$$

The components of $\Delta\underline{\psi}(k)$ can be obtained as follows:

$$\begin{aligned}\Delta\psi_{ij} &= b_i \cdot f_{ij}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)\\ &= b_i \cdot d_{ij}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)\\ &= b_i \cdot \mathcal{F}^i\left(\underline{\psi}_i\right)\cdot \overline{\mathcal{G}}^{ij}\left(\langle \underline{p}, \underline{\psi}\rangle\right).\end{aligned} \tag{19}$$

Lemma 2. Function $\underline{f}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$ whose components are given by Eq. (19) is Lipschitz continuous over the compact space $\langle \mathrm{K}, \mathrm{Y}\rangle$.

Proof. Function $\underline{f}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$ has compact support (it is defined over $\langle \mathrm{K}, \mathrm{Y}\rangle$), is bounded because $\forall \underline{p}, \underline{\psi}, i, j : -1 \le f_{ij}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right) \le 1$, and is also continuously differentiable with respect to $\underline{\psi}$ over $\langle \mathrm{K}, \mathrm{Y}\rangle$. Therefore, its first derivative with respect to $\underline{\psi}$ is also bounded. Thus, using the Cauchy's mean value theorem, it can be concluded that $\underline{f}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$ is Lipschitz continuous over the compact space $\langle \mathrm{K}, \mathrm{Y}\rangle$ with Lipschitz constant $K = \sup_{\underline{\psi}} \|\nabla_{\underline{\psi}} \underline{f}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)\|$.■

For different values of $\underline{b}$, Eq. (12) generates a different process and we shall use $\underline{\psi}^{\underline{b}}(k)$ to denote this process whenever the value of $\underline{b}$ is to be specified explicitly. To find the approximating ordinary differential equation (ODE) for the restructuring function given by (12), we define a sequence of continuous-time interpolation of (12), denoted by $\tilde{\underline{\psi}}^{\underline{b}}(t)$ and called an *interpolated process*, whose components are defined by:

$$\tilde{\underline{\psi}}_i^{\underline{b}}(t) = \underline{\psi}_i(k), \quad t \in [kb_i, (k+1)b_i), \tag{20}$$

where b_i is the restructuring rate of the simple restructuring function for the cell c_i. The interpolated process $\left\{\tilde{\underline{\psi}}^{\underline{b}}(t)\right\}_{t\ge 0}$ is a sequence of random variables that takes values from $[0,1]^{N\times|\Psi|}$, where $[0,1]^{N\times|\Psi|}$ 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 Y. Consider the following ODE:

$$\dot{\underline{\psi}} = \underline{f}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right), \tag{21}$$

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

$$\frac{d\psi_{ij}}{dt} = \mathcal{F}^i\left(\underline{\psi}_i\right)\cdot \overline{\mathcal{G}}^{ij}\left(\langle \underline{p}, \underline{\psi}\rangle\right) \tag{22}$$

In the following theorem, we will show that Eq. (12) is the approximation to the ODE (21). This means that if we have the solution to (21), then we can obtain information regarding the behavior of $\underline{\psi}(k)$.

Theorem 1. Using the simple restructuring function (12) and considering $\max\left\{\underline{b}\right\} \to 0$, $\underline{\psi}(k)$ is well approximated by the solution of the ODE (21).

Proof. Following conditions are satisfied by the restructuring function given by Eq. (12):

- $\left\{\underline{\psi}(k)\right\}_{k\ge 0}$ is a Markovian process.
- Given $\underline{\psi}(k)$, $\underline{\zeta}(k)$ is independent of $\underline{\zeta}(k-1)$.
- $\underline{f}^{\zeta}\left(\langle \underline{p}(k), \underline{\psi}(k)\rangle\right)$ is independent of k.
- $\underline{f}^{\zeta}\left(\langle \underline{p}(k), \underline{\psi}(k)\rangle\right)$ is Lipshitz continuous over the compact space $\langle \mathrm{K}, \mathrm{Y}\rangle$ (Lemma 2).
- $E\|\underline{\zeta}(.) - \underline{f}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right)\|^2$ is bounded since $\underline{\zeta}(.) \in [-1,1]^{N\times|\Psi|}$.
- Restructuring rates b_i, $i=1, \ldots, N$ are sufficiently small since $\max\left\{\underline{b}\right\} \to 0$.

Therefore, using theorem (A.1) in [20], we can conclude the theorem. ■

Eq. (12) is the so called Euler approximation to the ODE (21). Specifically, if $\underline{\psi}(k)$ is a solution to (12) and $\tilde{\underline{\psi}}(t)$ is a solution to (21), then for any $T>0$, we have

$$\lim_{b_i\to 0}\sup_{0\le k\le T/b_i}\|\underline{\psi}_{ij}(k) - \tilde{\underline{\psi}}(kb_i)\| = 0. \tag{23}$$

What Theorem 1 says is that $\underline{\psi}(k)$, given by Eq. (12), will closely follow the solution of the ODE (21), that is $\underline{\psi}(k)$ can be made to closely approximate the solution of its approximating ODE by taking $\max\left\{\underline{b}\right\}$ sufficiently small. Thus, if the ODE (21) has a locally asymptotically stable equilibrium point, then we can conclude that (by taking $\max\left\{\underline{b}\right\}$ sufficiently small), $\underline{\psi}(k)$, for large k, would be close to this equilibrium point if $\underline{\psi}(0)$ is within the region of attraction of that point.

To find the equilibrium points of the ODE (19), we have to solve equations of the following from:

$$\frac{d\psi^* ij(t)}{dt} = 0 \quad \textit{for all } i, j. \tag{24}$$

Using Eq. (22), (24) can be rewritten as

$$\mathcal{F}^i\left(\underline{\psi}_i^*\right)\cdot \overline{\mathcal{G}}^{ij}\left(\langle \underline{p}, \underline{\psi}^*\rangle\right) = 0, \quad \textit{for all } i, j. \tag{25}$$

Considering the assumption (A-3), the only solutions to the above equations are the set of tendency vectors $\underline{\psi}^*$ which satisfy the following equations:

$$\mathcal{F}^i\left(\underline{\psi}_i^*\right) = 0, \quad \textit{for every } i. \tag{26}$$

In other words, the equilibrium points of the ODE (19) coincide with the maximal tendency points of DICLA.

Since $\underline{\psi}^*$ is an absorbing state of the Markovian process $\left\{\underline{\psi}(k)\right\}_{k\ge 0}$, if $\underline{\psi}(0) = \underline{\psi}^*$, then DICLA, from the structural point of view, absorbed to $\underline{\psi}^*$. But if $\underline{\psi}(0) \ne \underline{\psi}^*$, then we cannot guarantee the convergence to a maximal tendency point. However, we give a sufficient condition under which convergence to a maximal tendency point can be assured.

