



## Image reconstruction with the use of evolutionary algorithms and cellular automata

Franciszek Seredyński<sup>1,2\*</sup>, Jarosław Skaruz<sup>2†</sup>, Adrian Piraszewski

<sup>1</sup>*Cardinal Stefan Wyszyński University in Warsaw,  
Department of Mathematics and Natural Sciences,  
Wycickiego 1/3, 01-938 Warsaw, Poland*

<sup>2</sup>*Siedlce University of Natural Sciences and Humanities,  
Institute of Computer Science,  
3 Maja 54, 08-110 Siedlce, Poland*

**Abstract** – In the paper we present a new approach to the image reconstruction problem based on evolution algorithms and cellular automata. Two-dimensional, nine state cellular automata with the Moore neighbourhood perform reconstruction of an image presenting a human face. Large space of automata rules is searched through efficiently by the genetic algorithm (GA), which finds a good quality rule. The experimental results show that the obtained rule allows to reconstruct an image with even 70% damaged pixels. Moreover, we show that the rule found in the genetic evolution process can be applied to the reconstruction of images of the same class but not presented during the evolutionary one.

### 1 Introduction

Cellular automata (CA) were originally developed in the 1940s by Ulam and von Neumann. Their purpose was to provide a framework to investigate the behaviour of complex systems. Since then there has been a lot of research towards different applications of cellular automata.

CA have often been used to demonstrate the full spectrum of dynamical behaviour, e.g. hydrodynamics, thermodynamics, A-life, synchronization, simple image processing and control problems. CA have also been found useful for simulating and studying

---

\*fseredynski@gmail.com

†jaroslaw.skaruz@uph.edu.pl

phenomena such as ordering, turbulence, chaos, symmetry-breaking, and have had wide application in modelling systems in such areas as physics, computer networks, biology, and sociology. Because of quite obvious similarities between the 2-dimensional cellular automata and the images (pixels mapping to cells) uniform cellular automata have been used quite a lot in some areas of image processing. In [1] the author presented an approach to noise filtering and thinning of binary images and sequential floating forward search strategy to find a rule. Hernandez and Herrmann [2] presented CA for elementary image enhancement whose behaviour is determined by operators called Lyapunov functionals. Their results over real images compared with the common use image processing software allow to propose CA as the first level of enhancement. Popovici et al. [3] applied CA implemented on a parallel machine to solve noise removal and border detection in images. Finally CA were also extensively applied in medicine e.g. to generate image representation for biological sequences [4].

The main bottleneck of CA is a difficulty of constructing CA rules producing a desired behaviour. In some applications of CA one can design an appropriate rule by hand, based on partial differential equations describing a given phenomenon. However, it is not always possible. In the 90-s of the last century Mitchell and co-workers proposed to use the genetic algorithms to discover CA rules able to perform one-dimensional density classification task [5] and the synchronization task [6]. The results produced by Mitchell et al. were interesting and started development of a concept of automating rule generation using artificial evolution. Breukelaar and Back applied GAs [7] to solve the density classification problem as well as AND and XOR problem in two dimensional CAs. Swiecicka et al. used [8] GAs to find CA rules able to solve the multiprocessor scheduling problem. Bandini et al. proposed [9] to use several Machine Learning techniques such as GAs, Support Vector Machines and neural networks to find automatically CA rules able to generate patterns, which are similar in some generic sense to those generated by a given target rule.

In this paper we present the results of experiments concerning evolution of CA rules to perform image reconstruction task, which is related to image processing. The paper is organized as follows. Section 2 describes CA. Section 3 explains the pattern reconstruction task in the context of CAs. Section 4 presents the details of the GA designed to find CA rules. The experimental results are reported in section 5. The last section presents the conclusions.

## 2 Cellular Automata

CA are the dynamical systems in which time and space are treated in a discrete manner. A cellular automaton consists of an  $n$ -dimensional grid of cells. Each of these cells can assign one of a finite number of possible states and are updated in parallel according to a rule. The state of a cell is determined by the previous states of a cell and its neighbourhood, which is a number of adjacent cells. Two types of neighborhood 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).

