Physica A 515 (2019) 600–616

Contents lists available at ScienceDirect

Physica A

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

A study on delay-sensitive cellular automata

Souvik Roy

Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, 711103, India

HIGHLIGHTS

- Cellular automata with delay and probabilistic loss of information is introduced.
- The effect of proposed CA in the dynamics of ECA and Game of Life is presented.
- Phase transition for both ECA and Game of Life is found.

ARTICLE INFO

Article history:
Received 8 February 2018
Received in revised form 27 September 2018
Available online 9 October 2018

Keywords:
Cellular automata(CAs)
Elementary cellular automata(ECA)
Game of Life
Delay
Probabilistic loss of information
Phase transition

ABSTRACT

Classically, in cellular automata, no delay in information sharing among the neighbouring cells is considered. However, this assumption is not generally true for natural complex systems (such as physical, biological, social systems) and distributed systems where information sharing (non-uniform) delay cannot generally be ignored. Moreover, sometimes in complex and distributed systems, messages are lost, which makes the system non-deterministic. In this context, the effect of (non-uniform) delay and probabilistic loss of information during information sharing between neighbours in the dynamics of elementary cellular automata and Game of Life is presented in this study. We study the wide variety of results, which include the study of *phase transitions*, for both the elementary cellular automata and Game of Life using a statistical experimental approach.

1. Introduction

A cellular automaton (CA) is a class of dynamic systems for modelling systems with local interactions. A CA is defined over a regular grid, each cell of which consists of a finite automaton that interacts with its neighbours to go to its next state [1]. The beauty of CA comes from simple local interaction which produces a (global) complex behaviour. Since their inception, cellular automata (CAs) have been utilized as an effective tool to model many natural complex (such as physical, biological, social, etc.) systems, distributed systems and networked systems, etc. [2,3]. CAs have always been a natural choice for modelling complex and distributed systems because of their built-in parallelism. Classically, in any cellular automaton, if a cell wants to get state information about its neighbours, the cell gets the information instantly in the next time step. That is, no delay in information sharing among the neighbouring cells is considered. In real natural complex, distributed and networked systems, on the other hand, information sharing delay cannot generally be ignored [4–6]. Most importantly, the delay of information sharing with neighbour cannot be uniform in any natural complex and distributed system.

In this context, this paper develops a different type of cellular automata where a cell shares its state with its neighbours with some delay. The delays of different neighbouring cells may be different. Information sharing in the proposed system is non-deterministic in nature, which implies that a cell shares its state information with its neighbours probabilistically.

E-mail address: svkr89@gmail.com.

https://doi.org/10.1016/j.physa.2018.09.195
0378-4371/

1.1. Notes on the history of delay (in CA)

In the literature of cellular automata, few researchers have explored a new type of asynchronism where a cell shares its state information with its neighbours probabilistically [7,8]. If a cell does not share its state with a neighbour, then it implies that the delay of information passing is infinite. Therefore, the works of Ref. [7,8] have dealt in some sense with delay during information passing. However, if a cell agrees to share its state with any one of its neighbours, it can do it within one time step only. This implies that either there is no delay during state sharing, or there is uniform delay, which is less than the duration of one time step. This fact can be seen in the models of Ref [7,8].

Different kinds of cellular automata models with time delay (for modelling purpose, i.e. cancer tumours, SEIR (susceptible–exposed–infected–recovered) epidemic, etc.) also have been proposed by CA researchers [5,6]. Iarosz et al. defined a cellular automata model for cancer tumours which included the time delay between the time in which the site is affected and the time in which its variable is updated, see Ref. [5]. This work have analysed the effect of time delay on the dynamics through cluster counting. However, the work have dealt with non-uniform time delay in general, but the time delay for all the cells is constant (i.e. uniform) for a particular time step. Recently, Sharma and Gupta [6] have presented a two-dimensional cellular automata model to simulate *SEIR* epidemic spread with time delay. The work of [6] have shown that the delay plays a central role to control disease progression in an infected host. In this context also note that Alonso-Sanz [9,10] have investigated the effect of memory of delay type in the dynamics of elementary cellular automata. Here, the 'memory of delay type' indicates that every cell retains historic memory of its past states. However, this study develops a different type of cellular automata where a cell shares its state with its neighbours with some delay. The works of [11–13] ([11] for one-dimensional case and [12,13] for Game of Life) have also alternatively dealt with 'probabilistic loss of information' in terms of *coupling parameter* (κ) which locally smooths the rigid specification of deterministic cellular automata rules and makes them fuzzier.

1.2. Overview of the proposed model

In this scenario, we target to study a cellular automaton system where delay is not uniform. Let us consider, a cell is updated at time t and cell has two neighbours (left and right). In traditional CA, the two neighbours both can see the updated state information in next time step. In the proposed model, we question this assumption. We consider here that at time $t+1$, both the neighbours may not be aware of the cell's update; only one (or none) can see the update at time $t+1$, and other can observe the update at time $t' > t+1$. This consideration makes the proposed CA more suitable for modelling natural complex and distributed systems.

Now, we develop our proposed CA system which involves delay during information sharing between two neighbours. It only concentrates on the delay when a cell shares its state information with its neighbours. In the proposed system we introduce a non-negative integer function $D(i,j)$ for each i and j where $i \neq j$ and i and j are two neighbouring cells of CA. $D(i,j)$ defines the delay involved in sharing of information of cell i with cell j. If $D(i,j) = 0$, then any update of cell i is immediately seen by cell j. If $D(i,j) = 1$, then the update of cell i is seen by cell j within 1 time step. This implies that if cell i updates its state at time t, then that updated state information is available to cell j at time $t+1$. In traditional cellular automata where delay is not explicitly mentioned, either $D(i,j) = 0$ or $D(i,j) = 1$ for each pair of i and j where cell i and cell j are neighbours to each other. In the proposed system, $D(i,j)$ can be any non-negative integer value. However, following property is obeyed by D in our system.

- $D(i,j) = D(j,i)$ for any pair of neighbouring cells i and j.

Apart from this, the following property is also desirable for our system.

- If i and j are neighbours, then $D(i,j) \geqslant 1$.

The above properties of D (that is, delay) are also suitable for any natural complex and distributed systems. Here, the delays are non-uniform in space, i.e. $D(i,i')$ may be different from $D(j,j')$, where i and i'; and j and j' are neighbouring cells. Note that the delays are uniform in time. However, the delays are deterministic here. Sometimes in natural complex and distributed systems messages are lost, which makes the system non-deterministic. So apart from 'delay', we introduce a probabilistic loss of information during state information sharing in our model.

The 'delay', in our system is not part of the cell's local rule, and the cells are governed by the local rule. When a cell acts to update its state, the cell has to know the neighbours' states. In our system, each cell has a *view* about the states of its neighbours. The view may change from time to time depending on the arrival of state information about neighbours. However, the cells act depending on the current state information about neighbours at that time. Therefore, each cell has to maintain the state information of neighbours to get a view of neighbours' states. When we simulate this system, we use some memory for each cell to store the state information of neighbours. The memory is updated if a new state information is arrived.

As an example, let us consider, a simple 3-cell, 3-neighbour cellular automaton of Fig. 1(a). State of the cells are given within the circles. Consider $D(0,1) = D(1,2) = 1$ and $D(0,2) = 2$. Let us consider that the cells act at time t. Now, each cell has to know the states of its neighbours. Fig. 1(a) shows that each cell has a *view* about the states of neighbours. Cell 0 *sees* that state of cell 1 is 1 (noted over the right side link) and state of cell 2 is 0 (noted over the left link). Now, assume

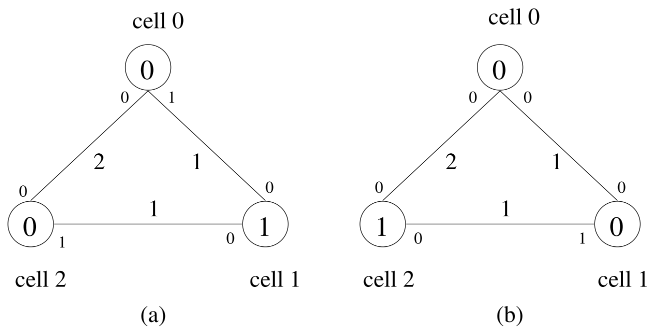


Fig. 1. (a) 3-cell cellular automaton with delay of each link at time t, (b) the system at time $t+1$.

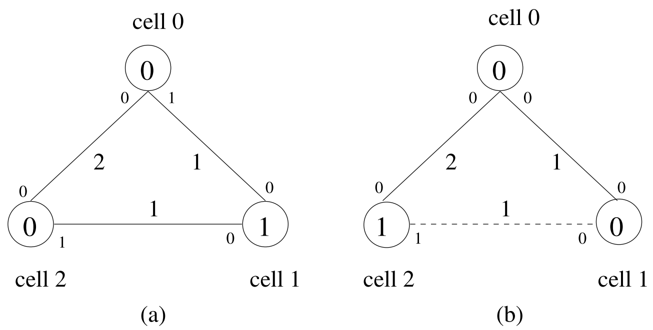


Fig. 2. (a) 3-cell cellular automaton with delay of each link at time t, (b) the system at time $t+1$, here, the probabilistic loss of information is represented by a dotted link.

that cells 1 and 2 change their states. Fig. 1(b) shows the system at time $t+1$. The information about state change of cell 2 does not reach to cell 0 at time $t+1$. So, to cell 0, state of cell 2 is 0, though it is 1. Now, Fig. 2 shows the same example cellular automaton (of Fig. 1) to introduce the probabilistic loss of information during state information sharing. In Fig. 2(b), the information about state change of cell 1 does not reach to cell 2 at $t+1$ time step due to probabilistic loss of information. Therefore, to cell 2, state of cell 1 is 1, though it is 0.

