BioSystems 186 (2019) 103976

Contents lists available at ScienceDirect

BioSystems

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

A statistical approach to the identification of diploid cellular automata based on incomplete observations

Witold Bołt[a,c,*], Aleksander Bołt[b], Barbara Wolnik[b], Jan M. Baetens[c], Bernard De Baets[c]

[a] Systems Research Institute, Polish Academy of Sciences, Newelska St. 6, 01-447 Warsaw, Poland
[b] Institute of Mathematics, Faculty of Mathematics, Physics and Informatics, University of Gdańsk, 80-308 Gdańsk, Poland
[c] KERMIT, Department of Data Analysis and Mathematical Modelling, Ghent University, B-9000 Ghent, Belgium

ARTICLE INFO

Keywords:
Stochastic cellular automata
Diploid cellular automata
Parameter estimation
System identification

ABSTRACT

In this paper, the identification problem of diploid cellular automata is considered, in which, based on a series of incomplete observations, the underlying cellular automaton rules and the states of missing cell states are to be uncovered. An algorithm for identifying the rule, based on a statistical parameter estimation method using a normal distribution approximation, is presented. In addition, an algorithm for filling the missing cell states is formulated. The accuracy of these methods is examined in a series of computational experiments.

1. Introduction

Cellular Automata (CAs) are commonly used modeling constructs for addressing a variety of practical and theoretical problems (Das, 2012). Yet, for that purpose one needs to understand the underlying mechanisms of the phenomenon at stake, and translate them into CA rules. This, however, hampers the use of CAs, since it is very hard to manually design such rules for most problems.

Many efforts have been made in the direction of developing automated methods for constructing CAs based on observed space-time diagrams. These include methods based on genetic algorithms (Bolt et al., 2018; Richards et al., 1990; Mitchell et al., 1996; Bäck et al., 2005; Sapin et al., 2003), genetic programming (Bandini et al., 2008; Maeda and Sakama, 2007; Andre et al., 1996), gene expression programming (Ferreira, 2001), other evolutionary algorithms (Kroczek and Zelinka, 2018), ant colony algorithms (Liu et al., 2008), machine learning approaches (Bull and Adamatzky, 2007; Gilpin, 2018), as well as direct construction algorithms (Adamatzky, 1994; Yang and Billings, 2000,2000; Sun et al., 2011). A review of the key methods is presented in Adamatzky (2012). Most recent research relates to deterministic CAs, with the notable exception of Billings and Yang (2003) where a Stochastic CA (SCA) is represented by a polynomial corrupted by noise, whose parameters are then discovered by a genetic algorithm. Despite the vast literature and the numerous attempts to solve the identification problem, it is still hard to outline an effective solution strategy that would work in the stochastic case and when observations of the SCA in question are incomplete, *i.e.* when not all cell states have been recorded.

In this paper, we focus on the identification of a class of Stochastic CAs (SCAs) called diploid CAs. Such SCAs recently gained a lot of attention in the research community (Fatès, 2017; Mendonça, 2017). The identification algorithm presented in this paper is an extension of the algorithm presented in Bołt et al. (2016), where the identification of α-asynchronous CAs in the case of incomplete observations was discussed. This extension allows for the identification of diploid CAs based on incomplete observations. The consideration of incomplete observations is motivated by the fact that in real-world problems it is practically impossible to capture the entire image of the phenomenon at stake. Indeed, due to technical limitations and the dynamical nature of the processes being observed, typically only some parts of the space-time history are available. The goal of the identification algorithm is to estimate the parameters of the underlying SCA and to estimate the missing states in the observations.

This paper is organized as follows. In Section 2 we present the key definitions. The identification problem and the description of the identification algorithm are presented in Section 3. Section 4 contains the results of our computational experiments. The paper is concluded by Section 5, where the results are summarized.

2. Preliminaries

In this paper, we consider 1D CAs whose N cells are arranged in a circular array. We focus on binary CAs with a symmetric neighborhood whose radius is denoted by r. A *configuration* of a given CA A is an

* Corresponding author.
E-mail address: witold.bolt@hope.art.pl (W. Bołt).

https://doi.org/10.1016/j.biosystems.2019.103976
Received 2 April 2018; Received in revised form 30 March 2019; Accepted 26 May 2019
Available online 29 May 2019

element $\boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1})$ of $\{0, 1\}^N$, and A is identified with its *global rule* $F: \{0, 1\}^N \to \{0, 1\}^N$, given by the formula $F(\boldsymbol{x}) = (x'_0, x'_1, \ldots, x'_{N-1})$, where:

$$x'_n = f(x_{n-r}, \ldots, x_{n-1}, x_n, x_{n+1}, \ldots, x_{n+r})$$

and all operations on the indices are performed modulo N. Here, the function $f: \{0, 1\}^{2r+1} \to \{0, 1\}$, called the *local rule*, is an update function, which may be deterministic or not. For the sake of readability, we enumerate the elements of $\{0, 1\}^{2r+1}$ as follows: $\boldsymbol{N}_0 = (0, \ldots, 0, 0)$, $\boldsymbol{N}_1 = (0, \ldots, 0, 1)$, …, $\boldsymbol{N}_{s-1} = (1, \ldots, 1, 0)$, $\boldsymbol{N}_s = (1, \ldots, 1, 1)$, where $s = 2^{2r+1} - 1$. Further, x_n^t will be used to denote the value of the nth cell after the tth application of F starting from the configuration $\boldsymbol{x}$.

CAs with a unit neighborhood radius and a deterministic local rule f are known as Elementary CAs (ECAs) (Wolfram, 1983). The local rule f of an ECA is a function of three variables, *i.e.* $f: \{0, 1\}^3 \to \{0, 1\}$. As the set $\{0, 1\}^3$ has only eight elements, *i.e.* $\boldsymbol{N}_0 = (0, 0, 0)$, $\boldsymbol{N}_1 = (0, 0, 1)$, …, $\boldsymbol{N}_7 = (1, 1, 1)$, the local rule f can be defined by collecting the values $\ell_i = f(\boldsymbol{N}_i) \in \{0, 1\}$, $i \in \{0, \ldots, 7\}$, in a lookup table (LUT) (see Table 1). Note that the order of the neighborhood configurations is fixed, so a given LUT can be stored using its last row.

The number $C = \sum_{i=0}^{7} f(\boldsymbol{N}_i)\, 2^i$ is called the rule number of the local rule f. We will write ECAC to refer to the ECA with rule number C (for example, ECA204 denotes the identity CA). The set of all 256 ECAs will be denoted by $\mathcal{E}$.

If the local rule of a CA is stochastic, we are dealing with an SCA. Here, we consider SCAs whose local rule can be expressed as:

$$x_n^{t+1} = X_{t,n}(x_{n-r}^t, \ldots, x_{n-1}^t, x_n^t, x_{n+1}^t, \ldots, x_{n+r}^t), \tag{1}$$

where $X_{t,n}(\boldsymbol{N}_i)$ are independent Bernoulli random variables satisfying:

$$\Pr(X_{t,n}(\boldsymbol{N}_i) = 1) = p_i, \tag{2}$$

i.e. the probability of turning the state of a cell into 1 in the next time step depends only on the states of the cells in its neighborhood and is independent of the time step t and the cell number i. Obviously, it then holds that:

$$\Pr(X_{t,n}(\boldsymbol{N}_i) = 0) = 1 - p_i, \tag{3}$$

which means that an SCA can be fully described by the sequence of probabilities $(p_0, p_1, \ldots, p_s)$, usually presented in a tabular form (pLUT). The general form of the pLUT of an SCA with $r = 1$ is given in Table 2. Although Table 2 does not look different from Table 1, its entries p_i are numbers belonging to [0,1], while each entry ℓ_i in Table 1 belongs to $\{0, 1\}$.

It is known that every SCA can be expressed as a stochastic mixture of a finite number of deterministic CAs (Bołt et al., 2015), *i.e.* for every SCA A, there exists a finite sequence of deterministic CAs $(A_1, \ldots, A_m)$ and a vector of probabilities $(\lambda_1, \ldots, \lambda_m)$ satisfying $\sum_{i=1}^{m} \lambda_i = 1$, such that A is equivalent to independently selecting A_i for every cell, at every time step, with probability λ_i. In this paper we focus on a special class of SCAs, the so-called *diploid* CAs, which can be expressed as stochastic mixtures consisting of only two deterministic CAs. Such SCAs have been studied earlier by several authors (e.g. Fatès, 2017; Mendonça, 2011). Note that a special class of diploid CAs is the class of α-asynchronous CAs (Fatès and Morvan, 2005), where one of the two deterministic CAs is the identity CA.

Definition 1 (*Diploid CA*). Let A_1 and A_2 be two different deterministic CAs with the same neighborhood radius r and with local rules f_1 and f_2, respectively. For any mixing rate $\lambda \in]0, 1]$, we define the diploid CA $(A_1, A_2)_\lambda$ as the SCA with the following probabilities in its pLUT:

$$p_i = \lambda f_1(\boldsymbol{N}_i) + (1 - \lambda) f_2(\boldsymbol{N}_i), \tag{4}$$

for any $i \in \{0, 1, \ldots, s\}$.

Table 1
General form of the LUT of the local rule of an ECA.

$\boldsymbol{N}_0$	$\boldsymbol{N}_1$	$\boldsymbol{N}_2$	$\boldsymbol{N}_3$	$\boldsymbol{N}_4$	$\boldsymbol{N}_5$	$\boldsymbol{N}_6$	$\boldsymbol{N}_7$
ℓ_0	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7

Table 2
General form of the pLUT of an SCA with unit radius.