There are many possible kinds of CA. Traditionally, CA are implemented as the uniform ones, which means that all cells have the same state transition function. There are also papers considering non-uniform CA with the cells having different state transition functions. The rule governing CA can take various forms e.g. totalistic, probabilistic. Large space of possible rules was also the inspiration for the application of GAs to find a rule in drastically smaller time periods. One of the advantages of CAs is that, although each cell can contain even only one simple rule, the combination of a grid of cells with their local interaction leads to more sophisticated global behaviour. Although each cell has only a view of a part of the system in the form of its neighbourhood, local information is propagated at each iteration, which has impact on the overall CA system.

### 3 Image reconstruction task

Image reconstruction is a process of recovering the original form of an image from its distorted version. The distortion may include noise, blurred shapes or damage of a number of pixels. The image enhancement is a broad field and various methods are used to restore the true image.

In this paper we consider the image reconstruction problem from different perspectives. Suppose that one day, the police caught the offender in the action. According to the criminal standard procedure he was taken to the police station and his face images were made. Unfortunately, the criminals managed to escape during the transport to the court. The search was announced based the photo made earlier. The offender to avoid imprisonment changes its appearance with a headgear and various kinds of glasses. Fig. 1 presents the example of a criminal face and his various aspects of face.

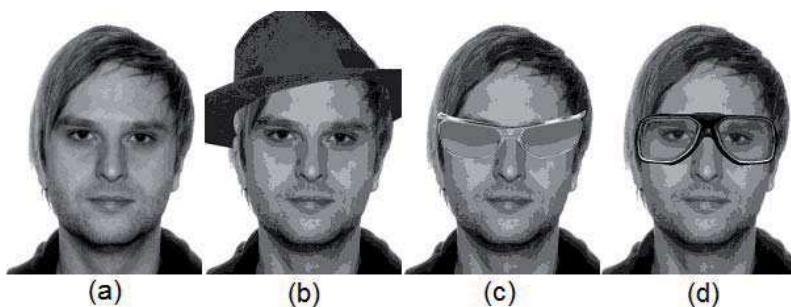


Fig. 1. The image reconstruction problem: the example of the image with the original face (a), the image presenting the same face with some modifications (b) - (d).

This article presents a new idea to solve the problem of image reconstruction. This idea consists in the use of cellular automata, whose task is to reconstruct the original

picture of a human face based on the image with a certain number of defective pixels. Due to the large number of potential rules controlling CA, the task of searching good quality rules is realized by the GA.

#### 4 The GA for searching the CA rules

In this section we present the application of GA for searching the CA rules. We assume that a given image is defined on a two-dimensional CA and one pixel corresponds to one cell of a CA. Common properties of images are the number of colours and the resolution. They are represented in CA in the form of possible states of cells and the size of CA, accordingly. In our experiments we used the images with 300x400 pixels in eight colours. As a consequence, our CA is 300x400 cells size and each cell can contain one of nine values representing a colour of a given pixel. The number in the range between zero and seven reflects a colour tone and the value of eight indicates, that a given pixel is damaged.

Greater resolution would allow to show more details in an image but it would also lead to a greater size of CA, which has a negative impact on both the time needed for image reconstruction and searching rules. In our experiments we consider CA with the Moore neighbourhood. The example of a cell with two defective pixels is illustrated in Fig. 2.

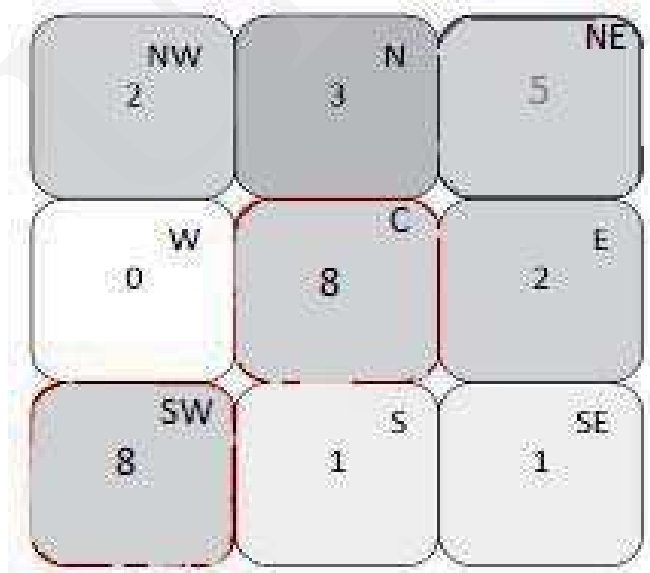


Fig. 2. Moore neighbourhood example with two cells corresponding to the damaged pixels.