Therefore, the proposed cellular automata involve delay and probabilistic loss of information during information sharing with neighbours. To sum up, the paper develops a special kind of cellular automata which involve following three type of updating schemes.

(i) Probabilistic loss of information perturbation;
(ii) Delay perturbation; and
(iii) Delay and probabilistic loss of information perturbation.

Note that 'probabilistic loss of information perturbation' and 'delay perturbation' are the special cases of the 'delay and probabilistic loss of information perturbation'. Therefore, following three research questions guided this study: (1) Are cellular automata sensitive to the effect of probabilistic loss of information perturbation? (2) Do cellular automata show any peculiar behaviour with the effect of delay perturbation? (3) What will be the effect of delay and probabilistic loss of information perturbation together?

To explore these questions, here, we have concentrated on two simple cellular automata models: elementary cellular automata (ECA) and Game of Life. In this scenario, we give a formal description of the proposed CA system in the next section.

2. The proposed CA system

2.1. Cellular automata

Classically, cellular automata (CAs) are dynamical systems that are homogeneous and discrete in both time and space. CA can be defined by 4-uplet (L, S, N, f), where

- $L \subset \mathbb{Z}^d$ is the d-dimensional cellular space; each element of L is called a *cell*.
- Set S is finite, where the elements are called states that each cell assumes at any time t.

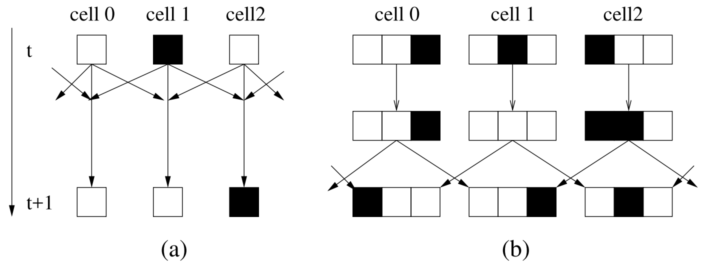


Fig. 3. Example of the (a) classical CA framework; and (b) proposed extended CA framework for a 3-cell simple case.

- $N \subset L$ is a finite set of vectors called the neighbourhood, which associates to a cell the set of its neighbouring cells. N and L are such that $\forall c \in L$ and $\forall n \in N$, the neighbour $c + n$ is in L.[1] Generally, the neighbours of a cell are the nearest cells surrounding the cell.
- f is called the local transition rule of the automaton, which defines the next state of a cell depending on the state of the cell and the state of its neighbouring cells.

The mapping $x^t : L \rightarrow S$ implies the configuration of the automation indicating the state of every cell at time t. Here, S^L represents the set of all configurations. The global transition function can be written as $x^{t+1} = G(x^t)$, so that, $\forall c \in L, x_c^{t+1} = f(x_c^t, x_{c+n_1}^t, \ldots, x_{c+n_k}^t)$, where $N = \{n_1, \ldots, n_k\}$. Traditionally, all the cells of a CA are forced to get updated simultaneously.

2.2. The proposed CA system

Now, to formally describe the proposed CA system which involves delay and probabilistic loss of information during information sharing between two neighbours, we need to extend the classical framework of CA.

- In the proposed CA system, each cell has a *view* about the states of its neighbours. The *view* may change from time to time depending on the arrival of state information about neighbours. In this context, the state set is distinguished by two parts—the *actualstate* of a cell; and a vector of k neighbours *viewstate*.[2] Now, the extended state set can be written as $S' = S \times S^k$. Therefore, a cell configuration x_c^t is distinguished by two parts:

  (i) $a_c^t \in S$ is the *actualstate*; and
  (ii) $\mathbf{v}_c^t \in S^k$ is the vector of k *viewstate* of neighbours. Note that the vector of viewstate associates an element of S to each element of the neighbourhood.

- In the proposed CA system, $D(i, j) \geq 1$, for each pair of neighbouring cells i and j where $i \neq j$. Here, the local transition function is also sub-divided into two parts:

  (i) The first *state update* step, a cell changes its *actualstate* depending on the *actualstate* of the self and *viewstates* of neighbours. Here, the viewstate of neighbours remain unchanged; and
  (ii) The second *information sharing* step, the cell shares its updated *actualstate* to its neighbouring cells. Here, the cell shares its updated actualstate to its neighbouring cells with probabilistic loss of information and delay perturbation during 'information sharing' step.

  Now, the extended local transition function can be written as $f' = f_u \circ f_s$, where, f_u is the *state update* function, and f_s is the *information-sharing* function. Here, the operator '$\circ$' indicates that the functions are applied sequentially to represent the actual update.

As an example, let us consider, a simple 3-cell, 3-neighbour cellular automata of Fig. 3. State 0 is represented in white, and state 1 in black. Here, Fig. 3(a) shows the classical framework of CA, whereas, Fig. 3(b) shows the proposed extended framework of CA. In Fig. 3(b), the *actualstate* of each cell (middle one) is surrounded by the *viewstates* (left and right one). Here, the first step shows the *state update* function and the second step shows the *information-sharing* function.

Recall that the cell shares its updated actualstate to its neighbouring cells with probabilistic loss of information and delay perturbation during 'information sharing' step. Therefore, the 'information sharing' step depends on the following two parameters:

---

1 Note that here, we intentionally assume that the neighbourhood does not contain the cell itself. This assumption is necessary to explicitly represent the information sharing between a cell and its neighbours.

2 Note that the 'actualstate' corresponding to the 'eigenstate' in Ref. [7]. The 'observation state' of Ref. [7] has also alternatively named as 'viewstate'.

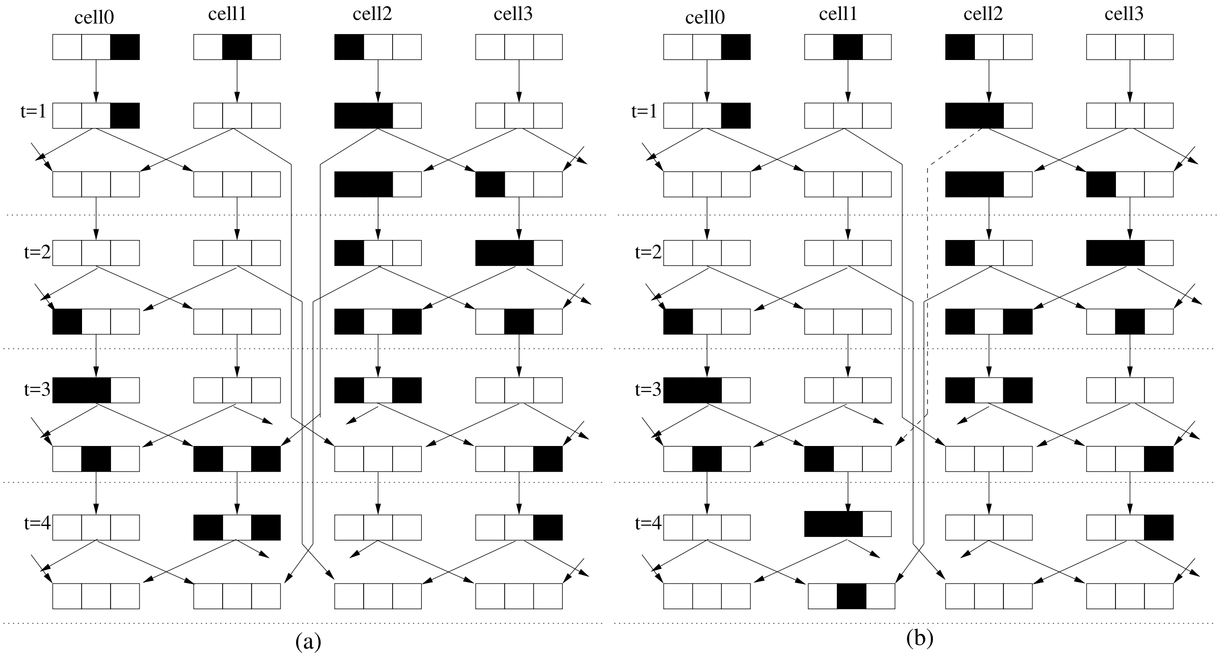


Fig. 4. Example of the (a) delay perturbation; and (b) delay and probabilistic loss of information perturbation updating scheme. The applied rule is ECA 152.

1. The probabilistic loss of information perturbation rate ι (iota) indicates the probabilistic loss of information during 'information sharing' step. Here, the actualstate of a cell is updated at each time step, but the atomic information sharing of the cell's actualstate to the neighbour's corresponding viewstate is realized with a probability ι, where $0 \leqslant \iota \leqslant 1$.
2. The delay perturbation parameter $D \in \mathbb{N}$ indicates the maximum delay limit during information sharing. Every pair of neighbouring cells are randomly initialized with delay between 1 to D following a *uniform*[3] distribution. In this work, $1 \leq D \leq 4$. Here, actualstate of a cell is updated at each time step, but the atomic information sharing of the cells actualstate to the neighbour's corresponding viewstate is realized after corresponding 'delay parameter value' time steps.