$\boldsymbol{N}_0$	$\boldsymbol{N}_1$	$\boldsymbol{N}_2$	$\boldsymbol{N}_3$	$\boldsymbol{N}_4$	$\boldsymbol{N}_5$	$\boldsymbol{N}_6$	$\boldsymbol{N}_7$
p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7

$$p_i = \begin{cases} 0, & \text{if } f_1(\boldsymbol{N}_i) = f_2(\boldsymbol{N}_i) = 0, \\ \lambda, & \text{if } f_1(\boldsymbol{N}_i) = 1 \text{ and } f_2(\boldsymbol{N}_i) = 0, \\ 1 - \lambda, & \text{if } f_1(\boldsymbol{N}_i) = 0 \text{ and } f_2(\boldsymbol{N}_i) = 1, \\ 1, & \text{if } f_1(\boldsymbol{N}_i) = f_2(\boldsymbol{N}_i) = 1. \end{cases} \tag{5}$$

Note that if $\lambda' = 1 - \lambda$, then the diploid CA $(A_1, A_2)_{\lambda'}$ is identical to $(A_2, A_1)_\lambda$, allowing us to restrict to $\lambda \in]0, 0.5]$.

Example 2. Let A_1 be ECA57 and A_2 be ECA120. The general form of the pLUT of $(A_1, A_2)_\lambda$ is shown in Table 3. Some space-time diagrams of $(A_1, A_2)_\lambda$ evolved from the same initial configuration for different values of λ are shown in Fig. 1. In these space-time diagrams, we adopt the convention that the initial configuration is shown at the top and time increases downwards in the diagram. It can be seen that the space-time diagram becomes increasingly similar to the one of ECA57 as λ approaches one, while ECA120 is the most influential one for $\lambda < 0.5$.

In general, the decomposition of an SCA as a stochastic mixture of CAs is not unique (Bołt et al., 2015), yet the following proposition (Fatès, 2017) gives a full characterization of diploid CAs, as well as the conditions for the existence of a unique representation.

Proposition 1. *Let $(p_0, p_1, \ldots, p_s)$ be the pLUT of an SCA A. Then A is a diploid CA if and only if there exists a $\lambda \in]0, 0.5]$ such that $p_i \in \{0, \lambda, 1 - \lambda, 1\}$ for each $i \in \{0, 1, \ldots, s\}$, but $(p_0, p_1, \ldots, p_s) \notin \{0, 1\}^{s+1}$. Moreover, if $\lambda \neq 0.5$, then there exist a unique couple (A_1, A_2) such that $A = (A_1, A_2)_\lambda$. Otherwise, if $\lambda = 0.5$, then there exist 2^d such couples, with d being the number of p_i's equal to 0.5, for $i = 0, 1, \ldots, s$.*

3. Identification and gap filling

The goal of this section is to formally define the identification problem and formulate the identification algorithm incorporating a gap filling procedure.

3.1. Formulation of the identification problem

Our formulation is based on the notion of an observation of a space-time diagram, which is assumed to originate from some unknown diploid CA $(A_1, A_2)_\lambda$. Solving the identification problem requires finding both CAs A_1 and A_2 and obtaining a good estimation of λ. More formally, let $I_1, I_2, \ldots, I_M$ be $T \times N$ arrays with binary entries. Each array I_m, for $m \in \{1, 2, \ldots, M\}$, will be referred to as an observation. The set of all observations will be denoted by $\mathcal{I}$. We assume that each observation $I \in \mathcal{I}$ is a space-time diagram of the same diploid CA $(A_1, A_2)_\lambda$, *i.e.* the element $I(t, n)$ is the state of the nth cell at the tth time step.

We choose a small $\alpha \in]0, 1]$ and we take $1 - \alpha$ as a confidence level[1]. Based on the set of observations $\mathcal{I}$, we construct candidates for A_1 and A_2, and we estimate λ by building a confidence interval $[\lambda_L, \lambda_U]$.

[1] Note that α is not related to α-asynchronous CAs mentioned briefly in Section 2.

Table 3
The LUTs of ECAs 120 and 57 and the pLUT of the diploid (ECA120, ECA57)$_\lambda$.

	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7
ECA120	1	0	0	1	1	1	0	0
ECA57	0	0	0	1	1	1	1	0
diploid CA	λ	0	0	1	1	1	$1-\lambda$	0

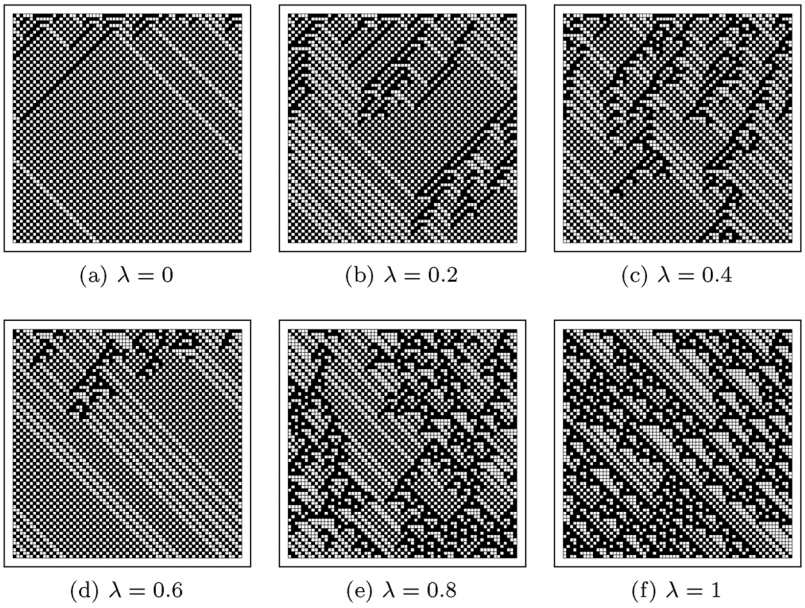


Fig. 1. Space-time diagrams of (ECA57, ECA120)$_\lambda$ for six different mixing rates λ, evolved from the same initial configuration.

We require that the probability that both CAs A_1 and A_2 are correctly identified and the true λ belongs to $[\lambda_L, \lambda_U]$ is at least $1-2\alpha$.

The above formulation is valid for so-called complete observations which are composed of only the symbols 0 and 1, meaning that all states have been (correctly) captured. To account for incomplete observations we extend the definition of an observation. An incomplete observation I is a $T \times N$ array composed of the symbols 0, 1 and ?, where ? represents an unknown state referred to as a gap. It is assumed that the first row of an observation I does not contain the symbol ?. The formulation of the identification problem given above remains valid in the case of incomplete observations. Yet, due to the introduction of gaps, we extend it with one additional requirement. After finding A_1, A_2 and the confidence interval for λ, we require to fill in the gaps with the most likely states in a way that does not conflict with the identified A_1 and A_2. In other words, we want to fill the gaps such that the obtained space-time diagram is a valid space-time diagram of $(A_1, A_2)_\lambda$. The latter requirement is very natural, yet the most obvious solution strategies fail to fulfill it.

Let I be an observation. We will write $I[t, n|r]$ to denote the vector $(I(t, n-r), ...,I(t, n+r))$ (assuming periodic boundary conditions). The value $I(t, n)$ can now be understood as a realization of the random variable $X_{t-1,n}(I[t-1, n|r])$. In this way, for any couple (t, n), we determine a *dependence region* $D(t, n)$ consisting of couples (t', n') that are linked with (t, n) by Eq. (1):

$$\begin{aligned} D(t, n) \;=\; & \{(t-1, n-r), ..., (t-1, n+r)\} \\ & \cup \{(t, n-2r), ..., (t, n+2r)\} \\ & \cup \{(t+1, n-r), ..., (t+1, n+r)\}. \end{aligned}$$

Assume that $I(t_1, n_1) = I(t_2, n_2) = ?$. We define a relation of *connection* between two gaps (t_1, n_1) and (t_2, n_2). We say that a gap at position (t_1, n_1) is *connected* with a gap at position (t_2, n_2) if $(t_2, n_2) \in D(t_1, n_1)$ (or, equivalently, $(t_1, n_1) \in D(t_2, n_2)$, due to the symmetry of the neighborhood). A gap is called *isolated* if it is not connected with any other gap.

We group the gaps in an observation into *clusters*. A cluster C is the smallest, nonempty subset of the set of gaps, so that if some gap belongs to C, also all the gaps connected with this gap belong to C. Note that a similar concept of cluster is considered in percolation theory (Broadbent and Hammersley, 1957). The clusters considered here, however, differ from the latter only by the definition of the connectivity.

This concept is illustrated in Fig. 2, where two clusters of gaps, for a neighborhood with unit radius, are shown. The first cluster consists of two gaps at cells (2, 3) and (3, 2), while the second one is formed by an isolated gap at cell (2, 8). The cells colored dark gray correspond to the union of the dependence regions of the the first cluster, while the cells colored light gray correspond to the second one. Note that periodic boundary conditions are used here.

In the design of the identification algorithm and the gap filling procedure we will restrict ourselves to the case of isolated gaps. The presented method can, however, be generalized to observations that contain larger clusters of gaps. The difficulty residing therein relates to the notational burden of a formal description of the solution strategy and the associated computational complexity of the required algorithm. This is due to the fact that for an effective gap filling algorithm it is necessary to consider all possible fillings of each of the clusters.

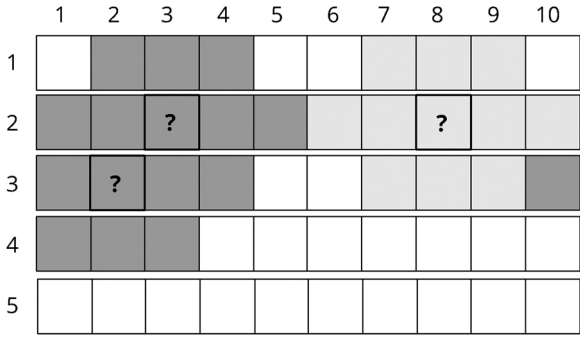


Fig. 2. Example of two clusters of gaps, the first cluster consisting of two cells: (2, 3) and (3, 2), while the second one is an isolated gap at cell (2, 8). In this example, a neighborhood with unit radius is used.

