

J Supercomput (2014) 69:9–16  
DOI 10.1007/s11227-014-1214-x

---

## Two-dimensional patterns and images reconstruction with use of cellular automata

Jaroslav Skaruz · Franciszek Sereczynski · Anna Piwonska

Published online: 6 June 2014

© The Author(s) 2014. This article is published with open access at Springerlink.com

**Abstract** An approach based on an application of cellular automata (CA) to the problem of two-dimensional (2D) patterns or images reconstruction from ones with only partial information available is presented in the paper. 2D CA are used to process patterns/images, and genetic algorithm (GA) is applied to discover CA rules, which will be able to reconstruct original patterns/images from, e.g. destroyed or modified ones. A number of experiments have been conducted to reconstruct patterns and human face images with use of the proposed approach. Results of experiments show that CA rules discovered by GA in the learning process allow to reconstruct images with large number of damaged pixels.

**Keywords** Pattern and image reconstruction · Cellular automata · Genetic algorithms

---

J. Skaruz (✉)

Institute of Computer Science, Siedlce University of Natural Sciences and Humanities,  
3 Maja 54, 08-110 Siedlce, Poland  
e-mail: jaroslaw.skaruz@uph.edu.pl

F. Sereczynski

Department of Mathematics and Natural Sciences, Cardinal Stefan Wyszyński University in Warsaw,  
Woycieckiego 1/3, 01-938 Warsaw, Poland  
e-mail: fsereczynski@gmail.com

A. Piwonska

European Higher School of Information Technology and Economic, Bialostocka 22,  
03-741 Warsaw, Poland  
e-mail: anna@piwonska.pl

## 1 Introduction

An object (pattern, image) reconstruction problem belongs to a wide area of image processing which is important in numerous applications, such as medicine, acquisition of sensor data, pattern recognition, etc. A number of classical methods (see e.g. [3]) which have been proposed to solve these problems, can be generally classified as ones belonging to analytical or iterative methods based on algebraic properties or methods based on statistical models. One of perspective directions of developing object reconstruction algorithms is applying CA [11]. CA are complex dynamical systems, which have been successively used in solving problems or modeling phenomena in such areas as physics, computer science, biology, sociology, and also in some areas of image processing. In [9] an approach based on CA to noise filtering and thinning of binary images has been proposed. The paper [4] presented CA for an elementary image enhancement whose behaviour is determined by Lyapunov functionals. Authors of [8] applied CA to solve noise removal and border detection in images.

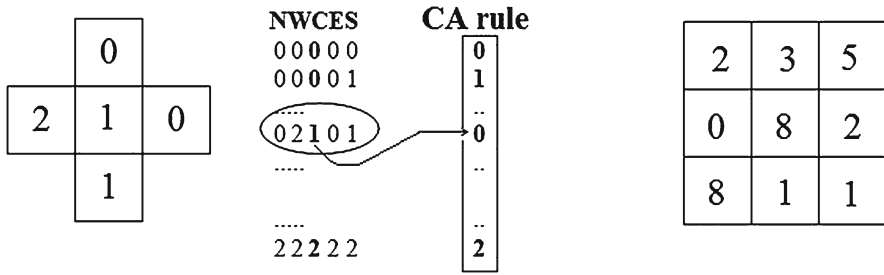
The main difficulty with applying CA is constructing CA rules producing a desired behaviour. During the 1990s, Mitchell and colleagues [6] proposed using GAs to discover CA rules, which are able to solve density classification problems. It opened a possibility of automatic generation of CA rules using artificial evolution. In particular, in [2] GA was proposed to solve AND and XOR problem in 2D CA, and Bandini et al. [1] proposed to use machine learning techniques to find CA rules able to generate patterns, which are similar to those generated by a given target rule.

In this paper we propose a GA-based approach to discover CA rules to perform object reconstruction tasks: pattern and human face image reconstruction. The rest of the paper is organized as follows: Section 2 shortly outlines CA. Section 3 explains object reconstruction problem in the context of CA. Section 4 presents details of the GA designed to find CA rules. Experimental results are reported in Sect. 5. The last section contains conclusions.