Theorem 2. Suppose there is a bounded differentiable function $\mathrm{H} : \mathrm{R}^{m_1+\cdots+m_N} \times \mathrm{R}^{N\times|\Psi|} \to \mathrm{R}$ such that for some constants $c_{ij} > 0$, we have $\frac{\partial \mathcal{H}\left(\langle \underline{p}, \underline{\psi}\rangle\right)}{\partial p_{iy}} = 0$ and $\frac{\partial \mathcal{H}\left(\langle \underline{p}, \underline{\psi}\rangle\right)}{\partial \psi_{ij}} = -c_{ij}\cdot \overline{\mathcal{G}}^{ij}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$. Then DICLA using the simple restructuring function and with sufficiently small value of the restructuring rate ($\max\left\{\underline{b}\right\} \to 0$), converges, from the structural points of view, to a maximal tendency point, for any initial configuration in $\langle \mathrm{K}, \mathrm{Y}\rangle$.

Proof. We have

$$\begin{aligned}\frac{d\mathcal{H}}{dt} &= \sum_i\sum_y \frac{\partial \mathcal{H}}{\partial p_{iy}}\cdot\frac{dp_{iy}}{dt} + \sum_i\sum_j \frac{\partial \mathcal{H}}{\partial \psi_{ij}}\cdot\frac{d\psi_{ij}}{dt}\\ &= -\sum_i\sum_j c_{ij}\cdot\mathcal{F}^i\left(\underline{\psi}_i\right)\cdot\left(\overline{\mathcal{G}}^{ij}\left(\langle \underline{p}, \underline{\psi}\rangle\right)\right)^2 \le 0.\end{aligned} \tag{27}$$

Thus, H is non-increasing along the trajectories of the ODE (21) and eventually converges to its minimum, where $\frac{d\mathrm{H}}{dt} = 0$. From (27), the derivative of H is zero if and only if $\mathcal{F}^i\left(\underline{\psi}_i\right) = 0$ for all i. This is an equilibrium point of the ODE (21). Thus, the ODE has to converge to some equilibrium point. Since an equilibrium point of the ODE (21) is also a maximal tendency point, the theorem follows. ■

To see some samples of the H function with the properties specified in Theorem 2, please refer to Table 2.

Table 2
Some samples of the Hfunction with the properties specified in Theorem 2.

$\overline{\mathcal{G}}^{ij}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$	$\mathcal{H}\left(\langle \underline{p}, \underline{\psi}\rangle\right)$	$\frac{\partial \mathcal{H}\left(\langle \underline{p}, \underline{\psi}\rangle\right)}{\partial \psi_{ij}}$
k_{ij}(for some constants $k_{ij} > 0$)	$-\sum_i \sum_j k_{ij} \cdot \psi_{ij}$	$-k_{ij}$
$k_{ij} \cdot \mathcal{N}_i^{\underline{\psi}}$(for some constants kij > 0)	$-\sum_i \sum_{l=1}^{N_i^{\underline{\psi}}} \sum_j k_{ij} \cdot \psi_{ij}$	$-k_{ij} \cdot \mathcal{N}_i^{\underline{\psi}}$

4.1.2. Structural dynamics of DICLA using vanishing restructuring function

Components of $\underline{Z}$, using vanishing restructuring function (Eq. (13)), can be obtained as given below:

$$Z_{ij}\left(\psi_{ij}, \zeta_{ij}\right) = \psi_{ij} + b_i \cdot \nu \cdot \zeta_{ij}. \tag{28}$$

As it was mentioned earlier, $\left\{\underline{\psi}(k)\right\}_{k\geq 0}$ is a discrete-time Markov process [20] defined on the tendency space Y. Once again, consider the following difference equation:

$$\Delta\underline{\psi}(k) = E\left[\underline{\psi}(k+1)\,|\underline{\psi}(k)\right] - \underline{\psi}(k). \tag{29}$$

The components of $\Delta\underline{\psi}(k)$ using vanishing restructuring function can be obtained as follows:

$$\Delta\psi_{ij} = b_i \cdot \nu \cdot d_{ij}^{\zeta}\left(\langle \underline{p}, \underline{\psi}\rangle\right). \tag{30}$$

Since $\nu(k) \to 0$ as $k \to \infty$, we can conclude that, for large k, $\Delta\psi_{ij}(k) \to 0$. This means that in the long run, the process $\left\{\underline{\psi}(k)\right\}_{k\geq 0}$ will eventually converges to a state $\underline{\psi}^*$, but no statement can be made about the properties of this state.

Remark 4. Note that the vanishing restructuring function must be used only if no fitness evaluation function with the properties specified in assumption (A-1) of the previous section can be found. In other words, when it is possible to find some fitness evaluation functions for evaluating the fitness of the tendency vectors of the cells of DICLA, it is rational to use such functions for guiding the exploration of DICLA through its tendency space until it reaches a well-fitted state. But, when it is hard or impossible to find such fitness evaluation functions, one can use the vanishing restructuring function to force DICLA to stop its exploration through its tendency space after a specific number of iterations. The rationale behind using a vanishing restructuring function is that explorations through tendency space, when thinking of the applications modeled by DICLA (such as mobile ad hoc and sensor networks), usually impose some costs (such as energy used for movements of mobile nodes), and such costs cannot be paid forever.

4.2. Behavioral dynamics of DICLA

The behavioral dynamics of DICLA is affected by the structural dynamics of DICLA. This is due to the fact that the local rule of DICLA, which rules its behavioral and structural dynamics, in each cell is a function of the state of that cell and the states of its neighboring cells. But the neighboring cells of each cell change according to the structural dynamics of DICLA and thus, the behavioral dynamics of DICLA is affected accordingly. Considering this affection, we study the behavioral dynamics of DICLA from the time instant at which its evolution within the tendency space Y is stopped at the point $\underline{\psi}^*$. We refer to DICLA, whose evolution within the tendency space Y is stopped at the point $\underline{\psi}^*$, as $DICLA^{\underline{\psi}^*}$.

In this section, we will study the asymptotic behavior of $DICLA^{\underline{\psi}^*}$, in which all the learning automata use the L_{RP} learning algorithm [3], when operating within an S-model environment. We refer to such a $DICLA^{\underline{\psi}^*}$ as $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata hereafter. The process $\left\{\underline{p}(k)\right\}_{k\geq 0}$ which evolves according to the L_{RP} learning algorithm can be described by the following difference equation:

$$\underline{p}(k+1) = \underline{p}(k) + \underline{a} \cdot \underline{g}\left(\underline{p}(k), \underline{\beta}(k)\right), \tag{31}$$

where $\underline{\beta}(k)$ is composed of the components $\beta_{iy}(k)$ (for $1 \leq i \leq N$, $1 \leq y \leq m_i$, and $\beta_{iy_1}(k) = \beta_{iy_2}(k)$ for every $1 \leq y_1, y_2 \leq m_i$), which are dependent on $\underline{p}(k)$. $\underline{a}$ is an $N \times N$ diagonal matrix with $a_{ii} = a_i$ and a_i represents the learning parameter for the learning automaton LA_i. $\underline{g}$ represents the learning algorithm, whose components can be obtained using L_{RP} learning algorithm in S-model environment as follows:

$$\underline{g}_{ir}\left(p_{ir}, \beta_{ir}\right) = \begin{cases} \left(1 - p_{ir} - \beta_{ir}\right); \alpha_i = r \\ \left(\dfrac{\beta_{ir}}{m_i - 1} - p_{ir}\right); \alpha_i \neq r \end{cases}. \tag{32}$$

From Eq. (31) it follows that $\left\{\underline{p}(k)\right\}_{k\geq 0}$ is a discrete-time Markov process [12] defined on the probability space K (given by Eq. (5)). Let (K, d) be a metric space, where d is the metric defined according to the Eq. (33) as given below:

$$\mathrm{d}\left(\underline{p}, \underline{q}\right) = \sum_i \|\underline{p}_i - \underline{q}_i\|, \tag{33}$$

where $\|\underline{X}\|$ stands for the size of the vector $\underline{X}$.