Therefore, the paper develops a special kind of cellular automata which involve following three type of updating schemes:

Probabilistic loss of information perturbation: The actualstate of a cell is updated at each time step, but the atomic information sharing of the cell's actualstate to the neighbour's corresponding viewstate is realized with a probabilistic loss of information perturbation rate ι (iota).

Delay perturbation: The actualstate of a cell is updated at each time step, but the atomic information sharing of the cells actualstate to the neighbour's corresponding viewstate is realized after corresponding 'delay parameter value' time steps.

Delay and probabilistic loss of information perturbation: The actualstate of a cell is updated at each time step, but the atomic information sharing of the cell's actualstate to the neighbour's corresponding viewstate is realized with the effect of both the delay and probabilistic loss of information perturbation.

As an example, Fig. 4 shows a simple 4-cell cellular automaton with initial configuration '0100'. Here, the applied rule is ECA 152 (see Appendix A). Note that a configuration of ECA 152 can be viewed as regular right shift branches of state '1' cell which is surrounded by state '0' cell. State 0 is represented in white, and state 1 in black. In Fig. 4, the actualstate of each cell (middle one) is surrounded by the viewstates (left and right one). For every time steps, the first step shows the state update function and the second step shows the information sharing function. Here, the probabilistic loss of information is represented by a dotted link and D(0,1) = D(2,3) = D(3,0) = 1 and D(1,2) = 3. Fig. 4(a) shows example for delay perturbation updating scheme and Fig. 4(b) shows example for delay and probabilistic loss of information perturbation updating scheme. In Fig. 4(a), the information about state change of cell 1 (resp. cell 2) at first time step reaches to cell 2 (resp. cell 1) at third time step due to delay perturbation. However, in Fig. 4(b), the information about state change of cell 2 at first time step does not reach to cell 1 at third time step due to probabilistic loss of information perturbation.

3 It means, all the delay (between 1 to D) are equally probable.

Table 1
Look-up table for ECA 51, 43, 129 and 22.

	111	110	101	100	011	010	001	000
51:	0	0	1	1	0	0	1	1
43:	0	0	1	0	1	0	1	1
129:	1	0	0	0	0	0	0	1
22:	0	0	0	1	0	1	1	0

2.3. Experimental protocol

Recall that, in this work, we mainly concentrate on two simple models: elementary cellular automata (ECA) and Game of Life. In order to understand the behavioural changes between the classical CA system and the proposed CA system, we compare the space–time diagrams and *density* variations. Following is the exact description of the two experimental methods.

• In the first approach, i.e. qualitative behaviour, we need to look at the evolution of the configurations (i.e. the space–time diagrams) by eye over a few time steps. Though this is not a formal method, but this approach can provide a good visual comparison.

• In the second approach, we study the ratio of cell with actualstate 1 in the formal second approach, i.e. quantitative behaviour. The *density* of a configuration $x \in S^L$ can be written as $d(x) = \#_1 x/|x|$, where $\#_1 x$ is the number of 1's in the configuration x and $|x|$ is the size of the configuration. Here, without loss of generality, we start with a configuration of initial density 0.5. Note that here, the initial configuration with density 0.5 is constructed using *Bernoulli* process.[4] Now, in a natural way, the experimental protocol is different for elementary cellular automata and Game of Life. For elementary cellular automata, we start with a configuration of CA size 500. Here, we let the system evolve during 2000 time steps and average the density parameter value for 100 time steps. Similarly, a 50×50 configuration evolves for 1000 time steps in the case of Game of Life.

Now, in the following sections, we explore the behaviour of elementary cellular automata and Game of Life for the proposed updating schemes following the experimental approach mentioned above.

3. Elementary cellular automata: Observations

In this section, we discuss the space–time diagram and density parameter behaviour of elementary cellular automata (ECA) for the proposed updating schemes which involve delay and/or probabilistic loss of information during information sharing among neighbours. In this scenario, we first define the preliminaries of ECA.

3.1. Preliminaries of ECA

The one-dimensional three-neighbour binary (i.e. two state: {0,1}) cellular automata with periodic boundary condition, where the cells are arranged as a ring, are commonly known as *elementary cellular automata* (ECA) [15]. Here, a cell changes its state depending on left neighbour, self, right neighbour. Therefore, the function $f : \{0,1\}^3 \to \{0,1\}$ can be expressed as a look-up table (see Table 1), which is called ECA "rule". Each "rule" is associated with a 'decimal code' w, where $w = f(0, 0, 0) \cdot 2^0 + f(0, 0, 1) \cdot 2^1 + \cdots + f(1, 1, 1) \cdot 2^7$, for the naming purpose. There are $2^8 = 256$ ECA rules in two-state three-neighbour dependency. Through the use of left/right reflexion and 0/1 complementarity, it is possible to narrow down the 256 ECA rule space to 88 classes, each represented by the rule of smallest number, i.e. the minimal representative ECA rule [16]. In our work, we consider 88 minimal representative ECA rules. Four such ECA rules (51, 43, 129 and 22) are shown in Table 1.

3.2. Observations

Here, ECA shows following peculiar behaviour for the proposed updating schemes.

- *Probabilistic loss of information perturbation*: To study the effect of the proposed probabilistic loss of information perturbation updating scheme in order to compare with classical CA system, we classify the ECA rules in the following categories (see Table 2) according to their qualitative and quantitative behaviour—(1) **Category** 1: No difference can be observed according to both qualitative and quantitative behaviour (see ECA 51 as an example case in Fig. 5); (2) **Category** 2: The space–time behaviour shows a radical difference but the quantitative behaviour remains constant (see ECA 43 as an example case in Fig. 5); and (3) **Category** 3: A radical difference can be observed according to both of the parameters (see ECA 22 and 129 as an example case in Fig. 5). Some ECA rules of this category (ECA **18**, **26**, **50**, **106**, and **146**) show a discontinuity behaviour after a critical value of probabilistic loss of information perturbation rate (see

[4] Here, the initial density is constructed in the following method using Bernoulli process: every cell has a probability 0.5 (resp. 1 − 0.5) to have state 1 (resp. 0) [14].

Table 2
Classification of the minimal ECA rules for the proposed updating schemes. The ECA rules in bold (resp. underlined) show discontinuity (resp. *closeness towards* discontinuity) behaviour.

Probabilistic loss of information perturbation	Category 1	0, 4, 8, 12, 13, 32, 36, 40, 44, 51, 72, 76, 77, 78, 104, 128, 132, 136, 140, 160, 164, 168, 170, 200, 204, 232
	Category 2	5, 7, 14, 15, 23, 27, 28, 29, 30, 35, 37, 41, 43, 45, 54, 57, 60, 90, 105, 108, 131, 133, 142, 150, 156
	Category 3	1, 2, 3, 6, 9, 10, 11, **18**, 19, 22, 24, 25, **26**, 33, 34, 38, 42, 46, **50**, 56, 58, 73, 74, **106**, 129, 130, 134, 137, 138, **146**, 152, 154, 161, 162, 170, 178, 184
Delay perturbation	Category 1	0, 4, 8, 12, 13, 32, 36, 40, 44, 51, 72, 76, 77, 78, 104, 128, 132, 136, 140, 160, 164, 168, 170, 200, 204, 232
	Category 2	3, 5, 6, 7, 11, 14, 15, 19, 23, 26, 27, 29, 30, 33, 34, 37, 38, 41, 43, 45, 50, 54, 57, 60, 73, 90, 105, 106, 108, 131, 133, 134, 142, 150, 154, 156, 170, 178, 184
	Category 3	1, 2, 9, 10, 18, 22, 24, 25, 28, 35, 42, 46, 56, 58, 74, 129, 130, 137, 138, 146, 152, 161, 162
Delay and probabilistic loss of information perturbation	Category 1	0, 4, 8, 12, 13, 32, 36, 40, 44, 51, 72, 76, 77, 78, 104, 128, 132, 136, 140, 160, 164, 168, 170, 200, 204, 232
	Category 2	5, 7, 14, 15, 23, 27, 29, 30, 37, 41, 43, 45, 54, 57, 60, 90, 105, 108, 131, 133, 142, 150, 156
	Category 3	1, 2, 3, **6**, 9, 10, 11, **18**, 19, 22, 24, 25, **26**, 28, 33, 34, 35, **38**, 42, 46, **50**, 56, 58, 73, 74, **106**, 129, 130, **134**, 137, 138, **146**, 152, 154, 161, 162, 170, 178, 184

Table 2, in bold). This discontinuity behaviour has also been previously identified by Bouré et al. [7] for γ-synchronism updating scheme.[5]

- *Delay perturbation:* Table 2 depicts the categorical classification of the minimal ECA rules for the delay perturbation updating scheme following the same classification criteria. Remark that none of the ECA rules show discontinuity behaviour for delay perturbation updating scheme. However, some ECA rules (see Table 2, the ECA rules in underline symbol) show a *closeness towards* discontinuity for the delay perturbation updating scheme. Here, as the D parameter value gets higher, the chance of discontinuity for these ECA rules get higher (see ECA 152 as an example case in Fig. 5). Now, the genuine question is that, can we get an *absolute* discontinuity after some critical value of D parameter? Therefore, we need to study the proposed CA system at local level to address this question. In order to study the local evolution for delay perturbation updating scheme, we have focused on ECA 152 as an example (see Appendix A).
- *Delay and probabilistic loss of information perturbation:* Table 2 depicts the categorical classification of the minimal ECA rules for the delay and probabilistic loss of information perturbation updating scheme following the same classification criteria. Some ECA rules of Category 2 and 3 show remarkable changes for this updating scheme which involves delay and information loss both. Eight ECA rules (ECA **6**, **18**, **26**, **38**, **50**, **106**, **134** and **146**) show a discontinuity behaviour after a critical value of probabilistic loss of information perturbation rate for this updating scheme (see Table 2, in bold). This critical behaviour is confirmed by both space–time diagram and density parameter (see Fig. 6). Note that 5 ECA rules (ECA 18, 26, 50, 106 and 146), out of 8, also show this discontinuity behaviour for the probabilistic loss of information perturbation updating scheme. For the remaining 3 ECA rules (ECA 6, 38 and 134), neither delay perturbation nor probabilistic loss of information perturbation updating scheme shows this peculiar discontinuity behaviour. In the next section, we will discuss about this peculiar discontinuity behaviour, which is called a *phase transition*, in the context of statistical physics.