Let us consider the Moore neighbourhood. For nine possible states of a cell, there exist  $9^9 = 387420489$  possible neighbourhood states. It means that the number of

possible rules equals to  $9^{387420489}$ . Searching the best rule is impossible in practice because each possible rule should be examined in the image reconstruction, which is time consuming. A less time demanding problem relates to the von Neumann neighbourhood which consists of five cells. To overcome the problem of finding a good rule in such a large search space we are going to employ GA, which is known as a good tool widely used in many combinatorial problems.

#### 4.1 Image coding scheme

While GA maintains a population of individuals - CA rules, their coding scheme must be defined. An individual must depict a state of a cell for each possible neighbourhood. Let us analyze an example of a neighbourhood and a part of an individual presented in Fig. 3.

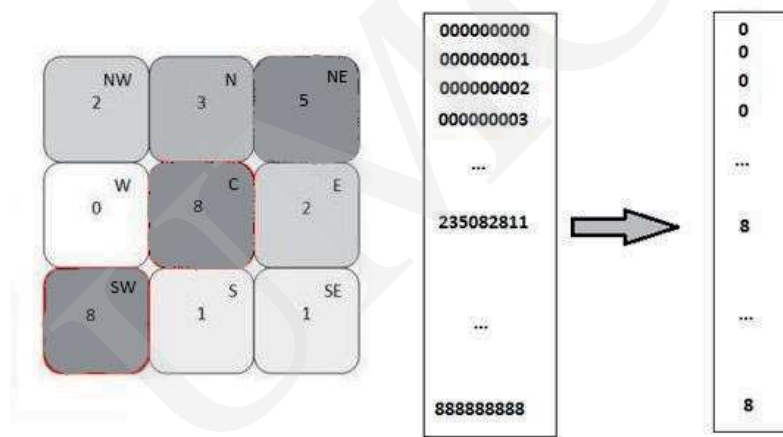


Fig. 3. The neighbourhood (on the left) and the part of the rule - the individual of the GA (on the right).

The CA rule is represented by the vector containing the values in the range between 0 and 8 representing a colour tone. Nine cells of the Moore neighbourhood are usually described by the directions: North-West (NW), North (N), North-East (NE), West (W), Central (C), East (E), South-West (SW), South(S), South-East(SE). Using this convention a neighbourhood can be interpreted as a vector containing numbers representing colour of pixels in this neighbourhood. For the neighbourhood presented in Fig. 3 such a vector will contain the values read from the left to the right and from the top to the bottom e.g. 235082811. For each possible neighbourhood there is also given a value whose central cell of a neighbourhood next will a step of CA.

#### 4.2 Fitness function

Evaluation of individuals is performed by calculation 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 proper state,  $n_d$  means the number of cells corresponding to the damaged pixels,  $n_i$  depicts the number of cells in a wrong state but not with the damaged pixel,  $t$  is a coefficient and  $n$  is the number of all cells. The objective of GA is to find such a rule that together with CA will allow to obtain a reconstructed image with the maximum number of pixels in the original colour.

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

### 4.3 Learning mode

The purpose of this stage is to find rules by the GA. 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 Eq. (1). Once we have evaluated the 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 image and create the corresponding CA;
3:   generate the initial population of CA rules of size  $P$ ;
4:   generate an image with  $p\%$  damaged pixels of the original image;
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 image with  $p\%$  damaged pixels of the original image;
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 from the population as the result;  
 21: **end**

#### 4.4 Testing mode

Having obtained the best rule from the GA, the next step is to examine how well it is able to recover an original image from its distorted version. Moreover, we want to check if the obtained rule has a generalization feature. It would mean that a rule corresponding to one image is able to perform image reconstruction for the other images of the same class. While in this paper we are concerned with the images presenting people faces, the other images relate to different persons. Testing mode of the GA is performed according to Algorithm 2.