Lemma 3 and Lemma 4, given below, state some properties of the Markovian process given by Eq. (31).

Lemma 3. The Markovian process given by Eq. (31) is strictly distance diminishing.

Proof. To prove this lemma, we will show that the Markovian process given by Eq. (31) follows the definition of the strictly distance diminishing processes given in [21] by Norman. The complete proof is given in Appendix A. ■

Corollary 1. Let $\underline{p}^{(h)}$ denotes $\underline{p}(k+h)$ when $\underline{p}(k) = \underline{p}$ and $\underline{q}^{(h)}$ denotes $\underline{p}(k+h)$ when $\underline{p}(k) = \underline{q}$. Then $\underline{p}^{(h)} \to \underline{q}^{(h)}$ as $h \to \infty$ irrespective of the initial configurations $\underline{p}$ and $\underline{q}$.

Proof. The proof is given in Appendix A. ■

Lemma 4. The Markovian process given by Eq. (31) is ergodic.

Proof. To prove the lemma we can see that the Markovian process given by Eq. (31) has the following two properties:

- There are no absorbing states for $\left\{\underline{p}(k)\right\}$. This is because there is no $\underline{p}$ that satisfies $\underline{p}(k+1) = \underline{p}(k)$.
- The proposed process is strictly distance diminishing (Lemma 3).

From the above two properties and considering the results given in Corollary 1, we can conclude that the Markovian process $\left\{\underline{p}(k)\right\}_{k\geq 0}$ is ergodic. ■

Now define

$$\Delta\underline{p}(k) = E\left[\underline{p}(k+1)\,|\underline{p}(k)\right] - \underline{p}(k). \tag{34}$$

Since $\left\{\underline{p}(k)\right\}_{k\geq 0}$ is Markovian and $\underline{\beta}(k)$ depends only on $\underline{p}(k)$ and $\underline{\psi}^*$ and not on k explicitly, then $\Delta\underline{p}(k)$ can be expressed as a function of $\underline{p}(k)$ and $\underline{\psi}^*$. Hence we can write

$$\Delta\underline{p}(k) = \underline{a}\underline{f}^{\beta}\left(\langle \underline{p}, \underline{\psi}^*\rangle\right). \tag{35}$$

The components of $\Delta\underline{p}(k)$can be obtained as follows:

$$\begin{aligned}\Delta p_{ir} &= a_i p_{ir}\cdot\left[1-p_{ir}-E\left[\beta_{ir}\right]\right]-a_i\sum_{y\neq r}p_{iy}\cdot\left[\frac{1}{m_i-1}E\left[\beta_{iy}\right]-p_{ir}\right]\\ &= a_i\cdot\left[\frac{1}{m_i-1}\sum_{y\neq r}p_{iy}E\left[\beta_{iy}\right]-p_{ir}E\left[\beta_{ir}\right]\right]\\ &= a_i\cdot\left[\frac{1}{m_i-1}\sum_{y\neq r}p_{iy}d_{iy}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)-p_{ir}d_{ir}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)\right]\\ &= a_i f_{ir}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right),\end{aligned}\tag{36}$$

where

$$\begin{aligned}f_{ir}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right) &= \frac{1}{m_i-1}\sum_{y\neq r}p_{iy}d_{iy}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)-p_{ir}d_{ir}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)\\ &= \frac{1}{m_i-1}\sum_{y}p_{iy}d_{iy}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)-\left(1+\frac{1}{m_i-1}\right)\cdot\left[p_{ir}d_{ir}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)\right]\\ &= \frac{1}{m_i-1}\cdot\left[D_i\left(\langle\underline{p},\underline{\psi}^*\rangle\right)-m_i p_{ir}d_{ir}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)\right].\end{aligned}\tag{37}$$

Lemma 5. Function $\underline{f}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)$ whose components are given by Eq. (37) is Lipschitz continuous over the compact space $\langle \mathrm{K},\mathrm{Y}\rangle$.

Proof. The proof is similar to the proof of Lemma 2. ■

For different values of $\underline{a}$, Eq. (31) generates a different process and we shall use $\underline{p}^{\underline{a}}(k)$ to denote this process whenever the value of $\underline{a}$ is to be specified explicitly. Define a sequence of continuous-time interpolation of (31), denoted by $\tilde{\underline{p}}^{\underline{a}}(t)$ and called an *interpolated process*, whose components are defined by:

$$\underline{\tilde{p}}_i^{\underline{a}}(t)=\underline{p}_i(k),\quad t\in[ka_i,(k+1)a_i),\tag{38}$$

where a_i is the learning parameter of the L_{RP} algorithm for learning automaton LA_i. The interpolated process $\left\{\tilde{\underline{p}}^{\underline{a}}(t)\right\}_{t\geq 0}$is a sequence of random variables that takes values from $\mathrm{R}^{m_1\times\ldots\times m_{\mathrm{N}}}$, where $\mathrm{R}^{m_1\times\ldots\times m_{\mathrm{N}}}$ 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 K, which is a bounded subset of $\mathrm{R}^{m_1\times\ldots\times m_{\mathrm{N}}}$. The objective is to study the limit of the sequence $\left\{\tilde{\underline{p}}^{\underline{a}}(t)\right\}_{t\geq 0}$ as $\max\left\{\underline{a}\right\}\to 0$, which will be a good approximation to the asymptotic behavior of (38). Consider the following ordinary differential equation (ODE):

$$\dot{\underline{p}}=\underline{f}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right),\tag{39}$$

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

$$\frac{dp_{ir}}{dt}=\frac{1}{m_i-1}\cdot\left[D_i\left(\langle\underline{p},\underline{\psi}^*\rangle\right)-m_i p_{ir}d_{ir}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)\right].\tag{40}$$

Theorem 3. Under the learning algorithm (32) and considering $\max\left\{\underline{a}\right\}\to 0$, $\underline{p}(k)$ is well approximated by the solution of the ODE (39).