4. Study of phase transitions

According to the literature of CA, the occurrence of phase transition is the most interesting phenomena of different non-classical CA model [7,17–20]. The occurrence of phase transition can be defined in the following way: there exists a critical value of *degree of perturbation*[6] which distinguishes the behaviour of the system in two different 'phases'—*passive phase* (i.e. the system converges to a homogeneous fixed point of all 0's) and *active phase* (i.e. the system oscillates around a fixed non-zero density).

For the α, β and γ-synchronous updating scheme, this brutal change of behaviour was noted by [7,21,18]. The phase transition for ECA with memory was identified by [19,20]. Recently, this abrupt change of behaviour has been studied by

5 Note that the probabilistic loss of information perturbation corresponding to the γ-synchronism updating scheme in Ref. [7].

6 Here, *degree of perturbation* means synchrony rate of a non-classical updating scheme or mixing rate of different CA rule or perturbation rate of any other perturbation scheme.

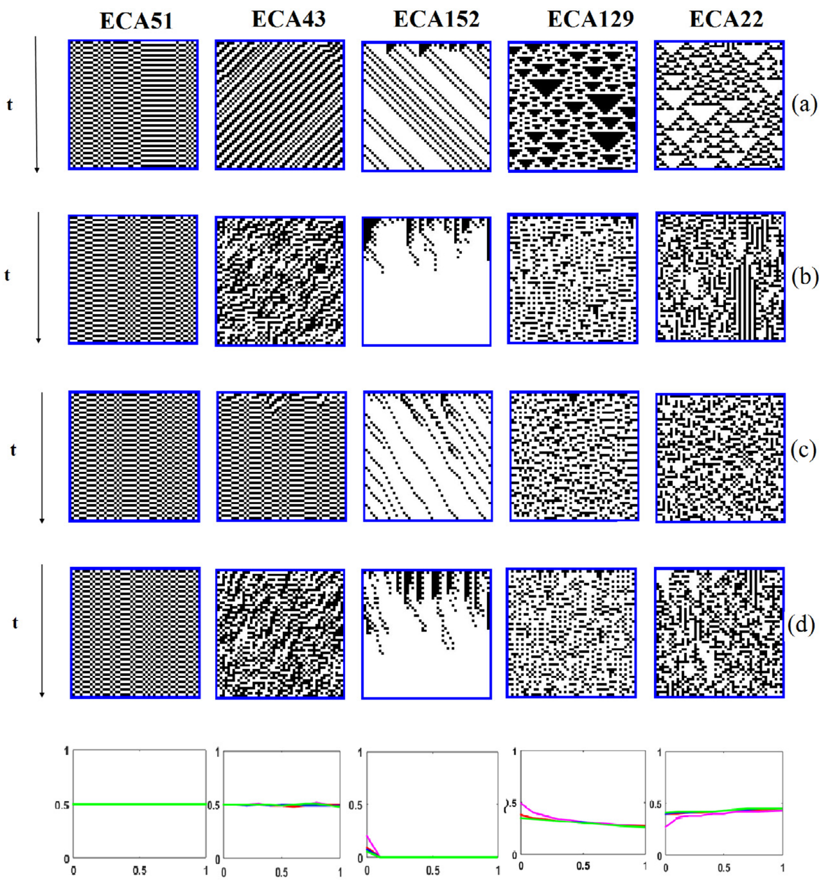


Fig. 5. The samples of space–time diagrams for the proposed updating schemes—(a) synchronous; (b) $D = 1$, $\iota = 0.5$; (c) $D = 2$, $\iota = 0.0$; and (d) $D = 2$, $\iota = 0.5$, starting from a random initial configuration. The plot shows the profile of density parameter as a function of the probabilistic loss of information perturbation rate for a fixed D parameter value, where $D = 1$: Magenta, $D = 2$: Red, $D = 3$: Blue, and $D = 4$: Green colour.

Fatés [22] for *Diploid*[7] cellular automata. The occurrence of phase transition appears for the proposed updating schemes which involve delay and/or information loss in the following way.

(i) **Probabilistic loss of information perturbation:** For ECA **18**, **26**, **50**, **106** and **146**, a phase transition is observed for the probabilistic loss of information perturbation updating scheme. Note that this result of phase transition for probabilistic loss of information perturbation updating scheme has been previously identified by Bouré et al. [7] under the name of γ-synchronism updating scheme. According to [7], this phase transition belongs to the directed percolation universality class. Figs. 6 and 7 show ECA 50 as an example of this phase transition.

(ii) **Delay perturbation:** This phase transition result for any ECA rule is not observed for the proposed delay perturbation updating scheme.

(iii) **Delay and probabilistic loss of information perturbation:**

- For ECA **18**, **26**, **50**, **106** and **146**, a phase transition is observed for delay and probabilistic loss of information perturbation updating scheme. A phase transition for these ECA rules is also observed for only probabilistic loss of information perturbation. However, the critical value (of probabilistic loss of information perturbation rate) for phase transition increases when the updating scheme is also associated with delay perturbation (see Figs. 6 and 7, example case: ECA 50).
- Now, for ECA **6**, **38**, and **134**, a phase transition also occurs for delay and probabilistic loss of information perturbation updating scheme (see Figs. 6 and 7, example case: ECA 38). However, a phase transition for these ECA rules cannot be observed for the probabilistic loss of information perturbation updating scheme. Note that

[7] The rules of diploid cellular automata are obtained with random mixing of two deterministic ECA rules.

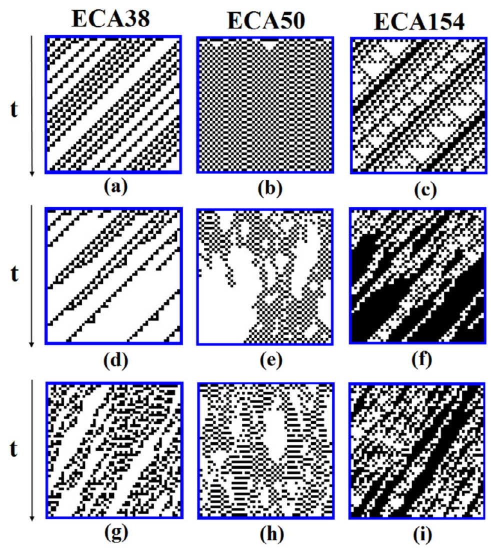


Fig. 6. The samples of space–time diagrams for ECA rules which show phase transition for the updating schemes—(a), (b), (c): synchronous; (d), (e): $D = 1$, $\iota = 0.5$; (f): $D = 1$, $\iota = 0.1$; (g), (h): $D = 2$, $\iota = 0.5$; and (i): $D = 2$, $\iota = 0.1$, starting from a random initial configuration.

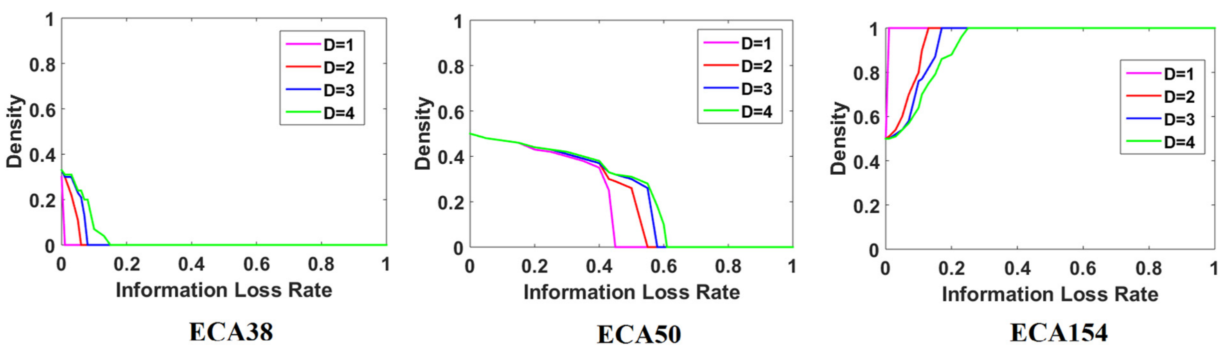


Fig. 7. The plot shows the profile of density parameter as a function of the probabilistic loss of information perturbation rate with a fixed D parameter for ECA rules.