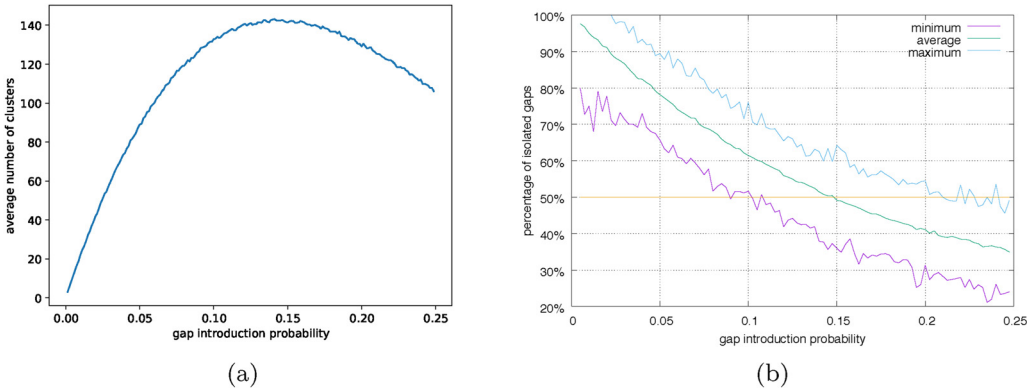


Fig. 3. Relationship between gap introduction probability and the number and size of the clusters: (a) average number of clusters vs. gap introduction probability; (b) minimum, average and maximum percentage of isolated gaps among all gaps.

Table 4
Minimum (min.), average (avg.), 95th-percentile (perc.), maximum (max.) and standard deviation (st. dev.) of the maximal relative error $E(A_1, A_2, \lambda)$ (Eq. (14)) for different values of λ.

	min.	avg.	95th-perc.	max.	st. dev.
$\lambda = 0.1$	0.89%	3.50%	10.21%	49.44%	3.46%
$\lambda = 0.2$	0.65%	2.42%	7.07%	38.20%	2.33%
$\lambda = 0.3$	0.45%	1.88%	5.49%	26.15%	1.79%
$\lambda = 0.4$	0.33%	1.51%	4.46%	20.58%	1.44%
all λs	0.33%	2.33%	6.68%	49.44%	2.49%

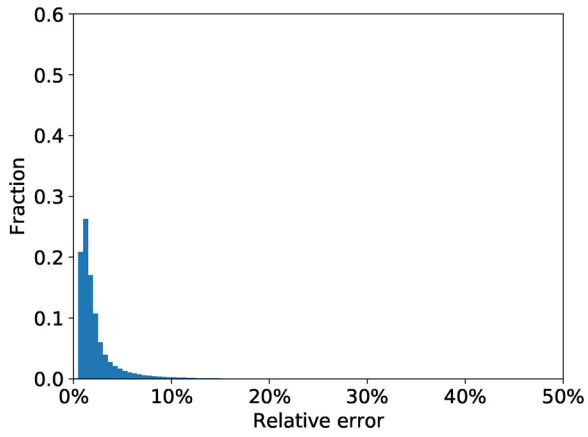


Fig. 4. Histogram of the maximal relative error $E(A_1, A_2, \lambda)$ (Eq. (14)) for all λ with bin size 0.5%.

Although it is possible to propose a divide and conquer strategy to tackle such problem, the complexity of a formal description is a serious bottleneck. Since considering only isolated gaps may seem very limiting, we verified how likely it is to obtain isolated gaps in observations in which the incompleteness is distributed randomly. We tested the relationship between the total number of gaps in an observation and the number of clusters and their sizes. For a fixed, complete 49×49 observation and a probability $p \in]0, 1]$, we randomly introduced gaps, with p being the cell-wise probability of turning the valid state into a gap. We will refer to p as gap introduction probability. In the obtained incomplete observation we calculated the number of obtained clusters, for a neighborhood with unit radius, and we measured the percentage of isolated gaps in the entire spectrum of observed gaps. We repeated this process 250 times for $p \in \{0.005, 0.01, ..., 0.995\}$ and we calculated the average for each p.

Fig. 3 depicts these results. We can see that for $p < 0.15$, on average, isolated gaps account for more than 50% of all of the gaps. In other words, when there is a significant, but not too large number of gaps in an observation, then most of them are isolated. Obviously, the obtained value of $p = 0.15$ depends on the choice of the neighborhood radius r and will likely be lower for larger radii. Yet, as the number of gaps is still reasonably low, we can expect to have many isolated gaps.

To sum up, we think it is justified to consider only isolated gaps in the identification and gap filling problems. Note that if most of the gaps are not isolated, most probably the number of gaps is quite large and thus the quality of the observation itself is very low and we should not expect much from any identification strategy.

3.2. Identification algorithm

Here, we propose an algorithm for solving the identification problem. For simplicity, a description of the algorithm is given in the case when there are no gaps. Following Proposition 1, it is obvious that it should be assumed that $\lambda \neq 0.5$, but to obtain the required confidence level, we additionally assume that λ is bounded between known bounds a and b, *i.e.* $0 < a \leq \lambda \leq b < 0.5$.

Based on a set of observations $\mathcal{I}$, we create frequency tables $L = (L_0, ..., L_s)$ and $K = (K_0, ..., K_s)$, where L_i denotes the number of occurrences of neighborhood configuration $\boldsymbol{N}_i$ in the observations $I \in \mathcal{I}$, where the last row of each observation is discarded. To build table K, we additionally check the state of the central cell in row $t + 1$ for each of the neighborhoods in row t, and we count the number of times it equals 1. The meaning of the numbers L and K is following. For every $i \in \{0, 1, ..., s\}$, L_i is the number of occurrences of the neighborhood configuration $\boldsymbol{N}_i$, while K_i is the number of cases in which the application of the unknown diploid CA to this neighborhood configuration resulted in state 1. Obviously, $L_i - K_i$ is the number of cases in which the outcome of the diploid CA's application to $\boldsymbol{N}_i$ was 0. We assume that the set of observations $\mathcal{I}$ is large enough to ensure that each neighborhood configuration was observed at least once (which is always possible if we have control over the initial configurations), hence $L_i > 0$ for every i. The following proposition is the basis of the identification algorithm presented in this section.

Proposition 2. *Assume that the observations in $\mathcal{I}$ are space-time diagrams of a diploid CA $(A_1, A_2)_\lambda$ and f_1 and f_2 are the local rules of A_1 and A_2, respectively. Then for any $i \in \{0, 1, ..., s\}$ the proportion $\widehat{p_i} = \frac{K_i}{L_i}$ is a random variable following a Bernoulli distribution with success probability p_i, where p_i is given by Eq. (5).*

The first step in the identification is to identify the deterministic CAs A_1 and A_2, *i.e.* to find their corresponding LUTs $(\ell_0^{(1)}, ..., \ell_s^{(1)})$ and $(\ell_0^{(2)}, ..., \ell_s^{(2)})$. For every $i \in \{0, ..., s\}$, we proceed as follows:

(a) if $K_i = 0$, then we put $\ell_i^{(1)} = \ell_i^{(2)} = 0$,
(b) if $K_i = L_i$, then we put $\ell_i^{(1)} = \ell_i^{(2)} = 1$,

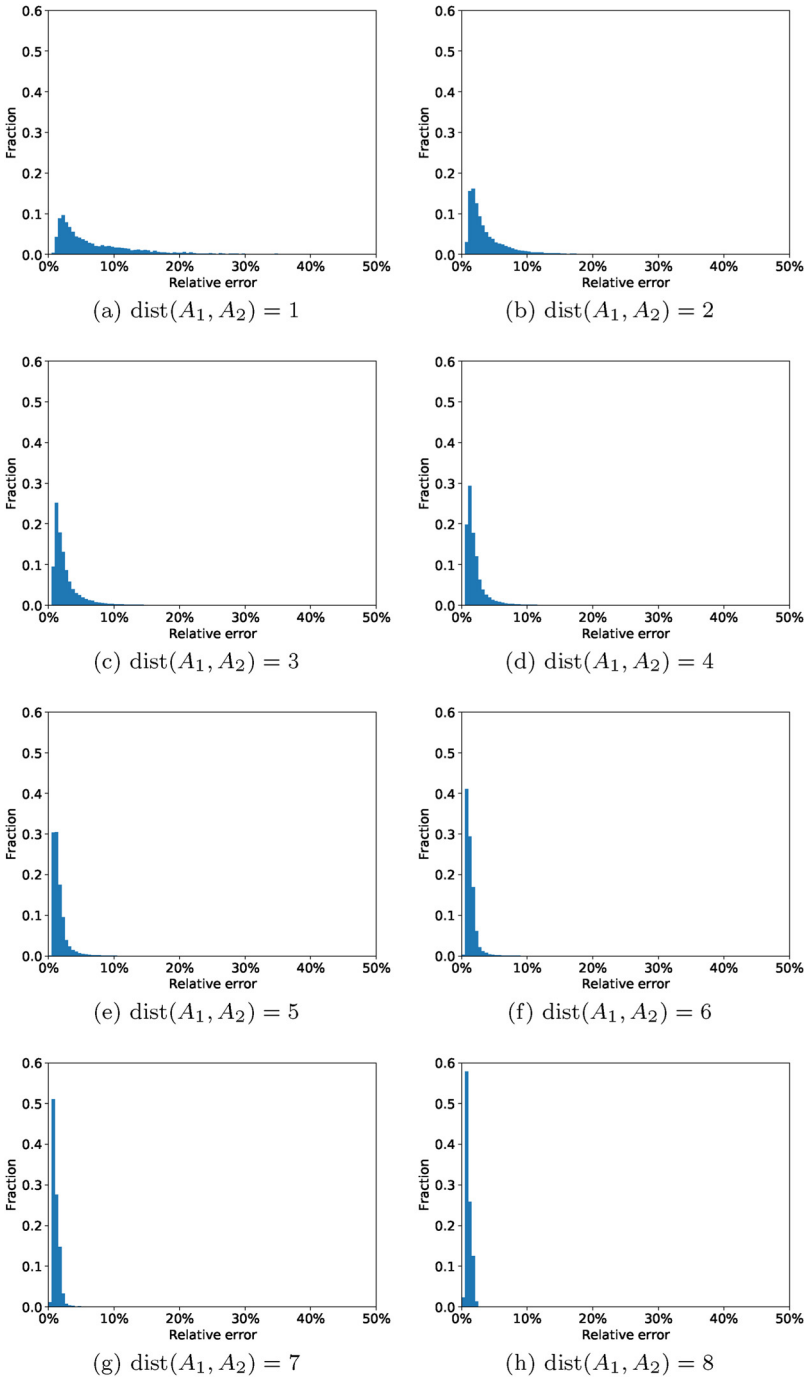


Fig. 5. Histogram of the maximal relative error $E(A_1, A_2, \lambda)$ (Eq. (14)) for all λ with bin size 0.5% grouped by the Hamming distance between the LUTs of A_1 and A_2.