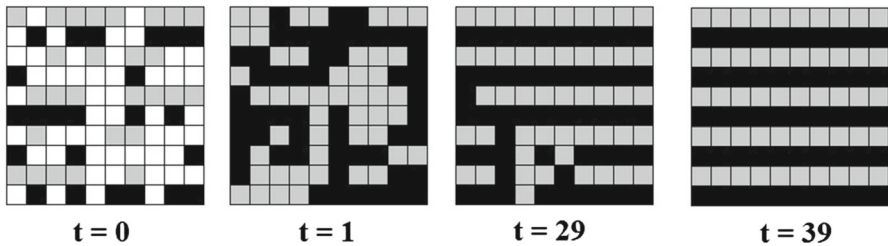
## 2 Cellular automata overview

CA are discrete computational systems composed of a number of cells arranged in a regular grid. Each cell is in one of a finite number of possible states and all cells are updated in parallel according to a CA rule, called also a transition function. The next state of a cell is determined by its previous state and states of its neighbourhood, composed by adjacent cells. Two types of neighborhood (see Fig. 1) are commonly used: the von Neumann neighborhood (the four cells orthogonally surrounding the central cell) and the Moore one (the eight cells around the central cell).

Usually CA are implemented as uniform CA, which means that all cells have the same transition function. One of the advantages of CA is that, although each cell is controlled only by one simple rule, the grid of cells by their local interactions leads to a sophisticated global behaviour.



**Fig. 1** An example of von Neumann neighbourhood, list of neighbourhood states, and example of CA rule (on the left), and Moore neighbourhood (on the right)



**Fig. 2** Configurations of the CA in time steps: 0, 1, 29 and 39

### 3 An object reconstruction problem and cellular automata

An object (pattern or image) reconstruction problem defines a recovery operation of an original form of an object from its distorted version. The malformation may be due to noise or damage of some number of pixels. This paper focuses on one aspect of an object reconstruction: complementing missing parts of an object. We assume that a given object is defined as 2D array. Each element of an array can take one of a set of possible values. We assume that some fraction  $p$  of values of grid elements is unknown. Based on such an incomplete object, it could be difficult to predict unknown cell values unless one is able to see some dependencies between values of cells.

An incomplete 2D object is interpreted as an initial configuration of a 2D CA with three states (see Fig. 2) of each cell: known values of cells corresponding to grey cells or black cells will be represented by CA states 1 or 2, respectively, and unknown values of grid elements will be represented by state 0. The object reconstruction problem can be described as follows: Let us assume that we have a finite number of random initial configurations, each of which is an incomplete pattern. We want to find a CA rule that is able to converge to a final configuration identical with a complete pattern. We also assume that a complete pattern is not known during searching process. The only data available are a series of incomplete patterns, randomly created from a given pattern.

In this paper we study two variants of the object reconstruction problem using the same approach. We will reconstruct patterns and also images corresponding to a human face. The space of potential solutions (CA rules) for each problem is huge, and therefore we will apply GA to discover solutions. This phase of the approach we

call learning (or training). In the second phase called testing we test performance of discovered solutions.

## 4 Application of GA for searching CA rules

### 4.1 Coding solution for GA-based search

To search a CA rule capable of performing object reconstruction task, we must first decide how a transition function (rule) depends on a CA neighborhood. For the pattern reconstruction problem each CA cell can be in one of three states and we define a CA rule as a function which depends on information available from cells formed by von Neumann neighborhood.

Figure 1 (left) shows that five cells of von Neumann neighborhood are described by directions on the compass: North (N), West (W), Central (C), East (E), South (S). We can list all possible states of von Neumann neighborhood related to a given central cell, and in particular  $\{02101\}$  is a current state of this neighborhood. An exemplary CA rule says that the central cell (currently in state 1) should change its state into 0 in the next moment of time. With three cell states we have  $3^5 = 243$  possible neighborhood states, and the length of the rule is also equal to 243. The number of possible rules equals to  $3^{243}$ .