Proof. Following conditions are satisfied by the learning algorithm given by equation (32):

- $\left\{\underline{p}(k)\right\}_{k\geq 0}$ is a Markovian process.
- Given $\underline{p}(k)$, $\underline{\alpha}(k)$ and $\underline{\beta}(k)$ are independent of $\underline{\alpha}(k-1)$ and $\underline{\beta}(k-1)$.
- $\underline{f}^{\beta}\left(\langle\underline{p}(k),\underline{\psi}^*\rangle\right)$ is independent of k.
- $\underline{f}^{\beta}\left(\langle\underline{p}(k),\underline{\psi}^*\rangle\right)$ is Lipshitz continuous over the compact space K (Lemma 5).
- $E\|\underline{g}\left(\underline{p}\right)-\underline{f}^{\beta}\left(\langle\underline{p},\underline{\psi}^*\rangle\right)\|^2$ is bounded since $\underline{g}\left(\underline{p}\right)\in[-1,1]^{m_1\times\ldots\times m_N}$.
- Learning parameters a_i, $i=1,\ldots,N$ are sufficiently small since $\max\left\{\underline{a}\right\}\to 0$.

Therefore, using theorem (A.1) in [20], we can conclude the theorem. ■

What Theorem 3 says is that $\underline{p}(k)$, given by Eq. (31), will closely follow the solution of the ODE (39), that is $\underline{p}(k)$ can be made to closely approximate the solution of its approximating ODE by taking $\max\left\{\underline{a}\right\}$ sufficiently small. Thus, if the ODE (39) has a globally asymptotically stable equilibrium point, then we can conclude that (by taking $\max\left\{\underline{a}\right\}$sufficiently small), $\underline{p}(k)$, for large k, would be close to this equilibrium point irrespective of its initial configuration $\underline{p}(0)$.

In the following subsections, we first find the equilibrium points of the ODE (39), then study the stability property of these equilibrium points, and finally state a theorem about the convergence of DICLA.

4.2.1. Equilibrium points

To find the equilibrium points of the ODE (39), we first show that this ODE has at least one equilibrium point and then specify a set of conditions which must be satisfied by a configuration $\underline{p}$ to be an equilibrium point of the ODE (39).

Lemma 6. The ODE (39) has at least one equilibrium point.

Proof. To prove this lemma, we first propose a continuous mapping $\varsigma\left(\langle\underline{p},\underline{\psi}^*\rangle\right)$ from $\langle \mathrm{K},\mathrm{Y}\rangle$ to $\langle \mathrm{K},\mathrm{Y}\rangle$. Then, using the Brouwer's fixed point theorem, we will show that any continuous mapping from $\langle \mathrm{K},\mathrm{Y}\rangle$ to $\langle \mathrm{K},\mathrm{Y}\rangle$ has at least one fixed point. Finally, we will show that the fixed point of $\varsigma\left(\langle\underline{p},\underline{\psi}^*\rangle\right)$ is the equilibrium point of the ODE (39). This indicates that the ODE (39) has at least one equilibrium point. The complete proof is given in Appendix B. ■

Theorem 4. The equilibrium points of the ODE (39) are the set of $\underline{p}^*$ which satisfy the set of conditions $p^*ir(t)=\dfrac{\prod_{y\neq r}d_{iy}^{\beta}\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)}{\sum_{y_1}\left(\prod_{y_2\neq y_1}d_{iy_2}^{\beta}\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)\right)}$ for all i, r.

Proof. To find the equilibrium points of the ODE (39), we have to solve equations of the from

$$\frac{dp^*ir(t)}{dt}=0\quad for\ all\ i,r.\tag{41}$$

Using Eqs. (40), (41) can be rewritten as

$$\frac{1}{m_i-1}\cdot\left[D_i\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)-m_i p^*ir(t)d_{ir}^{\beta}\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)\right]=0,\quad for\ all\ i,r.\tag{42}$$

These equations have solutions of the form:

$$p^*ir(t)=\frac{D_i\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)}{m_i d_{ir}^{\beta}\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)},\quad for\ all\ i,r,\tag{43}$$

which after some algebraic manipulations, can be rewritten as:

$$p^*ir(t)=\frac{\prod_{y\neq r}d_{iy}^{\beta}\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)}{\sum_{y_1}\left(\prod_{y_2\neq y_1}d_{iy_2}^{\beta}\left(\langle\underline{p}^*(t),\underline{\psi}^*\rangle\right)\right)},\quad for\ all\ i,r,\tag{44}$$

and hence the theorem. ■

It follows from Theorem 4 that the difference equation given by (35) has equilibrium points $\underline{p}^*$ that satisfy the set of conditions

$$p^*ir(k) = \frac{\prod_{y \neq r} d^{\beta}_{iy}\left(\langle \underline{p}^*(k), \underline{\psi}^* \rangle\right)}{\sum_{y_1}\left(\prod_{y_2 \neq y_1} d^{\beta}_{iy_2}\left(\langle \underline{p}^*(k), \underline{\psi}^* \rangle\right)\right)} \quad \text{for all } i, r.$$

4.2.2. The stability property

In this subsection, we characterize the stability of the equilibrium configurations of DICLA, that is, the equilibrium points of the ODE (39). To do this, the origin is first transferred to an equilibrium point $\underline{p}^*$, and then a Lyapunov-candidate function is introduced for studying the stability of this equilibrium point. Consider the following transformation:

$$\hat{p}_{ir} = p_{ir} - \frac{D_i\left(\langle \underline{p}^*, \underline{\psi}^* \rangle\right)}{m_i d^{\beta}_{ir}\left(\langle \underline{p}^*, \underline{\psi}^* \rangle\right)} \quad \text{for all } i, r. \tag{45}$$

Using this transformation, the origin is transferred to $\underline{p}^*$.

Lemma 7. Derivative of $\underline{\hat{p}}$ with respect to time has components of the following form:

$$\frac{d\hat{p}_{ir}}{dt} = -d^{\beta}_{ir}\left(\langle \underline{p}, \underline{\psi}^* \rangle\right)\hat{p}_{ir} \quad \text{for all } i, r. \tag{46}$$

Proof. The proof is given in Appendix C. ■

Corollary 2. $\underline{\hat{p}}_{ir}$ and its time derivative $\frac{d\hat{p}_{ir}}{dt}$ has different signs.

Proof. Considering the fact that $d^{\beta}_{ir}\left(\langle \underline{p}, \underline{\psi}^* \rangle\right) \geq 0$ for all points $\underline{p}$ and for all i and r, the proof is an immediate result of Eq. (46). ■

Theorem 5. A point $\underline{p}^*$ whose components satisfy the set of equations $p^*ir(k) = \frac{\prod_{y \neq r} d^{\beta}_{iy}\left(\langle \underline{p}^*(k), \underline{\psi}^* \rangle\right)}{\sum_{y_1}\left(\prod_{y_2 \neq y_1} d^{\beta}_{iy_2}\left(\langle \underline{p}^*(k), \underline{\psi}^* \rangle\right)\right)}$ for all i, r, is an asymptotically stable equilibrium point of the ODE (39) over K.

Proof. To prove this theorem, we first apply the transformation (45) to transfer the origin to $\underline{p}^*$. Then we propose a positive definite function $V\left(\underline{p}\right)$ and show that the time derivative of $V\left(\underline{p}\right)$ is globally negative definite over K. This indicates that $\underline{p}^*$ is an asymptotically stable equilibrium point of the ODE (39) over K. The complete proof is given in Appendix D. ■

Corollary 3. The equilibrium point of the ODE (39) is unique over K.