(c) if $\frac{K_i}{L_i} < 0.5$, then we put $\ell_i^{(1)} = 1$ and $\ell_i^{(2)} = 0$,
(d) if $\frac{K_i}{L_i} > 0.5$, then we put $\ell_i^{(1)} = 0$ and $\ell_i^{(2)} = 1$.

Note that in case $K_i = 0$ (case (a)), we are not sure that both $\ell_i^{(1)}$ and $\ell_i^{(2)}$ are equal to zero, as it is possible that p_i is equal to λ or $1 - \lambda$, while there is no sample in $\mathcal{I}$ with the outcome 1. Fortunately, the probability of this happening equals $(1 - \lambda)^{L_i}$ or λ^{L_i}, and thus is less than $(1 - a)^{L_i}$. The same consideration applies when $K_i = L_i$ (case (b)). Hence, to achieve the desired confidence level, we will assume that $(1 - a)^{L_i} \leq \frac{\alpha}{2^{s+1}}$. In cases (c) and (d) the situation is a bit more complicated. If $\frac{K_i}{L_i} < 0.5$, then to verify whether p_i is really less than 0.5, we can perform a hypothesis test on proportions with alternative hypothesis H_1: $p_i < 0.5$. We use the normal approximation method and a left-tailed test. If the obtained p-value is less than $\frac{\alpha}{2^{s+1}}$, then we may claim that p_i is really less than 0.5. If $\frac{K_i}{L_i} > 0.5$, the alternative hypothesis is H_1: $p_i > 0.5$ and the test is right-tailed. This completes the procedure of finding A_1 and A_2. Given the above assumptions, the total probability of picking wrong CAs is less than α.

We now turn to the second step of the algorithm, *i.e.* the estimation of λ by constructing a relatively small confidence interval $[\lambda_L, \lambda_U]$ that contains the true (unknown) λ with high probability, assuming that the CAs A_1 and A_2 have been correctly identified. Let us note that if $0 < \frac{K_i}{L_i} < 0.5$, then we know that the diploid CA $(A_1, A_2)_\lambda$ acted as A_1 K_i times during L_i independent transitions, while if $0.5 < \frac{K_i}{L_i} < 1$, then this diploid CA acted as A_1 $L_i - K_i$ times within these L_i independent transitions. As a consequence, we get the following proposition.

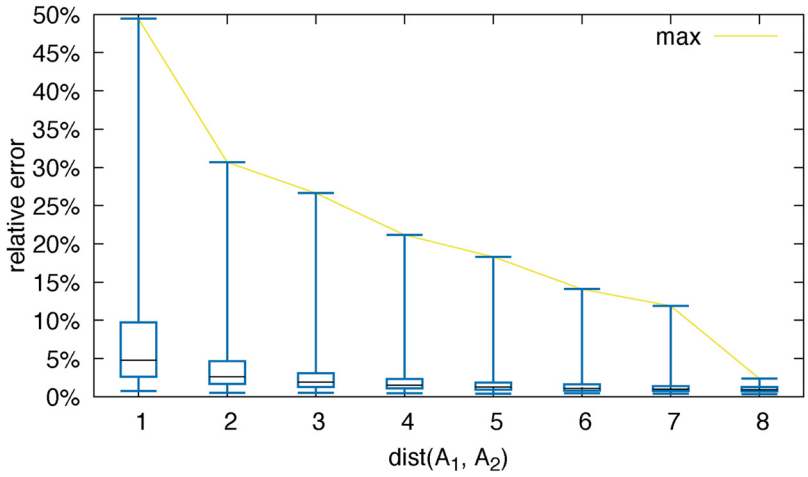


Fig. 6. Relation of the Hamming distance of the LUTs defining A_1 and A_2 to the cumulative relative error C_E.

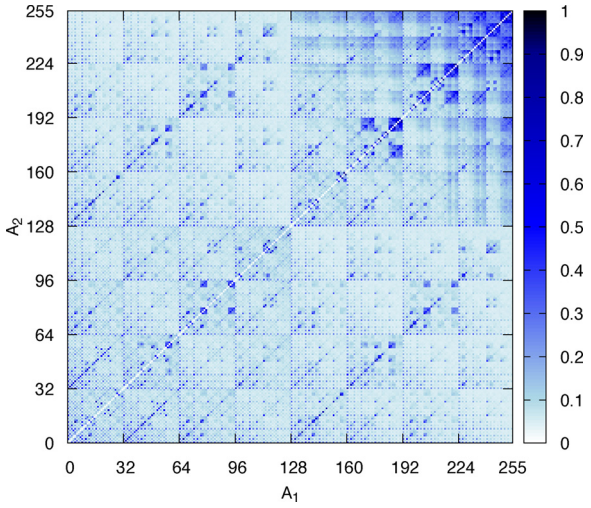


Fig. 7. Cumulative relative error $C_E(A_1, A_2)$ normalized with respect to the maximal cumulative error.

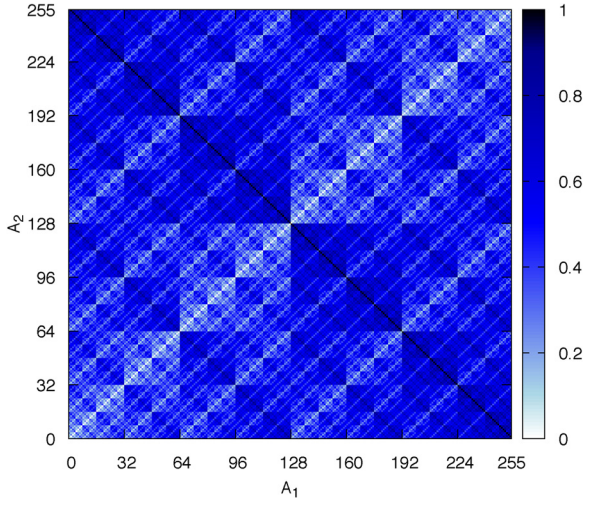


Fig. 8. Normalized Hamming distance between the LUTs of ECAs A_1 and A_2.

Table 5
Minimum (min.), average (avg.), 95th-percentile (perc.), maximum (max.) and standard deviation (st. dev.) of the obtained maximal distance from the confidence interval $D(A_1, A_2, \lambda)$ (Eq. (15)) for different values of λ.

	min.	avg.	95th-perc.	max.	st. dev.
$\lambda = 0.1$	0.0	0.0008	0.0024	0.0207	0.0010
$\lambda = 0.2$	0.0	0.0010	0.0034	0.0498	0.0014
$\lambda = 0.3$	0.0	0.0012	0.0040	0.0438	0.0017
$\lambda = 0.4$	0.0	0.0013	0.0044	0.0413	0.0018
all λs	0.0	0.0011	0.0036	0.0498	0.0015

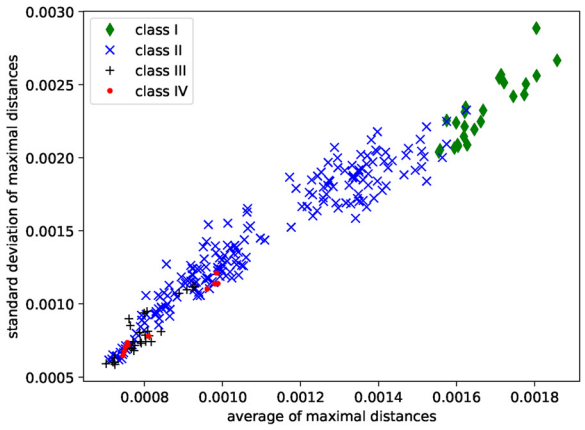


Fig. 9. Relation of the average and the standard deviation of the maximal distance to the confidence interval Δ. The shape and the color of points is assigned according to Wolfram's class of the corresponding ECA.

Table 6
Minimum (min.), 5th-percentile (perc.), average (avg.), maximum (max.) and standard deviation (st. dev.) of success rates obtained by the gap filling algorithm for different values of λ.

	min.	5th-perc.	avg.	max.	st. dev.
$\lambda = 0.1$	89.91%	93.36%	98.08%	100.00%	2.08%
$\lambda = 0.2$	79.84%	88.68%	96.45%	100.00%	3.63%
$\lambda = 0.3$	69.88%	84.05%	94.87%	100.00%	5.13%
$\lambda = 0.4$	59.89%	80.13%	93.53%	100.00%	6.53%
all λs	59.89%	85.30%	95.74%	100.00%	4.95%

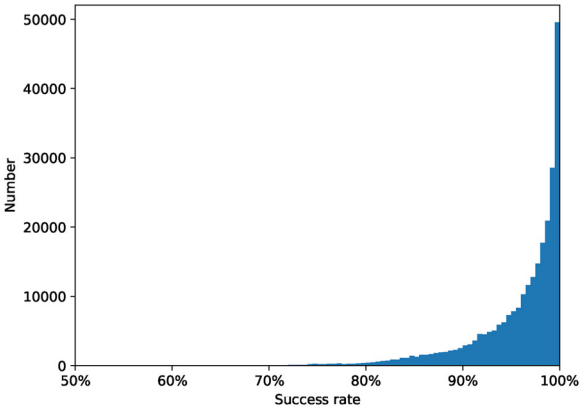


Fig. 10. Histogram of the success rates $\mathrm{SR}(A_1, A_2, \lambda)$ (Eq. (20)) for all the considered A_1, A_2 and λ.