**Algorithm 2.** GA for searching CA rules - testing mode

- 1: **begin**
- 2: get the best three rules  $rule\_IMG_1$ ,  $rule\_IMG_2$  and  $rule\_IMG_3$  from the GA that were evolved for the images:  $IMG_1$ ,  $IMG_2$  and  $IMG_3$ , respectively;
- 3: create three CAs:  $CA_1$ ,  $CA_2$  and  $CA_3$  that correspond to the same image  $IMG_1$  with the damaged pixels;
- 4: **while** *image quality increases* **do**
- 5:     perform one step of  $CA_1$  using  $rule\_IMG_1$ ;
- 6:     perform one step of  $CA_2$  using  $rule\_IMG_2$ ;
- 7:     perform one step of  $CA_3$  using  $rule\_IMG_3$ ;
- 8: **end**
- 9: **for** *each CA* **do**
- 10:     compute the quality function value;
- 11: **end**
- 12: **end**

First, we have obtained three rules in the learning mode of the GA. They were evolved to reconstruct the images  $IMG_1$ ,  $IMG_2$  and  $IMG_3$ . Next, three CAs are created that correspond to the same image  $IMG_1$  with  $p\%$  damaged pixels. The first CA is governed by the rule, which was assigned to the image  $IMG_1$ , the second CA is performed using the rule assigned to the image  $IMG_2$  and the third CA is run with the rule assigned to reconstruct the image  $IMG_3$ . For each CA, change the state of the CA until quality of the image that is reflected by the CA is not getting better. At the end of the testing mode, for each image calculate how many pixels were restored correctly. This value, denoted as  $f_1$ , 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.

## 5 Experimental results

### 5.1 Learning

We performed experiments on three images denoted as  $T0$ ,  $T2$  and  $T3$ . 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 experimental results show that even for 20 time steps CA converges. The parameters of the GA were following:  $P = 50$ ,  $p_c = 0.7$  and  $p_m = 0.02$ . The searching process was conducted for 50 iterations of the GA. Fig. 4 presents the performance of the GA.

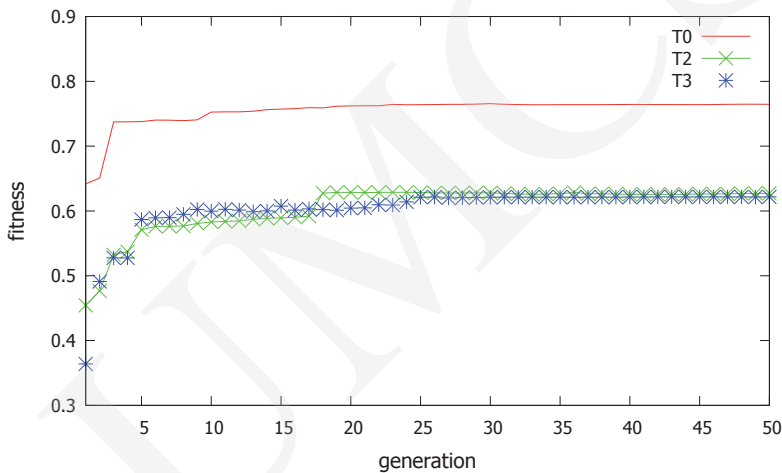


Fig. 4. GA performance

We can see that for the images  $T2$  and  $T3$  the fitness value for the randomly generated rules was between 0.35 and 0.45. In the evolutionary process GA was able to find better rules and in about 25 generation the fitness value of the best rules equals to 0.6. The performance of the GA for the image  $T0$  was better than that of the GA for the two previous images. In the next 20 iterations of the GA, the quality of found rules increased to the value about 0.77. Getting stuck in the local minimum around 25 iterations may be due to the vast space of solutions. Figs. 5 and 6 present how images are reconstructed in 1, 10 and 20 iterations of the GA and in 1, 5 and 10 time steps of the CA.

Both figures show the process of reconstructing images of two different people. Three rows of images correspond to the first, tenth and twentieth iterations of the GA. Three columns of images depict 1, 5 and 10 steps of CA. In both figures it is shown that the quality of rules found by GA increases in subsequent iterations. In the 20th iteration of the AG 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.