For the image reconstruction problem we use images with  $300 \times 400$  pixels with eight colors. As a consequence our CA consists of  $300 \times 400$  cells and each cell can contain one of nine values representing a color of a given pixel or information that a pixel is damaged. For this problem we use Moore neighborhood with nine cells [see Fig. 1 (right)]. For nine possible states of a cell, there exist  $9^9 = 387,420,489$  possible neighbourhood states, and the number of possible rules equals to  $9^{387,420,489}$  and creates a huge space.

### 4.2 Fitness function

Fitness of an individual (CA rule) of GA is calculated according to Eq. 1

$$f = (n_c - (n_d + n_i)^t) / n, \quad (1)$$

where  $n_c$  is the number of cells with the correct state,  $n_d$  means the number of cells corresponding to damaged pixels,  $n_i$  depicts the number of cells with wrong state but not damaged pixel,  $t$  is a coefficient and  $n$  is the number of all cells. The objective of GA is to find such a CA rule that CA will allow to obtain reconstructed object with maximum number of correct pixels.

At each generation of GA [5], individuals are evaluated and fitness value corresponding to each one is used by the selection operator. It selects individuals from the current population to the temporary one proportionally to their fitness and these individuals take part in genetic modifications through crossover and mutation operators.

### 4.3 Learning phase

The objective of the learning phase is to discover by GA the best CA rules to reconstruct an object. The incomplete image is an initial configuration of the CA. GA starts with a population of randomly generated rules. In the next step all individuals are evaluated against an image with  $p$  % randomly damaged pixels. Next, a rule is applied to the CA for  $T$  time steps and the obtained image is used to calculate fitness value of the rule according to the Eq. 1. Once we have evaluated individuals GA starts to improve them through the application of selection, crossover and mutation operators. The pseudocode of the GA is presented as Algorithm 1.

---

#### Algorithm 1: GA for searching CA rules - learning mode

---

```

1: begin
2:   present an original object and create the corresponding CA;
3:   generate the initial population of CA rules of size  $P$ ;
4:   generate an object with  $p$ % damaged pixels of the original object;
5:   for each rule in the population do
6:     run CA during  $T$  time steps;
7:     compute the fitness function value;
8:   end
9:   for  $i := 1$  to  $G$  (generations) do
10:    randomly choose  $P$  rules from the current population, with replacement;
11:    divide  $P$  chosen rules into disjoint pairs;
12:    cross each pair by means of one point crossover;
13:    mutate offsprings with the probability  $p_m$ ;
14:    generate an object with  $p$ % damaged pixels of the original object;
15:    for each rule in the population do
16:      run CA during  $T$  time steps;
17:      compute the fitness function value;
18:    end
19:    end
20:    choose the best individual (CA rule) from the population as the result;
21: end

```

---

### 4.4 Testing phase

After CA rules had been discovered, it is examined to see how well CA is able to recover an original object from its distorted version. CA is created that corresponds to the same object IMG1 with  $p$  % damaged pixels, which was not used in the learning mode. CA is governed by the rule, which was assigned to the object IMG1 and evolves until a quality of the object that is reflected by the CA is not getting better. The quality  $f_1$  of reconstructed object is computed according to Eq. 2

$$f_1 = n_c/n, \quad (2)$$

where  $n_c$  is the number of correctly recovered pixels and  $n$  depicts the number of all pixels. The approach presented in this section was used to reconstruct two kinds of objects: patterns and images.

## 5 Experimental results

### 5.1 Experiment 1: reconstructing patterns

Four 2D patterns were used in experiments (for details, see [7]). In the learning mode, for each pattern we applied GA to discover a CA rule to solve the problem. The parameters of GA were the following: population size  $p = 200$ , crossover and mutation probabilities  $p_c = 0.7$ ,  $p_m = 0.02$ , respectively. We tested the performance of GA for three values of  $p$ : 0.1, 0.3 and 0.5. The maximal number of time steps  $T$  during which CA has to converge to a desired final configuration was set to 100.

Results of experiments showed that GA needed around 30 generations to discover CA rules with a good performance, despite the value of  $p$ . This performance evaluated in testing mode on the base of 1,000 random configurations shows that: (a) for  $p = 0.1$  the probability of correct reconstructing pattern ranged from 0.971 (pattern 1) to 1 (pattern 4), and (b) for  $p = 0.5$  from 0.760 (pattern 2) to 1 (for pattern 4). Figure 2 presents the initial configuration of the CA in testing mode for pattern 4 and further configurations in time steps: 1, 29 and 39, when the image was fully reconstructed.