Proposition 3. *Let* $\Gamma = \{i \in \{0, 1, ..., s\} | 0 < \frac{K_i}{L_i} < 0.5\}$ *and* $\Omega = \{i \in \{0, 1, ..., s\} | 0.5 < \frac{K_i}{L_i} < 1\}$. *Then the proportion*

$$\widehat{\lambda} = \frac{\sum_{i\in\Gamma} K_i + \sum_{i\in\Omega} (L_i - K_i)}{\sum_{i\in\Gamma} L_i + \sum_{i\in\Omega} L_i} \tag{6}$$

is a random variable following a Bernoulli distribution with success probability λ.

Following Brown et al. (2009) there are various methods for estimating the confidence interval for λ using $\widehat{\lambda}$. Here, we choose the normal distribution approximation, even though the authors of Brown et al. (2009) advise against it. This choice is motivated by the fact that this method has a reasonable accuracy in our case, while its implementation is straightforward. Assuming that $1 - \alpha$ is the chosen confidence level, then the following holds with probability $1 - \alpha$:

$$\lambda_L := \widehat{\lambda} - z_\alpha \sqrt{\frac{\widehat{\lambda}(1-\widehat{\lambda})}{L^*}} \le \lambda \le \widehat{\lambda} + z_\alpha \sqrt{\frac{\widehat{\lambda}(1-\widehat{\lambda})}{L^*}} =: \lambda_U\,, \tag{7}$$

where $L^* = \sum_{i\in\Gamma} L_i + \sum_{i\in\Omega} L_i$, and z_α is the argument at which the cumulative standard normal distribution function takes the value $1 - \frac{\alpha}{2}$. The above holds if L^* is large enough, which in our case means that $L^*\lambda$ and $L^*(1 - \lambda)$ are greater than five (Brown et al., 2009). Since λ is unknown, due to the assumption $\lambda \ge a$, we can impose a stronger condition $L^* > \frac{5}{a}$, which is easy to verify. With these assumptions, it holds that $\lambda \in [\lambda_L, \lambda_U]$ with probability $1 - \alpha$, assuming that A_1 and A_2 have been correctly identified. As already shown, the probability of picking A_1 and A_2 correctly is also $1 - \alpha$. Therefore, the total probability of correctly identifing A_1, A_2 and finding an interval in which λ is contained is at least $(1 - \alpha)^2 \ge 1 - 2\alpha$.

Note that $\lambda_U - \lambda_L \le \frac{z_\alpha}{\sqrt{L^*}}$ and for commonly used confidence levels it holds that $z_\alpha < 3$. Thus, if L^* is sufficiently large, we are sure that the interval $[\lambda_L, \lambda_U]$ narrows as the number of observed cells grows.

To use the above algorithm in the case of incomplete observations, only a simple modification is needed: we calculate L_i and K_i discarding those entries that contain the symbol "?".

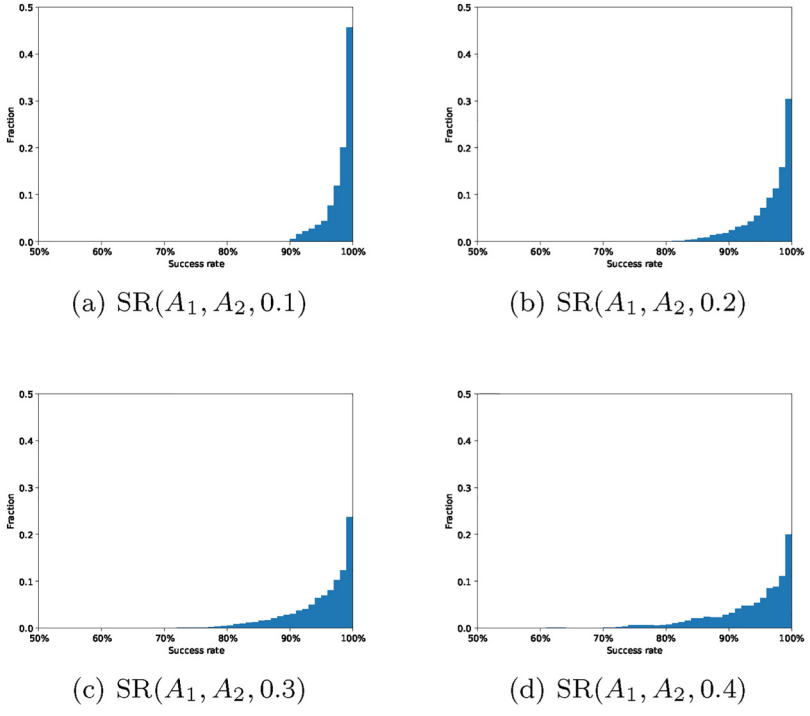
For further considerations we will need to have one numeric value as an estimate of λ instead of an interval. We take $\bar{\lambda} = \frac{1}{2}(\lambda_L + \lambda_U) = \widehat{\lambda}$ as such a point estimate. Note that this is the maximum maximum likelihood estimator for λ. The motivation of this choice is given in Section 3.3 below.

3.3. Gap filling algorithm

Using A_1, A_2 and $\bar{\lambda}$ found by the method described in Section 3.2, we propose an algorithm for estimating the missing states. Let $f_1, f_2 : \{0, 1\}^{2r+1} \to \{0, 1\}$ be the local rules of A_1, A_2, respectively. We consider the set of neighborhood configurations $C(f_1, f_2)$ consisting of neighborhoods on which f_1 and f_2 agree. In other words, $\boldsymbol{N}_i \in C(f_1, f_2)$ if $f_1(\boldsymbol{N}_i) = f_2(\boldsymbol{N}_i)$. Note that if $\boldsymbol{N}_i \in C(f_1, f_2)$, then the local rule of $(A_1, A_2)_\lambda$ is deterministic on the neighborhood $\boldsymbol{N}_i$.

The first step of our algorithm is to find gaps that result from the application of the diploid CA on the neighborhoods belonging to $C(f_1, f_2)$. To be precise, we are looking for (t, n) such that $I(t, n) = ?$ and $I[t - 1, n|r] \in C(f_1, f_2)$. In such case, we can set with certainty the value of $I(t, n)$ as $f_1(I[t - 1, n|r]) = f_2(I[t - 1, n|r])$.

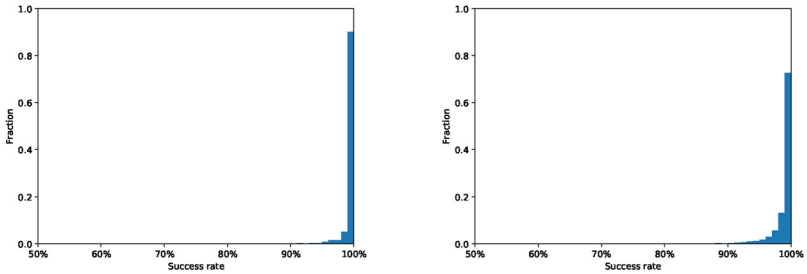
In the second step of our algorithm, we consider the case of neighborhoods that do not belong to $C(f_1, f_2)$. Let p_y be the probability



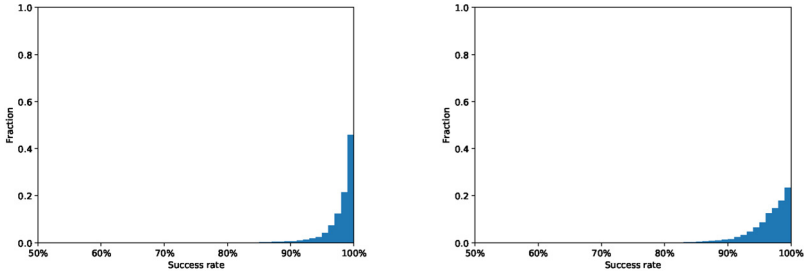
(a) $\mathrm{SR}(A_1, A_2, 0.1)$ (b) $\mathrm{SR}(A_1, A_2, 0.2)$

(c) $\mathrm{SR}(A_1, A_2, 0.3)$ (d) $\mathrm{SR}(A_1, A_2, 0.4)$

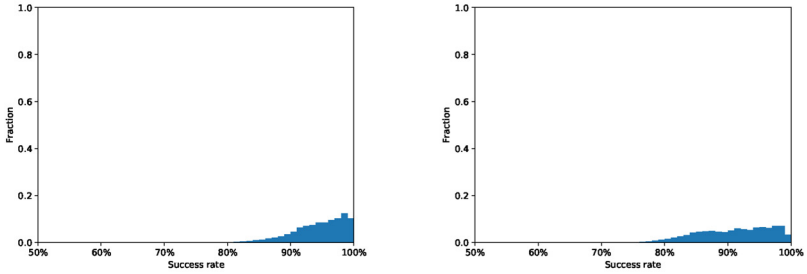
Fig. 11. Success rates obtained with the gap filling algorithm for all experiments grouped by λ.