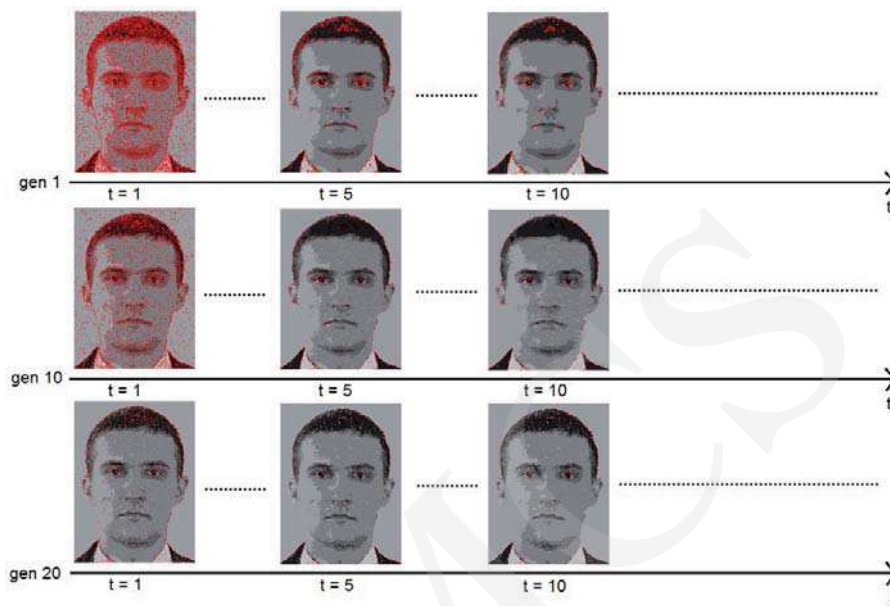


Fig. 5. Image  $T_0$  reconstruction during searching rules

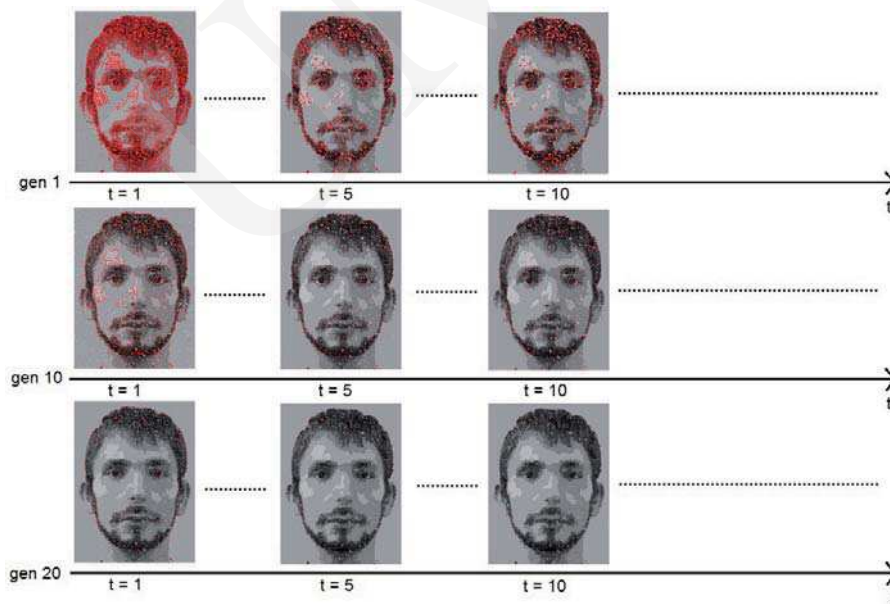


Fig. 6. Image  $T_2$  reconstruction during searching rules

## 5.2 Testing

In the testing phase of the experimental study we wanted to know if the obtained rule is able to reconstruct an image with the damaged pixels. We were also interested if the

other rules obtained for different images but of the same class are able to reconstruct an image. For this purpose three CAs were created for the same image. These CAs were performed using three different rules and at the end the quality values were calculated according to Eq. (2). Fig. 7 presents the image with defective pixels, which was used as input for three CAs.