### 5.2 Experiment 2: reconstructing human face images

In the second experiment we run Algorithm 1 for searching a CA rule for reconstructing images with a human face. Three images denoted as  $T_0$ ,  $T_2$  and  $T_3$  were used in the experiments. For each image, we tested the performance of the GA for a case of 70 % damaged pixels within an image. The maximal number of steps during which CA has to converge to a desired original image was set to 40. The parameters of GA were the following: population size = 50, crossover probability = 0.7 and mutation probability = 0.02. The searching process was conducted for 50 generations.

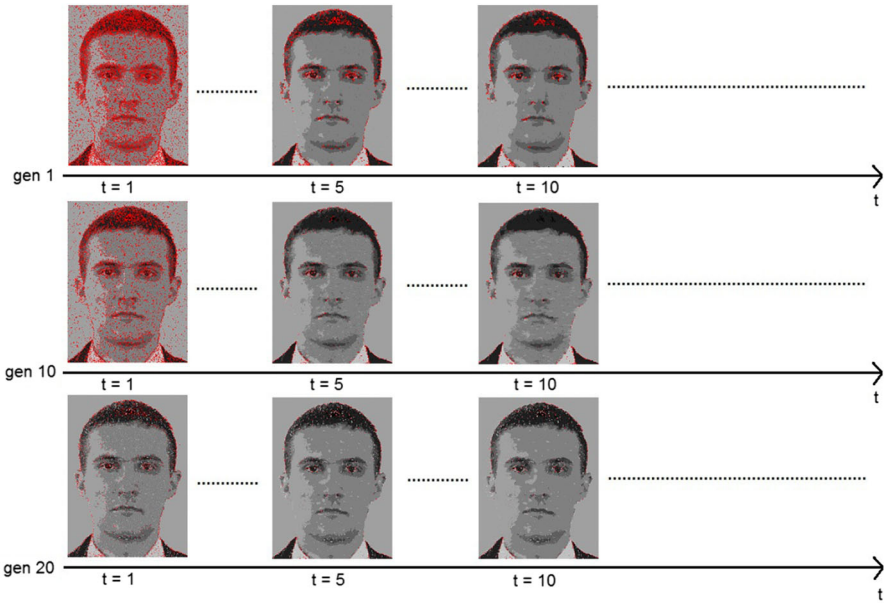
Figures 3 and 4 present how the images  $T_0$  and  $T_2$  are reconstructed in 1, 10 and 20 iterations of GA and in 1, 5 and 10 time steps of the CA.

The figures show the process of reconstructing images corresponding to one person. Three rows of the images are related to the first, tenth and twentieth iteration of GA. Three columns of images depict 1, 5 and 10 step of CA. It is shown that the quality of rules found by GA increases in subsequent iterations. In the 20th iteration of GA and the first step of the CA the number of defective pixels is very small and the quality of the images in the following steps of CA increases. Next, we examined in testing mode how the discovered CA rule is able to reconstruct an incomplete image. Figure 5 presents an image with defective pixels, which was used as input for CA.