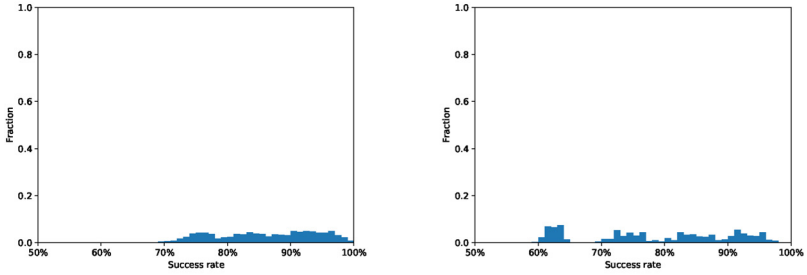
(a) $\mathrm{dist}(A_1, A_2) = 1$ (b) $\mathrm{dist}(A_1, A_2) = 2$



(c) $\mathrm{dist}(A_1, A_2) = 3$ (d) $\mathrm{dist}(A_1, A_2) = 4$



(e) $\mathrm{dist}(A_1, A_2) = 5$ (f) $\mathrm{dist}(A_1, A_2) = 6$



(g) $\mathrm{dist}(A_1, A_2) = 7$ (h) $\mathrm{dist}(A_1, A_2) = 8$

Fig. 12. Histogram of the success rates obtained with the gap filling algorithm grouped by the Hamming distance between the LUTs of A_1 and A_2.

of $I(t, n)$ being equal to $y \in \{0, 1\}$ in case the vector $I[t-1, n|r]$ is known. From the definition of a diploid $(A_1, A_2)_\lambda$, we have:

$$p_y = \Pr(I(t, n) = y) = \begin{cases} \lambda, & \text{if } f_1(I[t-1, n|r]) = y, \\ 1-\lambda, & \text{if } f_1(I[t-1, n|r]) = 1-y. \end{cases} \tag{8}$$

In order to fill the missing state $I(t, n)$, we examine the $(t+1)$th row of the observation. For $h \in \{-r, \ldots, r\}$, we consider the random event F_h meaning that starting from the configuration $I[t-1, n+h|r]$ an evolution of $(A_1, A_2)_\lambda$ leads to a state $I(t+1, n+h)$. From our assumption about isolated gaps, the value of $I(t+1, n+h)$ is known. Let us consider the probability $p_{h,y}$ of obtaining $I(t+1, n+h)$ given $I(t, n) = y$:

$$p_{h,y} = \Pr(F_h | I(t, n) = y). \tag{9}$$

From Eq. (4) we can easily find $p_{h,y}$. Indeed, if $I[t, n+h|r] \in C(f_1, f_2)$, then

$$p_{h,y} = \begin{cases} 0, & \text{if } f_1(I[n, m+h|r]) \neq I(t+1, n+h), \\ 1, & \text{if } f_1(I[n, m+h|r]) = I(t+1, n+h), \end{cases} \tag{10}$$

while if $I[t, n+h|r] \notin C(f_1, f_2)$, then

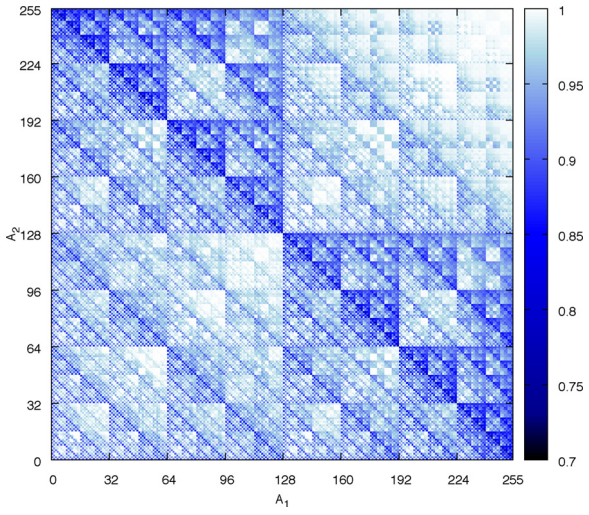


Fig. 13. Normalized cumulative success rate $C_{SR}(A_1, A_2)$.

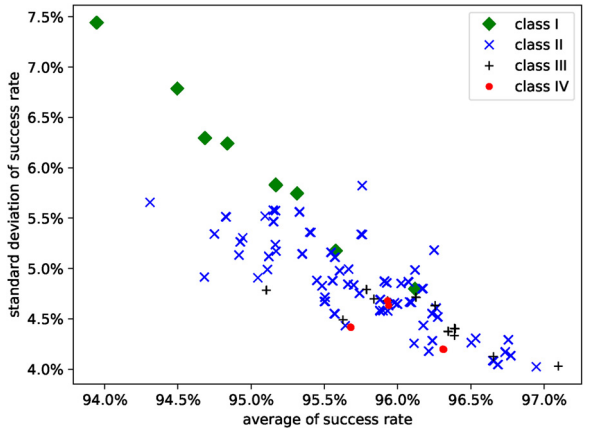


Fig. 14. Relation of the average and the standard deviation of the success rates obtained with the gap filling algorithm. The shape and the color of points is assigned according to Wolfram's class of the corresponding ECA.

$$p_{h,y} = \begin{cases} \lambda, & \text{if } f_1(I[n, m+h|r]) = I(t+1, n+h), \\ 1-\lambda, & \text{if } f_1(I[n, m+h|r]) \neq I(t+1, n+h). \end{cases} \tag{11}$$

Let $F = \bigcap_{h=-r}^{r} F_h$. Consider the probability $\Pr(I(t, n) = y|F)$ of a missing value equal to y given that the evolution from the $(t-1)$th row to the $(t+1)$th row goes as in observation I. The idea of our algorithm is to choose the $y \in \{0, 1\}$ that maximizes this probability. We calculate $\Pr(I(t, n) = 0|F)$ and $\Pr(I(t, n) = 1|F)$ using the formulas obtained in Bołt et al. (2016). For the sake of completeness, we recall these formulas below. From Bayes' rule we know that:

$$\Pr(I(t, n) = y|F) = \frac{p_y \Pr(F|I(t, n) = y)}{\Pr(F)}.$$

For $h_1 \neq h_2$, the events F_{h_1}, F_{h_2} are independent given that $I[t, n]$ is fixed, so from the total probability theorem it follows:

$$\begin{aligned} \Pr(F) &= \sum_{y=0}^{1} \Pr(I(t, n) = y) \Pr(F|I(t, n) = y) = \sum_{y=0}^{1} p_y \Pr(F|I(t, n) = y) \\ &= \sum_{y=0}^{1} p_y \prod_{h=-r}^{r} \Pr(F_h|I(t, n) = y) = \sum_{y=0}^{1} p_y \prod_{h=-r}^{r} p_{h,y}. \end{aligned} \tag{12}$$

Hence,

$$\Pr(I(t, n) = y|F) = \frac{p_y \prod_{h=-r}^{r} p_{h,y}}{\sum_{y=0}^{1} p_y \prod_{h=-r}^{r} p_{h,y}}. \tag{13}$$

Note that in order to maximize the above probability, we only need to choose the maximum of the numerators $\mathrm{NUM}_0 = p_0 \prod_{h=-r}^{r} p_{h,0}$ and $\mathrm{NUM}_1 = p_1 \prod_{h=-r}^{r} p_{h,1}$. Doing so, we can set $I(t, n) = y$ accordingly (if these two numbers are equal, we choose $I(t, n)$ randomly). Unfortunately, we do not know the exact value of λ, but only its estimate. However, according to Eqs. (8), (10) and (11), each of the numerators equals zero or is of the form $\lambda^j(1-\lambda)^k$, for some natural numbers j and k. If one of them is zero, then we are sure that the other is greater than zero. Thus, it remains to consider the case when both numerators are positive. So let $\mathrm{NUM}_0 = \lambda^{j_0}(1-\lambda)^{k_0}$ and $\mathrm{NUM}_1 = \lambda^{j_1}(1-\lambda)^{k_1}$. The following cases are possible:

1. $j_0 > j_1$ and $k_0 \geq k_1$ (or $j_0 \geq j_1$ and $k_0 > k_1$): $\mathrm{NUM}_0 < \mathrm{NUM}_1$ regardless of λ.
2. $j_0 < j_1$ and $k_0 \leq k_1$ (or $j_0 \leq j_1$ and $k_0 < k_1$): $\mathrm{NUM}_0 > \mathrm{NUM}_1$ regardless of λ.
3. $j_0 = j_1$ and $k_0 = k_1$: $\mathrm{NUM}_0 = \mathrm{NUM}_1$ regardless of λ.
4. $j_0 > j_1$ and $k_0 < k_1$: the result depends on λ.
5. $j_0 < j_1$ and $k_0 > k_1$: the result depends on λ.

Let us note that in case 4 (case 5 is analogous), the inequality $\mathrm{NUM}_0 > \mathrm{NUM}_1$ is equivalent to:

$$\lambda^{j_0 - j_1} > (1-\lambda)^{k_1 - k_0},$$

and both exponents are positive. It is easy to see that the inequality is satisfied for $\lambda \in]\lambda', 1[$, where λ' is the unique solution of $\lambda^{j_0-j_1} = (1-\lambda)^{k_1-k_0}$ in the interval $]0, 1[$. It means that if $\lambda \in]\lambda', 1[$, then $\mathrm{NUM}_0 > \mathrm{NUM}_1$, but if $\lambda \in]0, \lambda'[$, then $\mathrm{NUM}_0 < \mathrm{NUM}_1$. Although we do not know the exact value of λ, from the identification algorithm, we can assume that $\lambda \in [\lambda_L, \lambda_U]$. Since the interval $[\lambda_L, \lambda_U]$ is very small, in most cases it will be entirely contained in $]0, \lambda'[$ or in $]\lambda', 1[$. However, if $[\lambda_L, \lambda_U]$ overlaps with both $]0, \lambda'[$ and $]\lambda', 1[$, then the one with a greater common part will include also the center of $[\lambda_L, \lambda_U]$ which equals $\bar{\lambda}$. This consideration provides a heuristic argument for the use of a known value $\bar{\lambda}$ as point estimate, when the exact value of λ is not known. Using $\bar{\lambda}$ we can identify the largest numerator in cases 4 and 5 and thus we can always estimate the state.

Although we do not provide a formal proof, it holds that the outcome of this procedure of gap filling will never produce a neighborhood belonging to $C(f_1, f_2)$ that would be inconsistent with the output of f_1 and f_2. In other words, it is guaranteed that the complete space-time diagrams obtained with our method are valid space-time diagrams of the identified diploid CA.

4. Results

In this section, we present the results of our computational experiments to illustrate the accuracy of the algorithm described in Section 3. Firstly, we concentrate on the case of complete observations in order to verify the quality of the estimation of the parameters defining a diploid CA. Secondly, we consider incomplete observations with isolated gaps, and we measure the success rate of filling the gaps.