Fig. 7. Image with 70% damaged pixels

Fig. 8 presents the images from three CAs using three rules obtained from three various images. The images within the first row were obtained by the application of the rule dedicated to restore such an image. The images within the two next rows have been restored using the rules dedicated to the other, different images. The calculated quality value for the images within the first, second and third rows equals to 0.87, 0.82 and 0.85 respectively. It means that the images have been reconstructed with 87%, 82% and 85% correct pixels. These results show that the rule *rule\_IMG\_1* assigned to the reconstructed image allows to restore it with high quality. Moreover, despite the fact that the other two rules were assigned to different images, their application allows to restore the image with nearly the same quality as the rule *rule\_IMG\_1*. Such result proves that there exists a subset of rules that are enough to restore a number of images, but they must be the same class.

## 6 Conclusions

In this paper we have presented a new approach to the image reconstruction task based on the application of GA for searching rules for CA. Our simulation results show that the images with a large number of damaged pixels can be restored to nearly the original image, which allows to recognize a face presented in the image. Moreover, we proved that the rule found by GA has a generalization feature. It means that it can be used to reconstruct images of the same class as the one, for which the rule was evolved by the GA.

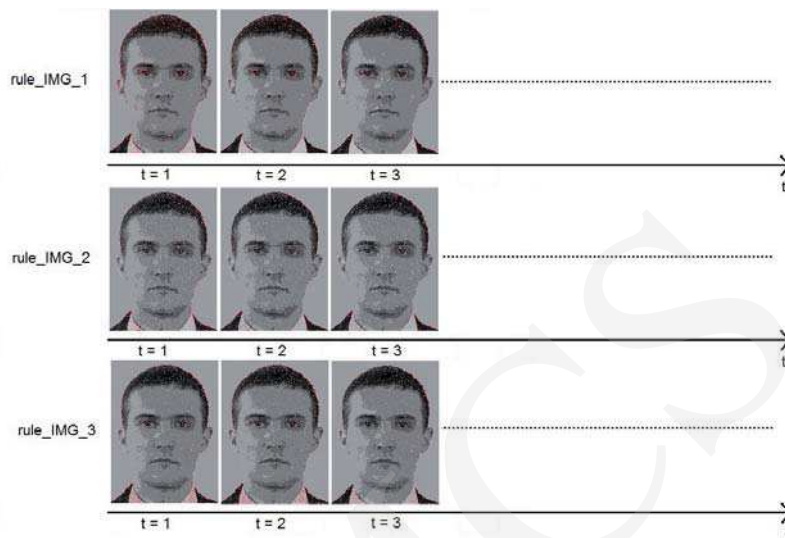


Fig. 8. GA Testing

The performance of the GA for searching rules and the CA for restoring the original image from a very low quality image is very promising. We believe that this work could be the initial research that would allow to put the new approach into practice and take advantage of it in the commercial software.

## References

- [1] Rosin P. L., Training Cellular Automata for Image Processing, *IEEE Transactions on Image Processing* 15(7) (2006): 2076.
- [2] Hernandez G., Herrmann H., Cellular automata for elementary image enhancement, *Graphical Models And Image Processing* 58(1) (1996): 82.
- [3] Popovici A., Popovici D., Cellular automata in image processing, in *Proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems*, University of Notre Dame (2002).
- [4] Xiao X., Shao S., Ding Y., Huang Z., Chou K-C., Using cellular automata images and pseudo amino acid composition to predict protein subcellular location, *Amino Acids* 30 (2006): 49.
- [5] Mitchell M., Hraber P., Crutchfield J., Revisiting the edge of chaos: Evolving cellular automata to perform computations, *Complex Systems* 7 (1993): 89.
- [6] Das R., Crutchfield J., Mitchell M., Evolving globally synchronized cellular automata, *Proceedings of the 6th International Conference on Genetic Algorithms* (1995): 336.
- [7] Breukelaar R., Back T., Evolving transition rules for multi dimensional cellular automata, *Lecture Notes in Computer Science* 3305 (2004): 182.
- [8] Swiecicka A., Sereďyński F., Zomaya A.Y., Multiprocessor scheduling and rescheduling with use of cellular automata and artificial immune system support, *IEEE Transactions on Parallel and Distributed Systems* 17(3) (2006): 253.
- [9] Bandini S., Vanneschi L., Wuensche A., Shehata A. B., A Neuro-Genetic Framework for Pattern Recognition in complex Systems, *Fundamenta Informaticae* 87(2) (2008): 207.