Table 3
The critical value for phase transition of ECA 6, 38, 134 and 154.

ECA	$\iota^c_{D=1}$	$\iota^c_{D=2}$	$\iota^c_{D=3}$	$\iota^c_{D=4}$
6	–	0.02	0.04	0.07
38	–	0.05	0.07	0.13
134	–	0.03	0.06	0.11
154	–	0.11	0.15	0.23

these ECA rules show phase transition for very low critical value of probabilistic loss of information perturbation rate, see Table 3. In Table 3, $\iota^c_{D=k}$ indicates the critical value of probabilistic loss of information perturbation rate for phase transition with D parameter value k. Although it is not in the scope of the current work, we still want to mention that these three ECA rules also change their phases for α-synchronous updating scheme [7]. Concluding any similarity from this similar *signature* behaviour of these two updating scheme (i.e. the proposed updating scheme and α-synchronous updating scheme) is still an open question.

- Here, for ECA **154**, a different kind of phase transition behaviour is observed for delay and probabilistic loss of information perturbation updating scheme (see Fig. 6). Here, the system converges to a homogeneous fixed point of all 1's for large value of probabilistic loss of information perturbation rate, whereas, the system does not stabilize at fixed point for small value of probabilistic loss of information perturbation rate (see Fig. 7, Table 3).

Now, we need to study the proposed updating schemes at microscopic level to understand the reason of these different behaviours. In order to study the local evolution of cells for the proposed updating schemes, we have focused on the following two ECA rules which represent different phase transition results for different updating schemes—(i) ECA 38: which shows phase transition for the delay and probabilistic loss of information perturbation updating scheme; but not for probabilistic loss of information perturbation updating scheme (see Appendix B); and (ii) ECA 50: which shows phase transition for both the probabilistic loss of information perturbation updating scheme; and the delay and probabilistic loss of information perturbation updating scheme (see Appendix C). Note that, here, the microscopic approach is not aimed to construct a formal proof about the behaviour of the proposed CA system. However, the target is to give an insight about the behaviour of the proposed CA system for different updating schemes.

In the next section, we explore the behaviour of Game of Life for the proposed updating schemes following the same experimental approach.

5. The game of life: Observations

The Game of Life was created by John Horton Conway and has been made famous by Martin Gardner [23].

5.1. Preliminaries of game of life

Classically, the Game of Life can be defined on a squared two-dimensional lattice of cells which are in either *alive* or *dead* state. The neighbourhood of each cell is constituted of the cell itself and the eight nearest neighbours, i.e. *Moore neighbourhood*. Here, we use the *periodic* boundary condition. Note that the type of boundary conditions plays an important role for Game of Life [24]. Game of Life transition function can be written as: (1) A live cell will remain alive if exactly two or three of its eight nearest neighbours are alive, otherwise it will die; and (2) If a dead cell has exactly three live neighbours, it will be toggled to the live state (birth). Note that the simple Game of Life can also be capable to perform universal computation [25].

In the literature of CA, the behaviour of Game of Life under different non-classical CA model attracts a huge attention of researchers [26,8,27–29]. For small-world network, the behaviour of Game of Life was noted in [27]. Here, a nonequilibrium phase transition from an "*inactive-sparse*" state to an "*active-dense*" state had been observed for a certain intermediate value of the network disorder. The brutal change of behaviour between classical (synchronous) version of Game of Life and sequential (asynchronous) updating of Game of Life was studied by Bersini and Detours [29]. Here, "*labyrinth phase*" was observed for sequential updating of Game of Life, see [29]. A phase transition for Game of Life from an "*inactive-sparse phase*" to a "*labyrinth phase*" was also observed by [8,28] for the value of synchrony rate.

5.2. Observations

In this scenario, we describe the space–time diagram and density parameter behaviour induced by the introduction of delay and/or probabilistic loss of information during information sharing between neighbours. Here, Game of Life shows following peculiar behaviour for the proposed updating schemes.

(i) **Probabilistic loss of information perturbation:** Fig. 8 shows that probabilistic loss of information perturbation produces a qualitative and quantitative behaviour change for Game of Life. Here, an "*inactive-sparse*" phase is observed for lower probabilistic loss of information perturbation rate, whereas, an "*active-dense*" phase is observed for higher probabilistic loss of information perturbation rate. The "active-dense" phase gradually appears after certain critical value of probabilistic loss of information perturbation rate, see Fig. 8(B). According to [26], this type of phase transition is continuous or second-order transition. In this context, note that Fatés [8,28] also studied the Game of Life for short range of probabilistic link removal.

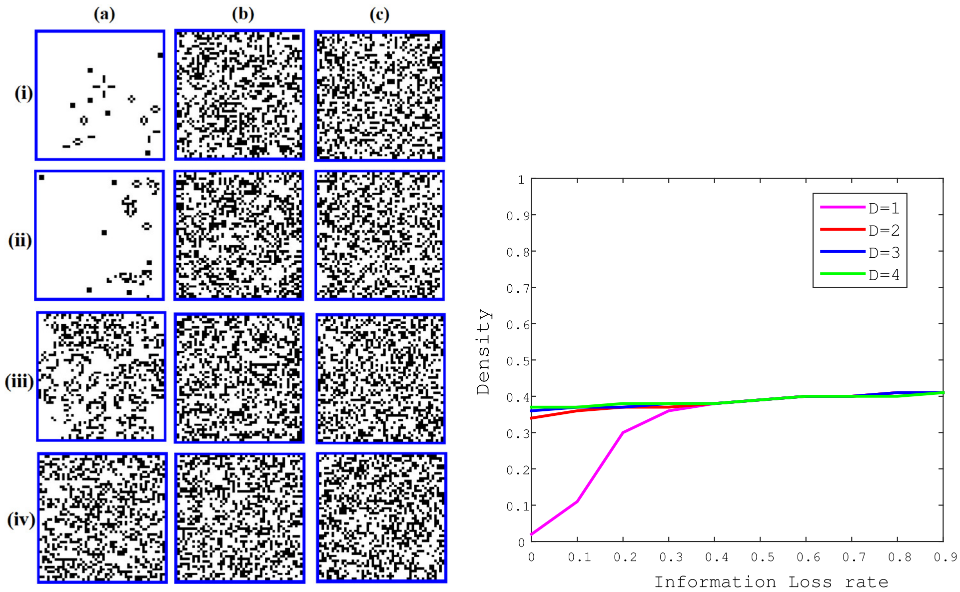
(ii) **Delay perturbation:** Game of Life shows an "*inactive-sparse*" phase for the classical CA updating scheme. However, an "*active-dense*" phase is observed for delay perturbation updating scheme where delay perturbation parameter $D \geq 2$. Though, the "*active-dense*" phase does not appear gradually. Rather, we can say that the "*active-dense*" phase appears abruptly due to the effect of delay. Fig. 8 shows that the space–time diagram of Game of Life depends on the effect of delay perturbation.

(iii) **Delay and probabilistic loss of information perturbation:** Finally, we will discuss about the behaviour of Game of Life for delay and probabilistic loss of information perturbation updating scheme. Fig. 8 shows that the qualitative and quantitative behaviours remain almost same when both delay and probabilistic loss of information are added. However, that behaviour is different from classical Game of Life.

6. Discussion

In this study, the paper introduced a new kind of cellular automata where a cell shares its state with its neighbours with some delay and probabilistic loss of information. This study highlighted following different responses of phase transition.

- For ECA **18**, **26**, **50**, **106**, **146** and **Game of Life**, a phase transition occurs for probabilistic loss of information perturbation updating scheme.



(A) (B)

Fig. 8. (A): Game of Life configurations for 50×50 after $t = 1000$ time steps, starting from a random initial configuration. The columns display with delay perturbation parameter (a) $D = 1$, (b) $D = 2$ and (c) $D = 3$; and the rows display with probabilistic loss of information perturbation rate (i) $\iota = 0.0$, (ii) $\iota = 0.1$, (iii) $\iota = 0.2$ and (iv) $\iota = 0.3$; (B): The plot shows the profile of density parameter as a function of the probabilistic loss of information perturbation rate with a fixed D parameter for Game of Life.

- For none of the ECA rule, a phase transition occurs for delay perturbation updating scheme. However, **Game of Life** shows abrupt change in phase due to the effect of delay perturbation.
- For ECA **6**, **18**, **26**, **38**, **50**, **106**, **134**, **146** and **154**, a phase transition occurs for delay and probabilistic loss of information perturbation updating scheme.

Here, different kind of signature behaviours for different versions of the proposed updating scheme seem very promising according to modelling point of view. In the context of modelling complex systems (i.e. physical, biological, social systems), various cellular automata models, proposed yet, have been successful to predict more realistic results in the presence of time delay [5,6]. In real-world complex systems, information sharing delay and probabilistic loss of information cannot generally be ignored. However, the literature of CA for modelling complex systems have been mainly concentrated on selection of proper rules without considering delay and probabilistic loss of information. Depending on the wide variety of results for the proposed updating schemes, the study can strongly conclude that the CA system considering delay and probabilistic loss of information can open many emergent aspects of modelling complex systems. However, the proper theoretical understanding to predict the behaviour of CA for the proposed updating schemes is still an open question.