4.1. Verification of the identification algorithm

This experiment concerns the identification of diploid CAs consisting of ECAs based on complete observations. More formally, we considered diploid CAs $(A_1, A_2)_\lambda$, with $A_1, A_2 \in \mathcal{E}$ and $\lambda \in \{0.1, \ldots, 0.9\}$, with the exception of 0.5. Since $(A_2, A_1)_\lambda$ is identical to $(A_1, A_2)_{1-\lambda}$, only the diploid CAs based on ECAs for $\lambda \in \{0.1, 0.2, \ldots, 0.4\}$ need to be examined. As $A_1 \neq A_2$, a total of $256 \times 255 \times 4 = 249900$ diploid CAs were considered. The same set of 100 random initial configurations was used for all considered cases. Each of the initial configurations contained 49

cells. Using these initial configurations, 100 observations, each containing 49 times steps, were generated for each $(A_1, A_2)_\lambda$. The identification algorithm was executed for these observations. The process of constructing the observation set and identifying the CAs was repeated 50 times for each of the considered diploid CAs. Consequently, for each diploid CA $(A_1, A_2)_\lambda$, 50 pairs of candidate CAs $(A_1^{(j)}, A_2^{(j)})$ and 50 confidence intervals $[\lambda_L^{(j)}, \lambda_U^{(j)}]$ for $j \in \{1, 2, ..., 50\}$ were obtained. In each run of the algorithm for each diploid CA, the obtained candidate CAs $(A_1^{(j)}, A_2^{(j)})$ were perfectly matching the ones defining the diploid CA in question. More formally, for every $(A_1, A_2)_\lambda$ it turned out that $A_1 = A_1^{(j)}$ and $A_2 = A_2^{(j)}$ for every j, meaning that the first step of the identification algorithm always resulted in a correct identification of the CAs making up the diploid CAs.

To verify the results of the second step of the algorithm, two error measures were used: the maximal relative error and the maximal distance to the confidence interval. Letting $\widehat{\lambda}^{(j)} = \frac{\lambda_L^{(j)} + \lambda_U^{(j)}}{2}$, the maximal relative error is defined as:

$$E(A_1, A_2, \lambda) = \max_{j=1,...,50} \frac{|\widehat{\lambda}^{(j)} - \lambda|}{\lambda} \times 100\%, \tag{14}$$

while the maximal distance to the confidence interval is defined as:

$$D(A_1, A_2, \lambda) = \max_{j=1,...,50} \mathrm{d}(\lambda, [\lambda_L^{(j)}, \lambda_U^{(j)}]), \tag{15}$$

where:

$$\mathrm{d}(x, [a, b]) = \begin{cases} 0, & \text{if } x \in [a, b], \\ a - x, & \text{if } x < a, \\ x - b, & \text{if } x > b. \end{cases} \tag{16}$$

A statistical summary containing the minimum, average, 95th-percentile, maximum and the standard deviation of the maximal relative error E is given in Table 4. The maximal error (49.44%) may seem high, but as the 95th-percentile values show, in the vast majority of cases the errors are significantly lower.

In Fig. 4 the overall histogram of the maximal relative error E from all data points is shown, while in Fig. 5 we show the results grouped according to the Hamming distance (dist) between the LUTs of the ECAs A_1 and A_2. As can be seen, the distributions of the relative error for each of the distances are quite different from each other. Note that each of the histograms has been normalized with respect to the maximal number of occurrences to account for a different number of instances in the classes.

To further analyze the obtained results, we define the cumulative relative error $C_E(A_1, A_2)$ as:

$$C_E(A_1, A_2) = \sum_{\lambda=0.1,...,0.4} E(A_1, A_2, \lambda) + \sum_{\lambda=0.1,...,0.4} E(A_2, A_1, \lambda), \tag{17}$$

which for each pair of ECAs combines the results for the different values of λ. We assume $C_E(A, A) = 0$ for any ECA A. Obviously, it holds that $C_E(A_1, A_2) = C_E(A_2, A_1)$.

As already shown in Fig. 5, the distance between the LUTs of ECAs A_1 and A_2 greatly influences the quality of estimation. We grouped the values of C_E according to the Hamming distance between the LUTs of A_1 and A_2 (Fig. 6) to better understand this influence. As expected, the closer the ECAs are to each other in terms of their LUTs, the higher the value of C_E. This can be understood by analyzing Eq. (5). The number of positions at which the LUTs of A_1 and A_2 differ determines the number of neighborhoods on which the diploid CA acts non-deterministically, and thus produces transitions that are useful for estimating. This means that CAs that are close to each other will likely produce less samples that can be used for the estimation of λ.

Although Fig. 6 suggests a strong impact of the Hamming distance of the LUTs defining A_1 and A_2 on the estimation error, there are also other factors contributing to this error. To illustrate this the values of the cumulative relative error C_E, normalized with respect to the maximal cumulative error, are shown in Fig. 7. As can be inferred from this graph, there are significant differences between the values of C_E in different areas of the ECA space. Moreover, many symmetries can be observed.

In Fig. 8 we illustrate the Hamming distance between ECAs A_1 and A_2 in the same layout as the cumulative error C_E shown in Fig. 7. Comparing the graphs in Figs. 7 and 8 we see many similarities, though there are also differences. As mentioned earlier, the Hamming distance between ECAs A_1 and A_2 corresponds to the number of entries in the pLUT of the diploid CA that are neither 0 nor 1. Such entries correspond to neighborhood configurations on which the diploid CA is non-deterministic. Intuitively, we expect that if the number of such entries in the pLUT increases, more non-deterministic behavior should occur in the evolution. Yet, for this to occur, the corresponding neighborhood configurations need to appear in the space-time diagram, and this depends on the initial configuration and on the dynamical properties of the ECAs A_1 and A_2. For instance, consider a diploid CA made up by ECA184 (so-called traffic rule) and ECA232 (so-called majority rule). The Hamming distance between the LUTs of ECA184 and ECA232 is 2. Yet, this particular diploid CA is known to be a stochastic solution of the density classification problem (Fatès, 2013), and thus it evolves to a homogeneous configuration with all 0s or all 1s. This means that the two neighborhood configurations on which this diploid CA is non-deterministic quickly vanish. The study of the dynamical properties of diploid CAs is ongoing (Fatès, 2017), so it is not yet possible to give a full characterization of their dynamical properties in this paper. Such a characterization will allow to fully understand the differences in accuracy of the identification algorithm between different diploid CAs.

We now focus our analysis on the maximal distance from the confidence interval $D(A_1, A_2, \lambda)$ (Table 5).

In general, the values of D are low, which shows that in most cases the real λ either belongs to the confidence interval or is very close to it. This shows a high accuracy in the estimation of λ, irrespective of λ. For that reason we concentrate our analysis on the cumulative maximal distance to the confidence interval:

$$C_D(A_1, A_2) = \sum_{\lambda=0.1,...,0.4} D(A_1, A_2, \lambda) + \sum_{\lambda=0.1,...,0.4} D(A_2, A_1, \lambda). \tag{18}$$

These values were then grouped, for each ECA A, as:

$$\Delta(A) = \left\{ C_D(A, A_2) | A_2 \in \mathcal{E} \setminus \{A\} \right\}. \tag{19}$$

In Fig. 9, the relation between the average and the standard deviation of $\Delta(A)$ is shown for each ECA A. Each point in this plot corresponds to a specific ECA. The shape and the color of each of the points are assigned according to Wolfram's classification scheme (Wolfram, 1983), where Class I corresponds to simple dynamics resulting in homogeneous configurations, Class II — periodic dynamics, Class III — chaotic/random dynamics and Class IV — complex dynamics. As can be seen, there is a strong correlation between the Wolfram class and $\Delta(A)$. In general, the accuracy of the estimation of λ grows with the growing complexity of the ECA in question. This is due to the fact that complex ECAs generate more diversified observations containing a lot of occurrences of all of the neighborhood configurations, which leads to greater accuracy of estimation. On the other hand, relatively simple ECAs often evolve towards homogeneous or close-to-homogeneous configurations, and the number of useful samples for the estimation is lower.

4.2. Verification of the gap filling algorithm

The goal of the second experiment is to verify the correctness of the results produced by the gap filling algorithm. We used a similar experimental setup as in the previous experiment. We examined diploid CAs constructed from each couple of ECAs for $\lambda \in \{0.1, 0.2, 0.3, 0.4\}$. For every diploid CA we used 100 observations of size 49×49. For every observation we randomly introduced isolated gaps for 5% of the cells.

For every diploid CA and set of observations, we repeated the identification and gap filling processes 50 times.

Our main point of interest is the success of the gap filling algorithm. For every observation we calculated the success rate as the number of correctly filled gaps divided by total number of gaps in this observation. More precisely, let $\mathcal{I}_i = \{I_{1,i}, ..., I_{100,i}\}$ be a set of incomplete observations in the ith repetition of the experiment for a given diploid CA $(A_1, A_2)_\lambda$. Let $\text{gaps}(I_{n,i})$ denote the number of gaps in observation $I_{n,i}$ and $\text{success}(I_n, i)$ the number of correctly filled gaps. The success rate $\text{SR}(A_1, A_2, \lambda)$ for a diploid CA $(A_1, A_2)_\lambda$ is defined as:

$$\text{SR}(A_1, A_2, \lambda) = \frac{\sum_{i=1}^{50}\sum_{n=1}^{100}\text{success}(I_{n,i})}{\sum_{i=1}^{50}\sum_{n=1}^{100}\text{gaps}(I_{n,i})} \times 100\%\,. \tag{20}$$

A statistical summary of the obtained success rates across all the considered diploid CAs is given in Table 6. As can be inferred from this table, the total average of the success rate was more than 95%, meaning that on overage more than 95% of gaps were correctly filled. The minimum result is much lower (58%), but still even the lowest success rate is higher than 50%, meaning that most of the gaps were successfully filled. Moreover, when we look at the 5th-percentile values, we see that in general the success rate was very high. To further illustrate this, a histogram of all success rates obtained in the experiment is presented in Fig. 10. From this graph we can clearly see the concentration of the data at very high values. As Table 6 suggests, there are strong differences between the results obtained for different values of λ. To better understand this relationship we grouped all success rates by λ:

$$\text{SR}(\lambda) = \{\text{SR}(A_1, A_2, \lambda) | A_1, A_2 \in \mathcal{E}\}. \tag{21}$$

We expect that the success rates will be better for diploids with smaller λ, since as λ goes to 0.5, the diploid CA goes to a purely random behavior on some of the neighborhoods and thus filling gaps on such neighborhoods becomes very hard. Our experiments confirm this expectation. We created four histograms of the values $\text{SR}(\lambda)$ (see Fig. 11). As can be seen, the success rates are significantly better for diploid CAs with $\lambda = 0.1$ as compared to the case $\lambda = 0.4$.