Proof. Let $\underline{p}^*$, $\underline{q}^*$ be two equilibrium points of the ODE (39). Theorem 5 proves that any equilibrium point of the ODE (39), including $\underline{p}^*$, is asymptotically stable over K. This means that all initial configurations within the probability space K converge to $\underline{p}^*$. Using a similar approach for $\underline{q}^*$, one can conclude that all initial configurations within the probability space K converge to $\underline{q}^*$. This implies that $\underline{p}^* = \underline{q}^*$, and thus, the equilibrium point of the ODE (39) is unique over K. ■

4.2.3. Convergence results

In this section, we summarize the main results specified in the above lemmas and theorems in a main theorem (Theorem 6).

Theorem 6. A $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata, regardless of its initial configuration, converges in distribution to a random configuration, in which the mean value of the action probability of any action of any LA is inversely proportional to the average penalty received by that action.

Proof. The evolution of a $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata is described by the Eq. (31). From this equation, it follows that $\left\{\underline{p}(k)\right\}_{k\geq 0}$ is a discrete-time Markov process. Lemma 4 states that this Markovian process is ergodic, and hence, it converges in distribution to a random configuration $\underline{p}^*$, irrespective of its initial configuration. Lemma 6 shows that such a configuration exists for $DICLA^{\underline{\psi}^*}$, Corollary 3 states that it is unique, and Theorem 5 proves that it is asymptotically stable. Theorem 4 specifies the properties of the configuration $\underline{p}^*$. It shows that $\underline{p}^*$ satisfies the set of conditions given by (43). According to Eq. (43), in configuration $\underline{p}^*$, the action probability of any action of any LA is inversely proportional to the average penalty received by that action. ■

4.3. Expediency of DICLA

In this subsection, we introduce the concept of expediency for DICLA and specify the set of conditions under which a DICLA becomes expedient.

Definition 9. A pure-chance automaton is an automaton that chooses each of its actions with equal probability i.e., by pure chance, that is, an m-action automaton is pure-chance if $p_i = \frac{1}{m}$, $i = 1, 2, \ldots, m$.

Definition 10. A DICLA is said to be expedient with respect to the cell c_i if $\lim_{k\to\infty}\langle \underline{p}(k), \underline{\psi}(k)\rangle = \langle \underline{p}^*, \underline{\psi}^*\rangle$ exists and the following inequality holds:

$$\lim_{k\to\infty} E\left[D_i\left(\langle \underline{p}(k), \underline{\psi}(k)\rangle\right)\right] < \frac{1}{m_i}\sum_{y} d^{\beta}_{iy}\left(\langle \underline{p}^*, \underline{\psi}^*\rangle\right). \tag{47}$$

In other words, a DICLA is expedient with respect to the cell c_i if, in the long run, the ith learning automaton performs better (receives less penalty) than a pure-chance automaton.

Definition 11. A DICLA is said to be expedient if it is expedient with respect to every cell in DICLA.

Theorem 7. A $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata, regardless of the local rule being used, is expedient.

Proof. To prove this theorem, we show that a $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata is expedient with respect to every cell in DICLA. The proof is given in Appendix E. ■

5. Numerical examples

To confirm the theoretical analysis given in Section 3, in this section we conduct a number of computer simulations. The first simulation is conducted to confirm the analytical results given in Theorem 2. The next two sets of simulations are used to confirm the analytical results given in Theorem 6. The last simulation is used to study the behavior of DICLA in terms of expediency.

5.1. Numerical example 1

This simulation is conducted to confirm that if a function $H: R^{m_1+\cdots+m_N} \times R^{N\times|\Psi|} \to R$ with the properties specified in Theorem 2 exists and the assumptions (A-1), (A-2), and (A-3) given in Section 4.1.1 hold, then DICLA using the simple restructuring function and with sufficiently small value of the restructuring rate ($\max\left\{\underline{b}\right\} \to 0$), converges, from the structural point of view, to a maximal tendency point. For this simulation, we use a DICLA with three cells and three interests. This DICLA uses the simple restructuring function with the restructuring rate $b = 10^{-4}$. Tendency vector of each cell is specified randomly at the startup of the simulation. We assume that DICLA has a maximal tendency point (specified in Table 3)

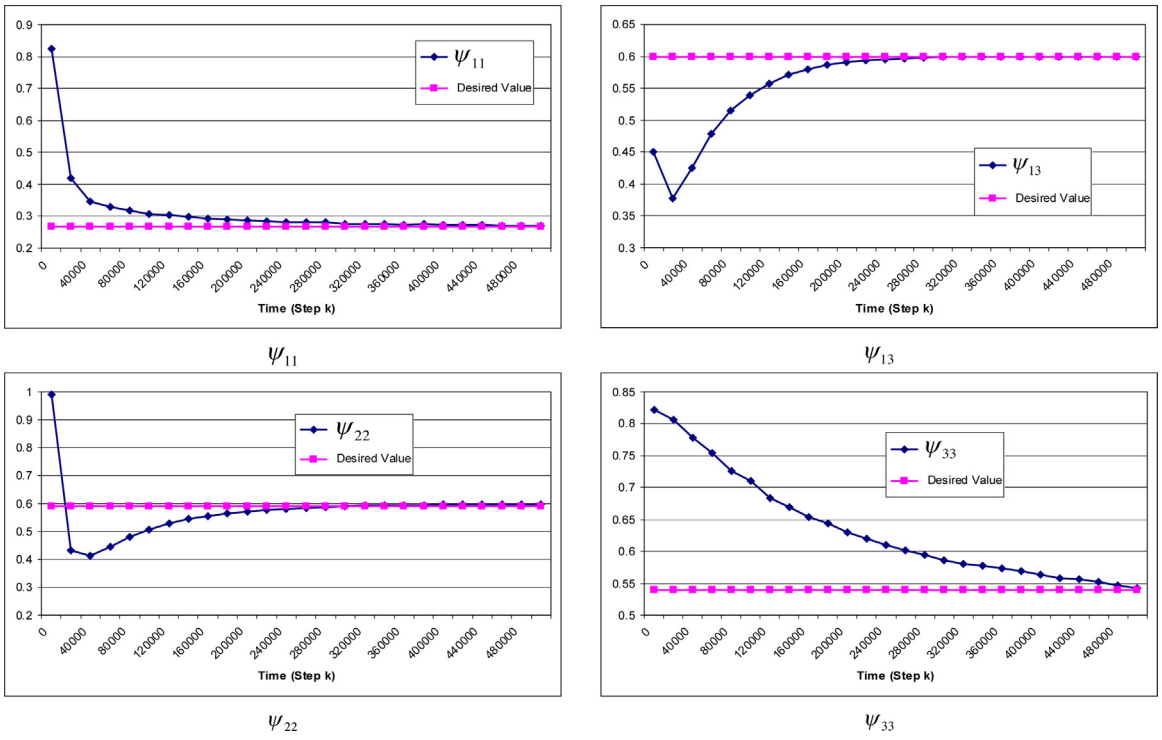


Fig. 5. Comparison of the randomly selected tendencies of every cell of DICLA with their desired values at the maximal tendency point $\underline{\psi}^*$ specified in Table 3.