This research, however, can also be extended in the following directions:

- Here, the delays are non-uniform in space but uniform in time. Therefore, an immediate research may be conducted to study the effect of non-uniform delay in both space and time.
- Here, the proposed CA system follows synchronous updating scheme. However, the choice of asynchronous updating scheme with independent cell dynamics as the models of complex and distributed systems is better [30,31,7,32]. Therefore, a research may be conducted to study the effect of asynchronous updating scheme on the proposed CA system.

Acknowledgements

The author is grateful to Pinaki Chakraborty for his work on this topic as an undergraduate final year project student. The author is also grateful to Dr. Sukanta Das for enlightening discussions on this topic. The author acknowledge the anonymous reviewers for their comments and suggestions, which have helped to improve the quality and readability of the paper.

Appendix A. Case study: ECA 152

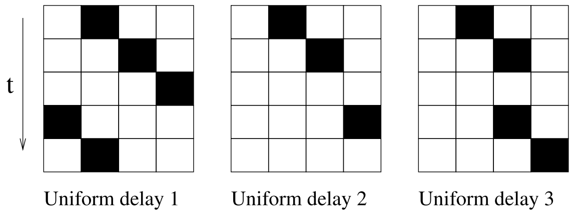


Fig. A.9. Evolution of a branch of state '1' cell for ECA 152.

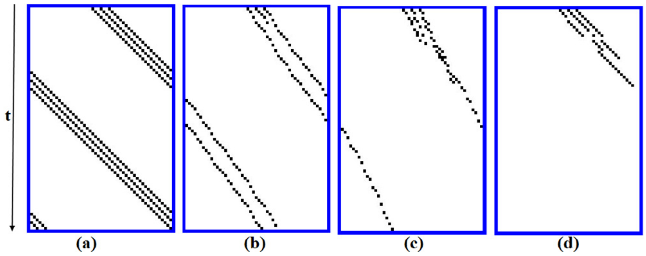


Fig. A.10. Samples of space–time diagrams of ECA 152 for following updating schemes : (a) $D = 1$, $\iota = 0.0$; (b) $D = 2$, $\iota = 0.0$; (c) $D = 3$, $\iota = 0.0$; (d) $D = 1$, $\iota = 0.3$.

Here, we discuss about the behaviour of ECA 152, which shows a *closeness towards* discontinuity for delay perturbation updating scheme. Now, to understand the existence of *absolute* discontinuity for ECA 152, we need to study the state change of cells at microscopic level. Note that, for ECA 152, a discontinuity always exists for probabilistic loss of information perturbation; and delay and probabilistic loss of information perturbation updating scheme. Following is the transition table for ECA 152.

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

A configuration of ECA 152 can be viewed as regular right shift branches of state '1' cell which is surrounded by state '0' cell, see Fig. A.9. To analyse the effect of delay perturbation in the evolution of ECA 152, we first consider the evolution of a single branch of state '1' cell for the updating scheme which involves same delay for all the cells during information sharing (i.e. *uniform delay*). Due to the effect of this delay, the regular shift of a branch is postponed for some time steps based on the delay value. Fig. A.9 shows the evolution of one single branch for the updating scheme with uniform delay 1, 2 and 3. As the updating scheme with uniform delay is a special case of the proposed updating scheme with (non-uniform) delay, the proposed updating scheme also reflects this restricted shift behaviour of branches.

For the proposed updating scheme with (non-uniform) delay, different branches shift regularly with different speeds. Therefore, two neighbouring branches can collide with high probability. As a result, one branch, the left one, can be destroyed. The destruction of left branch takes place following the transition function $110 \rightarrow 0$ and $101 \rightarrow 0$. However, there is no scope of 'creation' of new branch. Fig. A.10 shows the evolution of ECA 152 starting from a initial configuration with three branches of state '1' cell (i.e. $\cdots 00100100100 \cdots$). Remark that as the D parameter value gets higher, the non-uniformity in delay and the chance of collision increase. However, one single branch always remains for the worst possible case also. Therefore, for ECA 152, the *absolute* discontinuity does not exist for the delay perturbation updating scheme. Finally, remark that, for ECA 152, a *leader preserving tendency* [33–35] is observed for the delay perturbation updating scheme.

Appendix B. Case study: ECA 38

In this case study, we discuss about the peculiar behaviour of ECA 38 which shows a phase transition for delay and probabilistic loss of information perturbation updating scheme, but, a phase transition is not observed for probabilistic loss of information perturbation updating scheme. Now, to understand these two separate behaviours of ECA 38, we need to study the state change of cells at microscopic level. Following is the transition table for ECA 38.

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

The evolution of ECA 38 can be viewed as regular left shift branches of state '1' cell which is surrounded by state '0' cell, see Fig. B.11. However, "restricted shift", "creation", and "death" of those branches occur for the proposed updating schemes.

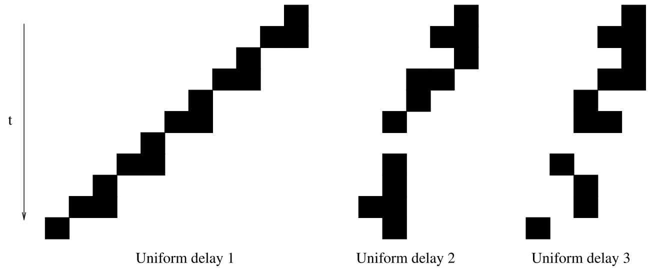


Fig. B.11. Evolution of a branch of state ‘1’ cell for ECA 38.

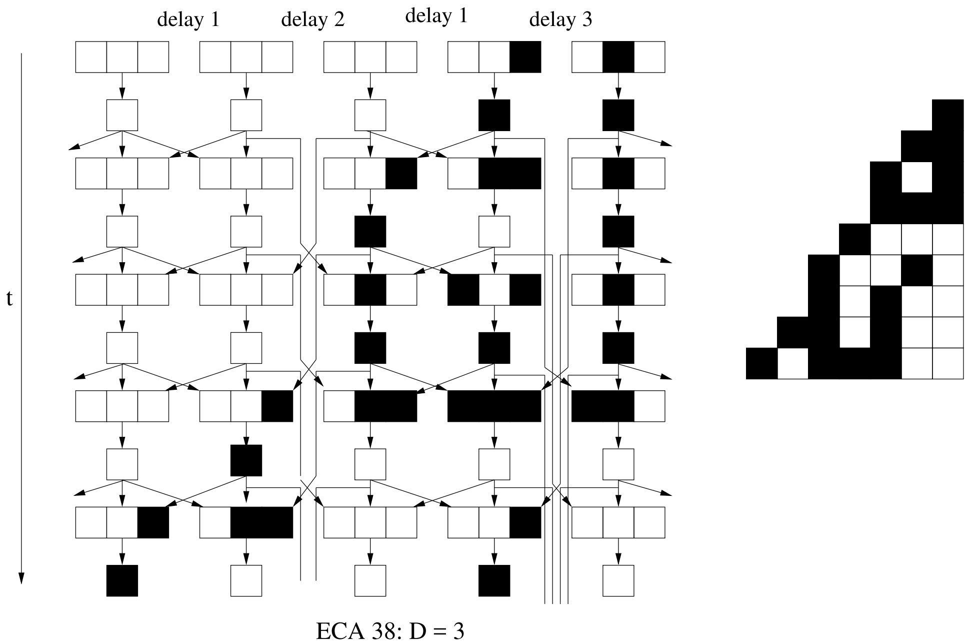


Fig. B.12. Branch generation capability of ECA 38 for the delay perturbation updating scheme with $D = 3$. Here, the stable *viewstate* informations remain hidden during the intermediate state update step for simplicity. Note that, throughout the rest of paper, we follow the same style.

In order to understand the behaviour of the proposed CA system, we analyse the system for the following two updating schemes—(i) delay perturbation; and (ii) probabilistic loss of information perturbation.

(i) Delay perturbation: To understand the effect of delay perturbation in the dynamics of ECA 38, we first consider the evolution of a single branch of state ‘1’ cell for the updating scheme which involves *uniform* delay, i.e. all the cells involve same delay during information sharing. Recall that a updating scheme with uniform delay is a special case of the proposed updating scheme with (*non-uniform*) delay. Under the classical CA environment (where, the uniform delay value = 1), this branches shift left regularly without any disturbance. However, a “restricted shift” behaviour of a branch can be observed for the updating scheme which involves uniform delay (> 1), see Fig. B.11 for uniform delay value 1, 2 and 3. This obvious behaviour is also reflected for the proposed updating scheme with (non-uniform) delay.