Further, we grouped the success rates from all of experiments by the Hamming distance between the LUTs of A_1 and A_2 used to define the diploid CAs. From Fig. 12 it is easily seen that our gap filling algorithm works better for diploids created from ECAs that are similar to each other, *i.e.* ECAs that have a small number of neighborhoods on which they differ. This is due to the fact that for such pairs, the number of nondeterministic transitions is lower, and thus estimation is easier.

To better understand the impact of the choice of specific ECAs for a diploid CA, we define the cumulative success rate for a pair of ECAs A_1, A_2 as:

$$C_{\text{SR}}(A_1, A_2) = \sum_{\lambda=0.1,\ldots,0.4} \text{SR}(A_1, A_2, \lambda) + \sum_{\lambda=0.1,\ldots,0.4} \text{SR}(A_2, A_1, \lambda)\,.$$

The obtained results in a normalized form are visualized in Fig. 13. As can be inferred from this picture, there are significant differences among the ECAs and moreover many symmetries are present. These differences might be influenced by the same factors as the cumulative error of estimation C_E discussed earlier in this section.

Finally, we grouped success rate by specific ECA A, as:

$$\Delta\text{SR}(A) = \left\{\text{SR}(A, A_2) | A_2 \in \mathcal{E}\setminus\{A\}\right\}. \tag{22}$$

Fig. 14 shows the relation between the average and the standard deviation of $\Delta\text{SR}(A)$. Each point on the graph corresponds to one ECA and is colored according to the corresponding Wolfram class. As can be seen, the accuracy of the gap filling algorithm grows with the complexity of the ECA in question, which can be attributed to the fact that complex ECAs result in a more diverse behavior, and thus more data that is potentially helpful for the gap filling procedure.

5. Summary

In this paper the identification of a diploid CA from given, potentially incomplete, space-time diagrams has been discussed. An identification algorithm has been described in detail. Computational experiments have shown that the algorithm is very effective. The deterministic CAs constituting the analyzed diploids CAs were always correctly identified and the accuracy of the estimation of λ was very high with an average relative error of 2.33%. In case of incomplete observations, the gap filling algorithm was able to estimate the correct states of cells that have not been observed with an average success rate of more than 95%, allowing to uncover full space-time diagrams with very high accuracy. Following the line of research of this paper, in future we intend to extend the presented algorithm to wider classes of SCAs.

References

Adamatzky, A., 1994. Identification of Cellular Automata. Taylor & Francis Group, London, United Kingdom. https://doi.org/10.1201/9781315274355.

Adamatzky, A., 2012. Identification of cellular automata. In: Meyers, R.A. (Ed.), Computational Complexity: Theory, Techniques, and Applications. Springer New York, New York, NY, USA, pp. 1564–1575. https://doi.org/10.1007/978-1-4614-1800-9_100.

Andre, D., Bennett III, F.H., Koza, J.R., 1996. Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. In: Proceedings of the 1st Annual Conference on Genetic Programming. MIT Press Cambridge MA, USA. pp. 3–11.

Bäck, T., Breukelaar, R., Willmes, L., 2005. Inverse design of cellular automata by genetic algorithms: An unconventional programming paradigm. In: Banâtre, P., Fradet, P., Giavitto, J.-L., Michel, O. (Eds.), Unconventional Programming Paradigms, volume 3566, of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 161–172. https://doi.org/10.1007/11527800_13.

Bandini, S., Manzoni, S., Vanneschi, L., 2008. Evolving robust cellular automata rules with genetic programming. In: Adamatzky, A., Alonso-Sanz, R., Lawniczak, A.T., Martínez, G.J., Morita, K., Worsch, T. (Eds.), Automata. Luniver Press, Frome, UK, pp. 542–556.

Billings, S.A., Yang, Y., 2003. Identification of probabilistic cellular automata. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 33, 225–236. https://doi.org/10.1109/TSMCB.2003.810437.

Bołt, W., Baetens, J.M., De Baets, B., 2015. On the decomposition of stochastic cellular automata. J. Comput. Sci. 11, 245–257. https://doi.org/10.1016/j.jocs.2015.09.004.

Bołt, W., Wolnik, B., Baetens, J.M., De Baets, B., 2016. On the identification of α-asynchronous cellular automata in the case of partial observations with spatially separated gaps. In: de Tré, G., Grzegorzewski, P., Kacprzyk, J., Owsiński, J.W., Penczek, W., Zadrożny, S. (Eds.), Challenging Problems and Solutions in Intelligent Systems. Springer International Publishing, Cham, Switzerland, pp. 23–36. https://doi.org/10.1007/978-3-319-30165-5_2.

Bolt, W., Baetens, J.M., De Baets, B., 2018. Identification of cellular automata based on incomplete observations with bounded time gaps. IEEE Trans. Cybernet. 1–14. https://doi.org/10.1109/TCYB.2018.2875266.

Broadbent, S.R., Hammersley, J.M., 1957. Percolation processes: I. crystals and mazes. Math. Proc. Cambridge Philosophical Soc. 53, 629–641. https://doi.org/10.1017/S0305004100032680.

Brown, L.D., Cai, T.T., DasGupta, A., 2001. Interval estimation for a binomial proportion. Stat. Sci. 16, 101–133. https://doi.org/10.1214/ss/1009213286.

Bull, L., Adamatzky, A., 2007. A learning classifier system approach to the identification of cellular automata. J. Cell. Automata 2, 21–38.

Das, D., 2012. A survey on cellular automata and its applications. In: Global Trends in Computing and Communication Systems. Springer. pp. 753–762. https://doi.org/10.1007/978-3-642-29219-4_84.

Fatès, N., Morvan, M., 2005. An experimental study of robustness to asynchronism for elementary cellular automata. Complex Systems 16, 1–27.

Fatès, N., 2013. Stochastic cellular automata solutions to the density classification problem - when randomness helps computing. Theory Comput. Syst. 53, 223–242. https://doi.org/10.1007/s00224-012-9386-3.

Fatès, N., 2017. Diploid cellular automata: First experiments on the random mixtures of two elementary rules. In: Dennunzio, A., Formenti, E., Manzoni, L., Porreca, A.E. (Eds.), Cellular Automata and Discrete Complex Systems: 23rd IFIP WG 1.5 International Workshop, AUTOMATA 2017, Milan, Italy, June 7–9, 2017. Springer International Publishing, Cham, Switzerland, pp. 97–108. https://doi.org/10.1007/978-3-319-58631-1_8.

Ferreira, C., 2001. Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst. 13, 87–129.

W. Gilpin, Cellular automata as convolutional neural networks, arXiv e-prints 2018, 1809.02942.

Kroczek, L., Zelinka, I., 2018. Investigation on cellular automaton rule estimation. J. Cell. Automata 13, 307–323.

Liu, X., Li, X., Liu, L., He, J., Ai, B., 2008. A bottom-up approach to discover transition rules of cellular automata using ant intelligence. Int. J. Geographical Inform. Sci. 22,

1247–1269. https://doi.org/10.1080/13658810701757510.
Maeda, K., Sakama, C., 2007. Identifying cellular automata rules. J. Cell. Automata 2, 1–20.
Mendonça, J., de Oliveira, M., 2011. An extinction-survival-type phase transition in the probabilistic cellular automaton $p182$–$q200$. J. Phys. A: Math. Theoretical 44https://doi.org/10.1088/1751-8113/44/15/155001. Article ID 155001.
Mendonça, J.R.G., 2017. Critical behaviour of a probabilistic cellular automaton model for the dynamics of a population driven by logistic growth and weak allee effect. J. Phys. A: Math. Theoretical 51https://doi.org/10.1088/1751-8121/aab165. Article ID 145601.
Mitchell, M., Crutchfield, J.P., Das, R., 1996. Evolving cellular automata with genetic algorithms: A review of recent work. Proceedings of the First International Conference on Evolutionary Computation and its Applications (EvCA'96).
Richards, F.C., Meyer, T.P., Packard, N.H., 1990. Extracting cellular automaton rules directly from experimental data. Physica D: Nonlinear Phenomena 45, 189–202. https://doi.org/10.1016/0167-2789(90)90182-O.
Sapin, E., Bailleux, O., Chabrier, J.-J., 2003. Research of a cellular automaton simulating logic gates by evolutionary algorithms. Proceedings of the 6th European Conference on Genetic Programming, EuroGP'03. Springer-Verlag, Berlin, Heidelberg, pp. 414–423. https://doi.org/10.1007/3-540-36599-0_39.
Sun, X., Rosin, P.L., Martin, R.R., 2011. Fast rule identification and neighborhood selection for cellular automata. IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 41, 749–760. https://doi.org/10.1109/TSMCB.2010.2091271.
Wolfram, S., 1983. Statistical mechanics of cellular automata. Rev. Modern Phys. 55, 601–644. https://doi.org/10.1103/RevModPhys.55.601.
Yang, Y., Billings, S.A., 2000a. Extracting Boolean rules from CA patterns. IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 30, 573–580. https://doi.org/10.1109/3477.865174.
Yang, Y., Billings, S.A., 2000b. Neighborhood detection and rule selection from cellular automata patterns. IEEE Trans. Syst. Man Cybernet. Part A: Syst. Hum. 30, 840–847. https://doi.org/10.1109/3468.895912.