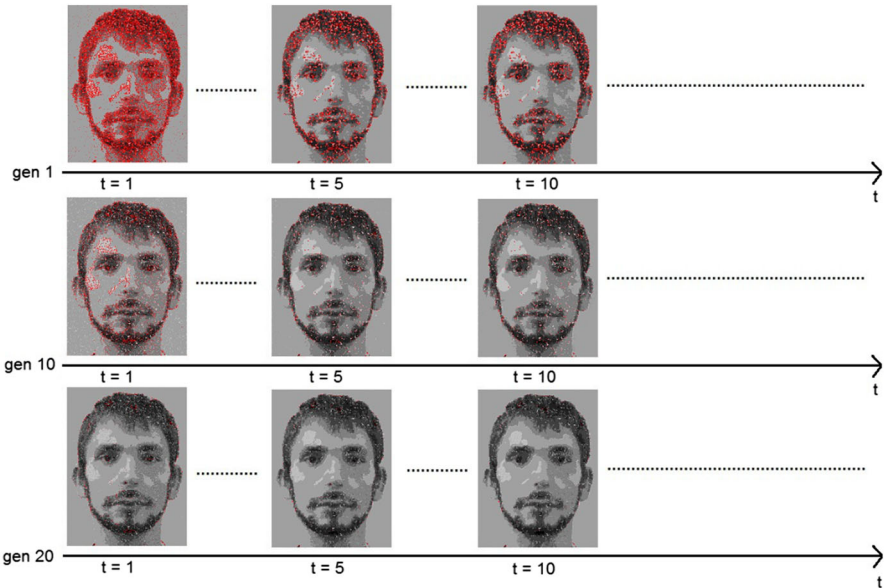
Figure 6 presents an image obtained from CA using the best rule received from the learning mode of GA. It shows that the rule *rule\_IMG\_1* assigned to the reconstructed image allows to restore the image with a high quality.

## 6 Conclusions

An approach to the pattern/image reconstruction problem based on an application of GA for searching CA rules has been proposed. Experimental results have shown that



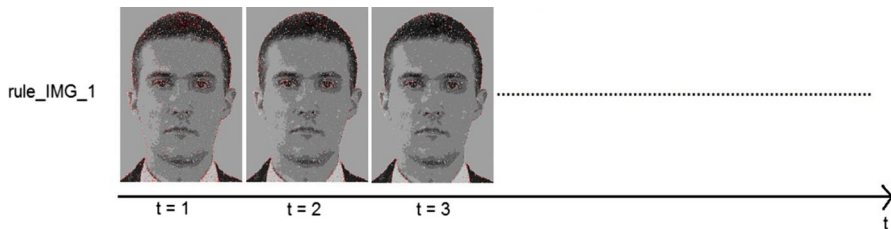
**Fig. 3** Reconstruction of  $T_0$  image in the training phase



**Fig. 4** Reconstruction of  $T_2$  image in the training phase

GA finds such good rules for CA, that even images with large number of damaged pixels can be restored to near the original image, which allows to recover incomplete patterns and images with a human face to a great extent. Performance of the pro-

**Fig. 5** Input image for testing  
CA



**Fig. 6** Testing phase: image reconstruction performed by CA

posed approach for restoring images from ones of very low quality is very promising. We believe that these results open a new possibilities in developing highly effective reconstruction algorithms.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

1. Bandini S, Vanneschi L, Wuensche A, Shehata AB (2008) A neuro-genetic framework for pattern recognition in complex systems. *Fundam Inf* 87(2):207–226
2. Breukelaar R, Back T (2004) Evolving transition rules for multi dimensional cellular automata. LNCS 3305. Springer, Berlin, pp 182–191
3. Chan TF, Shen JJ (2005) Image processing and analysis: variational, PDE, wavelets and stochastic methods. *Soc Ind Appl Math*
4. Hernandez G, Herrmann H (1996) Cellular automata for elementary image enhancement. *Graph Model Image Process* 58(1):82–89
5. Michalewicz Z (1996) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin
6. Mitchell M, Hrabec P, Crutchfield J (1993) Revisiting the edge of chaos: evolving cellular automata to perform computations. *Complex Syst* 7:89–130
7. Piwonska A, Seredynski F (2010) Discovery by genetic algorithm of cellular automata rules for pattern reconstruction task. In: ACRI 2010. LNCS 6350. Springer, Berlin, pp 198–208
8. Popovici A, Popovici D (2002) Cellular automata in image processing. In: *Proceedings of the 15th international symposium on mathematical theory of networks and systems*
9. Rosin PL (2006) Training cellular automata for image processing. *IEEE Trans Image Process* 15(7):2076–2087
10. Slatnia S, Batouche M, Melkemi K (2007) Evolutionary cellular automata based-approach for edge detection. LNCS 4578. Springer, Berlin, pp 404–411
11. Wolfram S (2002) *A new kind of science*. Wolfram Media, Champaign