Now, we consider the proposed actual updating scheme which involves (non-uniform) delay during information sharing between two neighbours. As an example, Fig. B.12 shows the creation and death of branches for the delay perturbation updating scheme. For this updating scheme, the branches are created and destroyed in the following circumstances: (a) Recall that the branches shift left depending on the following transition functions: 001 $\rightarrow$ 1 and 010 $\rightarrow$ 1. Now, let us consider a situation, a cell in actualstate ‘0’ changes its actualstate to ‘1’ following the transition function 001 $\rightarrow$ 1, but its left neighbouring cell still follows transition function 010 $\rightarrow$ 1 due to the effect of delay. Therefore, the left neighbouring cell can again be capable of activating branch. However, the (left neighbouring) cell cannot activate any branch until some space is created between branches. For delay perturbation updating scheme, non-uniform delay plays the roll to create space between branches, see Fig. B.12. Here, once a branch becomes an isolated branch, it remains stable. Remark that as the D parameter value gets higher, the non-uniformity in delay and the chance of new branch creation increase. The high chance of new branch creation starting from a single branch is confirmed by experiments, see Fig. B.13; (b) Two consecutive

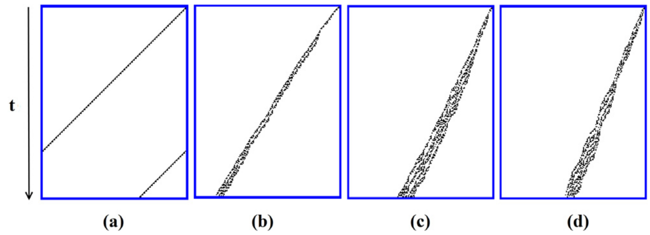


Fig. B.13. Samples of space–time diagrams of ECA 38 starting from a single branch for following updating schemes : (a) $D = 1$; (b) $D = 2$; (c) $D = 3$; (d) $D = 4$.

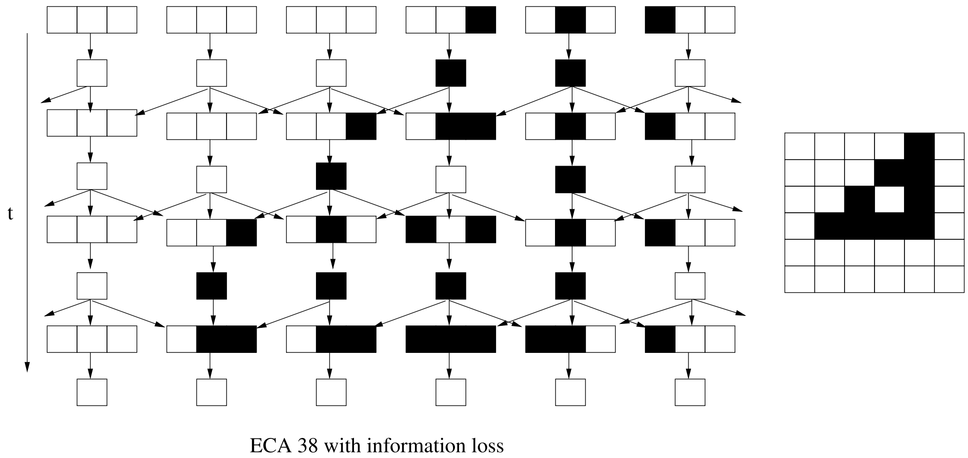


Fig. B.14. Convergence capability of ECA 38 for the probabilistic loss of information perturbation updating scheme.

branches can also activate branches by following the transition function $101 \rightarrow 1$ (see Fig. B.12). However, these branches are destroyed immediately without activating neighbours as there exists no space between branches to survive; (c) For delay perturbation updating scheme, different branches shift regularly with different speed due to the effect of non-uniform delay. Hence, one branch can also be destroyed as a result of collision.

According to the discussion, we can conclude the following remarks about delay perturbation updating scheme—(1) There exists a high possibility of branch creation whenever some space is created. Moreover, an isolated branch remains stable (case a); (2) A branch without any space cannot survive in this environment (case b, c); Therefore, for ECA 38, there always exists a non-convergence phase for delay perturbation updating scheme which can be described as a group of isolated branches, see Fig. 6 for evidence.

(ii) Probabilistic loss of information perturbation: Now, we consider probabilistic loss of information perturbation updating scheme. Fig. B.14 shows an example of creation and death of branches for this updating scheme. Here, the branches are created and destroyed in the following circumstances: (a) a branch is associated with a high chance of 'death' for a single *mistake*, i.e. a cell moves to actualstate '1' but its right neighbouring cell still has a viewstate '0' about its (left) neighbour, see Fig. B.14; (b) This updating scheme also has a capacity of branch creation. Let us consider a situation, a cell in actualstate '0' changes its actualstate to '1' following the transition function $001 \rightarrow 1$. However, its (left) neighbouring cell still follows transition function $010 \rightarrow 1$ due to the effect of information loss and can again be capable to generate new branch. However, these branches cannot survive due to lack of space. Whereas, the branch generation capability of that (left neighbouring) cell disappears as soon as that cell is associated correct viewstate information about neighbouring cells, see Fig. B.14.

Remark that a convergence phase always exists for probabilistic loss of information perturbation updating scheme as (1) The branches always have a high chance of "dead" for small probability of information loss (case a); and (2) a new branch cannot survive in the proposed environment (case b).

Now, we consider the updating scheme which combines delay and information loss. In order to describe the behaviour of this updating scheme, we analyse the following two events : (i) 'creation' of branch with the effect of 'delay perturbation'; and (ii) 'death' of branch with the effect of 'probabilistic loss of information perturbation'. For low value of probabilistic loss of information, the occurrence of new branch creation is more probable than the death of a branch. However, an opposite behaviour, where the death of a branch is more probable than the creation of a branch, is observed for high value of probabilistic loss of information. Therefore, two different phases may occur for two different values of probabilistic loss of information. This phase transition is also confirmed by experiments, see Fig. 7.

To sum up, the above arguments can provide a possible explanation for the two different behaviours for two different updating schemes.

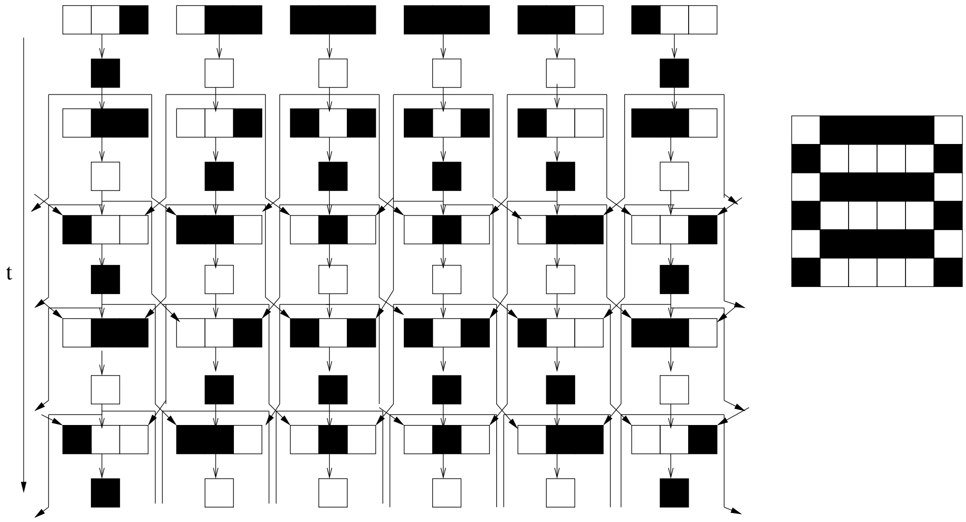


Fig. C.15. Particles of length ⩾ 2 for ECA 50.

Appendix C. Case study: ECA 50

Now, we discuss about the behaviour of ECA 50 which shows phase transition for probabilistic loss of information perturbation updating scheme. However, the *critical value* (i.e. probabilistic loss of information perturbation rate) for phase transition increases when the updating scheme is also associated with delay perturbation (see Fig. 7). To understand these separate critical values for phase transition, we need to study the space–time diagrams at a microscopic level. By looking at the space–time diagrams for the proposed updating schemes with only delay perturbation; with only probabilistic loss of information perturbation; and with both delay and probabilistic loss of information perturbation, we clearly observe a notable difference in behaviour. Note that ECA 50 can be expressed by the following transition table.

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

The space–time diagram of ECA 50 for the classical CA environment can be viewed as the following patterns: *chequerboard pattern* (i.e. the cells are in alternative actualstates and will flip their actualstates during update) and *particles* of length two (i.e a pair of two cells are in same actualstates and both of them will flip their actualstates during update), see Fig. C.16(a). However, the following behavioural radical change in space–time diagram is observed for the proposed updating schemes.

• The *particles* of length ⩾ 2 are observed for delay perturbation updating scheme, see Fig. C.15. In Fig. C.15, the delay value is two time steps for all the cells.

• The "chequerboard" and "particle" (of length two) patterns are destroyed during the evolution of proposed CA system for probabilistic loss of information perturbation updating scheme. Here, "white holes" are created in the "chequerboard" pattern due to the effect of probabilistic loss of information. On the other hand, particles can displace them spatially or can destroy them alone or by pair due to the effect of probabilistic loss of information (see Fig. C.16(b)).

• The effect of delay and probabilistic loss of information perturbation updating scheme shows the following interesting patterns: (i) *particle growth*: the length of the particle increases, see Fig. C.16(c); (ii) *particle compression*: the length of the particle decreases, see Fig. C.16(c); (iii) *particle unification*: two particles of same or different length are concatenated with each other, see Fig. C.16(d); and (iv) *particle division*: one particle is divided into two or more particles, see Fig. C.16(d).

Now, we will discuss about the effect of these microscopic behaviours on the global behaviour of ECA 50 for the following updating schemes.