Table 3
Tendencies of each cell to each of the interests of DICLA at the maximal tendency point $\underline{\psi}^*$.

	Ψ_1	Ψ_2	Ψ_3
c_1	$\psi^*_{11} = .26$	$\psi^*_{12} = .43$	$\psi^*_{13} = .6$
c_2	$\psi^*_{21} = .91$	$\psi^*_{22} = .59$	$\psi^*_{23} = .09$
c_3	$\psi^*_{31} = .44$	$\psi^*_{32} = .73$	$\psi^*_{33} = .54$

Table 4
Tendencies of each cell to each of the interests of DICLA at the end of the simulation time ($k > 3 \times 10^6$) of the numerical example 1.

	Ψ_1	Ψ_2	Ψ_3
c_1	$\psi_{11} = 0.27$	$\psi_{12} = 0.41$	$\psi_{13} = 0.58$
c_2	$\psi_{21} = 0.88$	$\psi_{22} = 0.6$	$\psi_{23} = 0.07$
c_3	$\psi_{31} = 0.44$	$\psi_{32} = 0.72$	$\psi_{33} = 0.53$

(Assumption (A-1)) and the restructuring signal generator ($\mathcal{F}_\zeta$) of every cell can be described as a multiplication of $\mathcal{F}^i$ and $\mathcal{G}^i$ (Assumptions (A-2) and (A-3)). At each step k, $\overline{\mathcal{G}}^{ij}\left(\langle \underline{p}(k), \underline{\psi}(k)\rangle\right)$ is selected randomly from the range $\left[-\psi_{ij}(k), 1-\psi_{ij}(k)\right]$. Table 4 gives the results of this numerical study. By comparing the results given in this table and the desired values of tendencies given in Table 3, it can be concluded that the tendency of each cell to each of the interests approaches its desired value. This confirms the theoretical analysis given in Theorem 2. Fig. 5 also shows the approach of the tendency to its desired value over the simulation time for randomly selected tendencies of every cell of DICLA. As it can be seen from this figure, the approach of tendencies towards their desired values is regardless of their initial values, as it was stated in the theorem. Another observation in this figure is that tendency values have very few deviations and seamlessly approach towards $\underline{\psi}^*$. The reason behind this observation is that the number of iterations, depicted in the figure, is very high (about 5×10^5) and thus, small deviations between subsequent iterations cannot be seen in the figure.

5.2. Numerical example 2

This set of simulations are conducted to confirm that the action probability of any action of any learning automaton in a $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata converges in distribution to a random variable, whose mean is inversely proportional to the average penalty received by that action. We consider 6 different DICLAs with different number of cells and different number of actions for the learning automata. Table 5 gives the specifications used for this set of simulations. In this table, $DICLA_{(N,\ m)}$ refers to a $DICLA^{\underline{\psi}^*}$ with N cells, each has a learning automaton with m actions. Table 6 compares the action probabilities of the actions of the learning automata in each $DICLA^{\underline{\psi}^*}$ at the end of the simulation time ($k > 3 \times 10^6$) with their theoretical values obtained from the equation (43). As it can be seen from these tables, the action probability of all actions and their theoretical values approach each other. This is in coincidence with the results of the theoretical analysis given in Theorem 6, that is, a $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata converges in distribution to a random configuration, in which the mean value of the action probability of any action of any LA is inversely proportional to the average penalty received by that action. Fig. 6 also shows the approach of these two values to each other over the simulation time for randomly selected actions of three randomly selected learning automata from $DICLA_{2,3}$, $DICLA_{5,5}$. Note the difference between the time scale of Fig. 6, about 3×10^6 iterations, with that of Fig. 5, about 5×10^5 iterations. This difference is due to the fact at first, structural dynamicity of the model enters its steady state (after about 5×10^5 iterations), and then, its behavioral dynamicity starts to diminish.

5.3. Numerical example 3

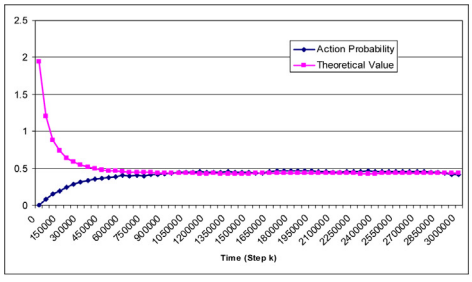
The goal of conducting this set of simulations is to study the behavioral dynamics of $DICLA^{\underline{\psi}^*}$ when it starts to evolve within the environment from different initial configurations. For this study, we

Table 5
Specifications used for numerical example 2.

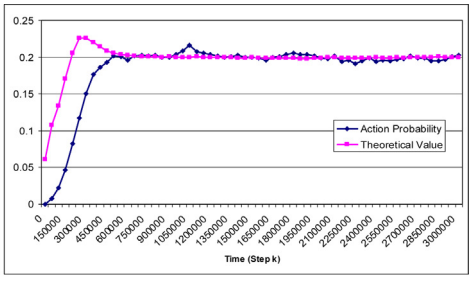
$DICLA^{\psi^*}_{\frac{}{}}$	$DICLA_{2,3}$	$DICLA_{3,2}$	$DICLA_{5,5}$
Neighborhood	All cells are neighbors		
Learning Parameter	$a = 5 \times 10^{-5}$		
Initial Configuration	$p_{i1} = 1$, $p_{ir} = 0$, $\forall i,r,\ r \neq 1$		
Environment Response	$\mathcal{F}^1(\alpha_1, \ldots, \alpha_N)$ is selected uniformly at random from the range [0,1] at the beginning of the simulation		

Table 6
Results of the numerical example 2 for $DICLA_{2,3}$, $DICLA_{3,2}$, $DICLA_{5,5}$ (p_{ij} is the action probability of the jth action of the ith learning automaton obtained from the simulation study and p^*_{ij} is the corresponding theoretical value).