1. First, we consider the proposed probabilistic loss of information perturbation updating scheme. Recall that the "chequerboard" and "particle" patterns are destroyed due to the effect of probabilistic loss of information. However, the natural tendency of ECA 50 is to 'repair' those patterns. Therefore, for low value of probabilistic loss of information, the 'repair' process is more probable than the 'damage' process. However, for high value of probabilistic loss of information, the 'damage' process is more probable than the 'repair' process. Hence, for ECA 50, two different phases occur for probabilistic loss of information perturbation updating scheme.
2. Now, we consider delay and probabilistic loss of information perturbation updating scheme. Here, for low value of probabilistic loss of information, *particle growth* and *unification* is more probable than *particle compression* and *division*. However, an opposite behaviour is observed for high value of probabilistic loss of information. Therefore, two different phases may occur for two different values of probabilistic loss of information. This phase transition is also confirmed by experiments, see Fig. 7.

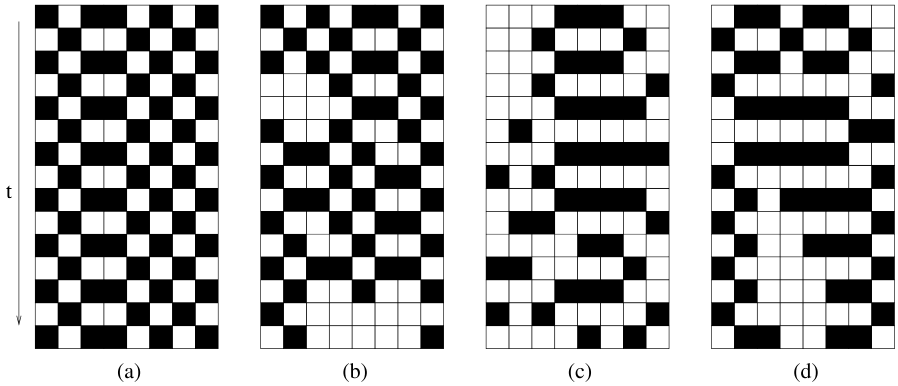


Fig. C.16. Sample patterns of space–time diagrams of ECA 50: (a) chequerboard pattern and particles of length two; (b) white holes, spacial displacement of particles, and destroy of particles; (c) particle growth and compression; and (d) particle unification and division.

Here, length-two particles are observed for probabilistic loss of information perturbation updating scheme, whereas, large particles are observed for delay and probabilistic loss of information perturbation updating scheme. A single information loss can be able to break a two length particle (i.e. breaking a transition function 100 $\rightarrow$ 1 or 001 $\rightarrow$ 1). However, multiple (two) information losses can only be able to break a large particle (i.e. breaking a transition function 101 $\rightarrow$ 1). Therefore, the critical value for the delay and probabilistic loss of information perturbation updating scheme is greater than the critical value for the probabilistic loss of information perturbation updating scheme (see Fig. 7 for evidence). Remark that as the D parameter gets higher, the critical value for phase transition also increases (see Fig. 7). This increasing tendency of critical value depending on the D parameter is still an open question for us.

References

[1] J. von Neumann, in: A.W. Burks (Ed.), Theory of Self-Reproducing Automata, University of Illinois Press, Champaign, IL, USA, 1966.
[2] K. Bhattacharjee, N. Naskar, S. Roy, S. Das, A survey of cellular automata: types, dynamics, non-uniformity and applications, Nat. Comput. (2018).
[3] J. Kari, Theory of cellular automata: A survey, Theoret. Comput. Sci. 334 (1) (2005) 3–33.
[4] H. Attiya, J.L. Welch, Distributed Computing: Fundamentals, Simulations and Advanced Topics, Mc Graw-Hill, London-UK, 1998.
[5] K.C. Iarosz, C.C. Martins, A.M. Batista, R.L. Viana, S.R. Lopes, I.L. Caldas, T.J.P. Penna, On a cellular automaton with time delay for modelling cancer tumors, J. Phys. Conf. Ser. 285 (1) (2011) 012015.
[6] N. Sharma, A.K. Gupta, Impact of time delay on the dynamics of SEIR epidemic model using cellular automata, Physica A 471 (Supplement C) (2017) 114–125.
[7] O. Bouré, N. Fatès, V. Chevrier, Probing robustness of cellular automata through variations of asynchronous updating, Nat. Comput. 11 (4) (2012) 553–564.
[8] N. Fatès, M. Morvan, Perturbing the topology of the game of life increases its robustness to asynchrony, in: P.M.A. Sloot, B. Chopard, A.G. Hoekstra (Eds.), Cellular Automata: 6th International Conference on Cellular Automata for Research and Industry, ACRI 2004, Amsterdam, The Netherlands, October 25-28, 2004. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 111–120.
[9] R. Alonso-Sanz, Reversible cellular automata with memory of delay type, Complexity 20 (1) (2014) 49–56.
[10] R. Alonso-Sanz, Elementary cellular automata with memory of delay type, in: J. Kari, M. Kutrib, A. Malcher (Eds.), Cellular Automata and Discrete Complex Systems: 19th International Workshop, AUTOMATA 2013, Gießen, Germany, September 17-19, 2013. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 67–83.
[11] V. García-Morales, From deterministic cellular automata to coupled map lattices, J. Phys. A 49 (29) (2016) 295101.
[12] V. García-Morales, J.A. Manzanares, S. Mafe, Weakly coupled map lattice models for multicellular patterning and collective normalization of abnormal single-cell states, Phys. Rev. E 95 (2017) 042324.
[13] S. Adachi, F. Peper, J. Lee, The game of life at finite temperature, Physica D 198 (3) (2004) 182–196.
[14] S.M. Ross, Introduction to Probability Models, ninth ed, Academic Press, Inc., Orlando, FL, USA, 2006.
[15] S. Wolfram, Theory and Applications of Cellular Automata, World Scientific, Singapore, 1986, ISBN: 9971-50-124-4 pbk.
[16] W. Li, N. Packard, The structure of the elementary cellular automata rule space, Complex Syst. 4 (1990) 281–297.
[17] N. Fatès, Guided tour of asynchronous cellular automata, J. Cell. Autom. 9 (5–6) (2014) 387–416.
[18] N. Fatès, M. Morvan, An experimental study of robustness to asynchronism for elementary cellular automata, Complex Syst. 16 (1) (2005).
[19] R. Alonso-Sanz, Phase transitions in an elementary probabilistic cellular automaton with memory, Physica A 347 (Supplement C) (2005) 383–401.
[20] S. Ninagawa, A. Adamatzky, R. Alonso-Sanz, Phase transition in elementary cellular automata with memory, Int. J. Bifurcation Chaos 24 (09) (2014) 1450116.
[21] N. Fatès, Asynchronism induces second-order phase transitions in elementary cellular automata, J. Cell. Autom. 4 (1) (2009) 21–38.
[22] N. Fatès, Diploid cellular automata: First experiments on the random mixtures of two elementary rules, in: Cellular Automata and Discrete Complex Systems - 23rd IFIP WG 1.5 International Workshop, AUTOMATA 2017, Milan, Italy, June 7-9, 2017, Proceedings, 2017, pp. 97–108.
[23] M. Gardner, Wheels, Life and Other Mathematical Amusements, Freeman, New York, 1983.
[24] H.J. Blok, B. Bergersen, Effect of boundary conditions on scaling in the "game of Life", Phys. Rev. E 55 (1997) 6249–6252.
[25] E.R. Berlekamp, J.H. Conway, R.K. Guy, Winning Ways for your Mathematical Plays, Vol. 2, Academic, 1982.
[26] H.J. Blok, B. Bergersen, Synchronous versus asynchronous updating in the "game of Life", Phys. Rev. E 59 (1999) 3876–3879.
[27] S.-Y. Huang, X.-W. Zou, Z.-J. Tan, Z.-Z. Jin, Network-induced nonequilibrium phase transition in the "game of Life", Phys. Rev. E 67 (2003) 026107.
[28] N. Fatès, Critical phenomena in cellular automata: perturbing the update, the transitions, the topology, Acta Phys. Polon. B 3 (2) (2010) 315–325.

[29] H. Bersini, V. Detours, Asynchrony induces stability in cellular automata based models, in: Proceedings of Artificial Life IV, MIT Press, 1994, pp. 382–387.
[30] B. Sethi, S. Roy, S. Das, Convergence of asynchronous cellular automata: Does size matter? J. Cell. Autom. 13 (5–6) (2018) 527–542.
[31] R. Cori, Y. Metivier, W. Zielonka, Asynchronous mappings and asynchronous cellular automata, Inform. and Comput. 106 (2) (1993) 159–202.
[32] B. Sethi, S. Roy, S. Das, Asynchronous cellular automata and pattern classification, Complexity 21 (S1) (2016) 370–386.
[33] P. Banda, J. Caughman, J. Pospichal, Configuration symmetry and performance upper bound of one-dimensional cellular automata for the leader election problem, J. Cell. Autom. 10 (1–2) (2015) 1–21.
[34] S. Roy, S. Das, Distributed mutual exclusion problem in cellular automata, J. Cell. Autom. 12 (6) (2017) 493–512.
[35] S. Roy, A. Ray, S. Das, A cellular automaton that solves distributed spanning tree problem, J. Comput. Sci. 26 (2018) 39–54.