$DICLA_{2,3}$	LA_1	α_1	p_{11}=0.352	LA_2	α_1	p_{21}=0.289										
			p^*_{11} = 0.361			p^*_{21} = 0.288										
		α_2	p_{12} = 0.454		α_2	p_{22} = 0.268										
			p^*_{12} = 0.441			p^*_{22} = 0.268										
		α_3	p_{13} = 0.194		α_3	p_{23} = 0.443										
			p^*_{13} = 0.198			p^*_{23} = 0.444										
$DICLA_{3,2}$	LA_1	α_1	p_{11} = 0.746	LA_2	α_1	p_{21} = 0.797	LA_3	α_1	p_{31} = 0.663							
			p^*_{11} = 0.75			p^*_{21} = 0.8			p^*_{31} = 0.659							
		α_2	p_{12} = 0.254		α_2	p_{22} = 0.203		α_2	p_{32} = 0.337							
			p^*_{12} = 0.25			p^*_{22} = 0.2			p^*_{32} = 0.341							
$DICLA_{5,5}$	LA_1	α_1	p_{11} = 0.21	LA_2	α_1	p_{21} = 0.208	LA_3	α_1	p_{31} = 0.189	LA_4	α_1	p_{41} = 0.176	LA_5	α_1	p_{51} = 0.214	
			p^*_{11} = 0.211			p^*_{21} = 0.211			p^*_{31} = 0.192			p^*_{41} = 0.173			p^*_{51} = 0.212	
		α_2	p_{12} = 0.212		α_2	p_{22} = 0.215		α_2	p_{32} = 0.222		α_2	p_{42} = 0.208		α_2	p_{52} = 0.183	
			p^*_{12} = 0.213			p^*_{22} = 0.212			p^*_{32} = .215			p^*_{42} = 0.207			p^*_{52} = 0.188	
		α_3	p_{13} = 0.2		α_3	p_{23} = 0.201		α_3	p_{33} = 0.174		α_3	p_{43} = 0.189		α_3	p_{53} = 0.212	
			p^*_{13} = 0.199			p^*_{23} = 0.2			p^*_{33} = 0.177			p^*_{43} = 0.192			p^*_{53} = 0.214	
		α_4	p_{14} = 0.213		α_4	p_{24} = 0.178		α_4	p_{34} = 0.182		α_4	p_{44} = .197		α_4	p_{54} = 0.174	
			p^*_{14} = 0.212			p^*_{24} = 0.179			p^*_{34} = 0.181			p^*_{44} = 0.196			p^*_{54} = 0.175	
		α_5	p_{15} = 0.165		α_5	p_{25} = 0.198		α_5	p_{35} = 0.233		α_5	p_{45} = 0.23		α_5	p_{55} = 0.217	
			p^*_{15} = 0.165			p^*_{25} = 0.198			p^*_{35} = 0.235			p^*_{45} = 0.232			p^*_{55} = 0.211	



***DICLA*$_{2,3}$: *LA*$_1$: Action 2**

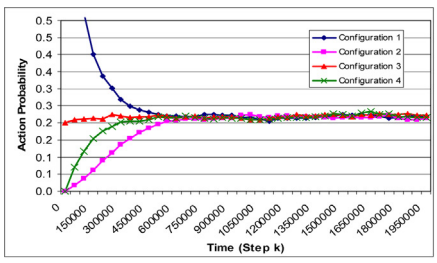


***DICLA*$_{5,5}$: *LA*$_4$: Action 4**

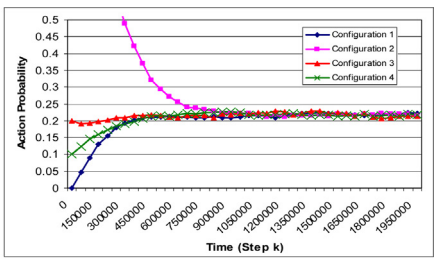
Fig. 6. Results of the numerical example 2 for randomly selected actions of three randomly selected learning automata from $DICLA_{2,3}$, $DICLA_{5,5}$.

Table 7
Initial configurations for $DICLA_{5,5}$ used in numerical example 3.

Configuration 1					Configuration 2					Configuration 3					Configuration 4				
LA_1	LA_2	LA_3	LA_4	LA_5	LA_1	LA_2	LA_3	LA_4	LA_5	LA_1	LA_2	LA_3	LA_4	LA_5	LA_1	LA_2	LA_3	LA_4	LA_5
0	0	0	0	0	1	0	0	0	0	0.2	0.4	0.3	0.5	0.6	0.1	0.8	0.9	0.2	0.4
0	0	0	0	0	0	1	0	0	0	0.2	0	0.1	0.1	0.3	0.2	0.05	0.05	0.6	0.1
0	0	0	0	0	0	0	1	0	0	0.2	0.4	0.3	0.1	0	0.3	0	0	0.1	0.3
0	0	0	0	0	0	0	0	1	0	0.2	0	0.1	0.2	0	0.4	0.05	0	0.1	0
1	1	1	1	1	0	0	0	0	1	0.2	0.2	0.2	0.1	0.1	0	0.1	0.05	0	0.2



***LA*$_1$: Action 5**



***LA*$_4$: Action 4**

Fig. 7. Results of the numerical example 3 for randomly selected actions of two randomly selected learning automata.

Table 8
Results of the numerical example 4.

		$E\left[D_i\left(\langle \underline{p}(k), \underline{\psi}^*\rangle\right)\right]$	$\frac{1}{m_i}\sum_y d_{iy}^{\beta}\left(\langle \underline{p}(k), \underline{\psi}^*\rangle\right)$			$E\left[D_i\left(\langle \underline{p}(k), \underline{\psi}^*\rangle\right)\right]$	$\frac{1}{m_i}\sum_y d_{iy}^{\beta}\left(\langle \underline{p}(k), \underline{\psi}^*\rangle\right)$
$DICLA_{2,3}$	LA_1	3.61×10^{-2}	5.05×10^{-2}	$DICLA_{5,5}$	LA_1	1.368×10^{-1}	1.372×10^{-1}
	LA_2	3.61×10^{-2}	4.48×10^{-2}		LA_2	1.368×10^{-1}	1.371×10^{-1}
$DICLA_{3,2}$	LA_1	1.02×10^{-1}	1.04×10^{-1}		LA_3	1.368×10^{-1}	1.374×10^{-1}
	LA_2	1.016×10^{-1}	1.018×10^{-1}		LA_4	1.368×10^{-1}	1.373×10^{-1}
	LA_3	1.02×10^{-1}	1.03×10^{-1}		LA_5	1.368×10^{-1}	1.374×10^{-1}

use a $DICLA_{5,5}$ with SL_{RP} learning automata. Table 7 gives the initial configurations of $DICLA_{5,5}$ used in this study. In this table, the initial action probability vector for each learning automaton in each configuration is given beneath that learning automaton. For example, in configuration 3, the initial action probability vector of the learning automaton LA_4 is $[0.5, 0.1, 0.1, 0.2, 0.1]^T$. Fig. 7 plots evolution of the action probabilities of two sample actions of two of the LA in the DICLA for different initial configurations. As it can be seen from this figure, no matter what the initial configuration of DICLA is, DICLA converges to its equilibrium configuration. Thus, the results of this set of examples coincide with the results given in Theorem 6 in Section 3, that is, the convergence of DICLA to its equilibrium configuration is independent of its initial configuration.

5.4. Numerical example 4

This set of simulations is conducted to study the behavior of $DICLA^{\underline{\psi}^*}$ in terms of the expediency. For this study, we use the specifications of the second numerical example, given in Table 5. Table 8 compares $E\left[D_i\left(\langle \underline{p}(k), \underline{\psi}^*\rangle\right)\right]$ and $\frac{1}{m_i}\sum_y d_{iy}^{\beta}\left(\langle \underline{p}(k), \underline{\psi}^*\rangle\right)$ at the end of the simulation time ($k > 2.85\times10^6$) for all learning automata of all DICLAs given in Table 5. As it can be seen from this table, the average penalty received by any learning automaton LA_i is less than that of a pure chance automaton. This is in coincidence with the theoretical results given in Theorem 7, that is, a $DICLA^{\underline{\psi}^*}$ with SL_{RP} learning automata, regardless of the local rule being used, is expedient.

6. Conclusion, discussion, and future work

In this paper, we proposed dynamic irregular cellular learning automaton (*DICLA*) as an extension to the ICLA model. In contrast to the ICLA model, DICLA has a dynamic structure which is more suitable for modeling problems in certain domains such as mobile ad hoc and sensor networks. We introduced the concept of expediency for the proposed learning model and analytically identify sufficient conditions under which a DICLA becomes expedient. We also gave some numerical examples for illustrating the analytical results.

There are some limitations for the proposed DICLA model. One is the speed of convergence of the model. Numerical analysis indicated that the model enters its steady state after millions of iterations. This is far too slow for practical applications. There exists solutions to this limitation like the multiple model given in [22]. As a possible future work, one may consider applying this model in the DICLA.

Another limitation of the proposed model is the complexity of the analysis of its behavior when the structure as well as action probability vectors change together. In this paper, we consider restructuring signals which eventually reach stable structure, from that point on, it is easier to study the behavior of learning automata. More complex restructuring signals, such as random, chaotic, repetitive, pattern-based, etc., can be considered and analyzed as another possible future work to this study.

The local environment of learning automata in the DICLA, as defined in this paper, are stationary; that is, for a fixed combination of selected actions by neighboring learning automata, the reinforcement signal would be a random variable with a predefined expected value. But in many real world applications, environments are non-stationary and thus, it is required to study the model under such environments. This could be another suggestion for interested researchers to continue this research.

Finally, in this study, we have carried out the steady-state behavior of the DICLA only for SL_{RP} learning algorithm. SL_{RP} learning algorithm results in an expedient model which performs only better than an equivalent pure-chance model. The study could be repeated with $SL_{R\varepsilon P}$ learning algorithm which is expected to result in ε-optimal models.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.jocs.2017.08.012.

References

[1] M. R.Meybodi, H. Beigy, M. Taherkhani, Cellular learning automata and its applications, Sharif J. Sci. Technol. 19 (25) (2003) 54–77.
[2] N.H. Pakard, S. Wolfram, Two-dimensional cellular automata, J. State Phys. 6 (1985) 901–946.
[3] K.S. Narendra, M.A.L. Thathachar, Learning Automata: An Introduction, Prentice-Hall Inc, 1989.
[4] H. Beigy, M.R. Meybodi, A mathematical framework for cellular learning automata, J. Adv. Complex Syst. 7 (3) (2004) 295–320.
[5] H. Beigy, M.R. Meybodi, Asynchronous cellular learning automata, Automatica 44 (5) (2008) 1350–1357.
[6] H. Beigy, M.R. Meybodi, Open synchronous cellular learning automata, Adv. Complex Syst. 10 (4) (2007) 527–556.
[7] H. Beigy, M.R. Meybodi, Cellular learning automata with multiple learning automata in each cell and its applications, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 40 (1) (2010) 54–66.
[8] M. Esnaashari, M.R. Meybodi, Irregular cellular learning automata, IEEE Trans. Cybern. (2014), http://dx.doi.org/10.1109/tcyb.2014.2356591.
[9] M. Esnaashari, M.R. Meybodi, A cellular learning automata based clustering algorithm for wireless sensor networks, Sens. Lett. 6 (5) (2008) 723–735.
[10] M. Esnaashari, M.R. Meybodi, Dynamic point coverage problem in wireless sensor networks: a cellular learning automata approach, J. Ad Hoc Sens. Wireless Netw. 3 (2010) 193–234.
[11] A. Enami Eraghi, J. Akbari Torkestani, M.R. Meybodi, Cellular learning automata-based graph coloring problem, in: Proceedings of 2009 International Conference on Machine Learning and Computing, Perth, Australia, 2009, pp. 163–167.
[12] M. Vahidipour, M.R. Meybodi, M. Esnaashari, Adaptive petri net based on irregular cellular learning automata and its application in vertex coloring problem systems with unknown parameters, Appl. Intell. 46 (2) (2017) 272–284.
[13] M. Vahidipour, M.R. Meybodi, M. Esnaashari, Finding the shortest path in stochastic graphs using learning automata and adaptive stochastic petri nets, Int. J. Uncertainty Fuzziness Knowledge Based Syst. 25 (3) (2016) 427–455.
[14] A. Enami Eraghi, J. Akbari Torkestani, M.R. Meybodi, A.H. Fathy Navid, Cellular learning automata-based channel assignment algorithms in mobile ad hoc network, in: Proceedings of 2009 International Conference on Machine Learning and Computing, Perth, Australia, 2009, pp. 173–177.
[15] M. Esnaashari, M.R. Meybodi, A cellular learning automata-based deployment strategy for mobile wireless sensor networks, J. Parallel Distrib. Comput. (2011) 988–1001.
[16] M.L. Tsetlin, On the behavior of finite automata in random media, Autom. Remote Control 22 (1962) 1210–1219.

[17] A.M. Saghiri, M.R. Meybodi, A closed asynchronous dynamic model of cellular learning automata and its application to peer-to-peer networks, Genet. Progr. Evolv. Mach. (2017), http://dx.doi.org/10.1007/s10710-017-9299-7.
[18] A.M. Saghiri, M.R. Meybodi, An adaptive super-peer selection algorithm considering peers capacity utilizing asynchronous dynamic cellular learning automata, Appl. Intell. (2017), 10.117/s10489-017-0946-9.
[19] D.L. Isaacson, R.W. Madsen, Markov Chains: Theory and Applications, John Wiley & Sons, New York, 1976.
[20] M.A.L. Thathachar, P.S. Sastry, Networks of Learning Automata, Kluwer Academic Publishers, USA, 2004.
[21] M.F. Norman, Some convergence theorems for stochastic learning models with distance diminishing operators, J. Math. Psychol. 5 (1968) 61–101.
[22] K.S. Narendra, Y. wang, S. Mukhopadhay, Fast reinforcement learning using multiple models, in: 55th IEEE International Conference on Decision and Control, Las Vegas, NV, USA, 2016.

Mehdi Esnaashari received the B.S., M.S., and Ph.D. degrees in Computer Engineering all from Amirkabir University of Technology, Tehran, Iran, in 2002, 2005, and 2011 respectively. He worked at Iran Telecommunications Research Center as an Assistant Professor from 2012 to 2016. Currently, he is an Assistant Professor in Computer Faculty of K. N. Toosi University of Technology. His research interests include computer networks, learning systems, soft computing, and information retrieval.

M.R. Meybodi received the B.S. and M.S. degrees in Economics from the Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M.S. and Ph.D. degrees from the Oklahoma University, USA, in 1980 and 1983, respectively, in Computer Science. Currently he is a Full Professor in Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. Prior to current position, he worked from 1983 to 1985 as an Assistant Professor at the Western Michigan University, and from1985 to 1991 as an Associate Professor at the Ohio University, USA. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.