International Journal of Bifurcation and Chaos, Vol. 16, No. 5 (2006) 1419–1434

GENERATION OF SYMMETRICAL COLORED IMAGES VIA SOLUTION OF THE INVERSE PROBLEM OF CHEMICAL REACTIONS DISCRETE CHAOTIC DYNAMICS

V. GONTAR* and O. GRECHKO
International Group for Chaos Studies,
Department of Industrial Engineering and Management,
Ben-Gurion University of the Negev,
P. O. Box 653, Beer Sheva 84105, Israel
**galita@bgumail.bgu.ac.il*

Received February 10, 2005; Revised May 12, 2005

An automatic procedure for generating colored two-dimensional symmetrical images based on the chemical reactions discrete chaotic dynamics (CRDCD) is proposed. The inverse problem of derivation of symmetrical images from CRDCD mathematical models was formulated and solved using a special type of genetic algorithm. Different symmetrical images corresponding to the solutions of a CRDCD mathematical model for which the parameters were obtained automatically by the proposed method are presented.

Keywords: Discrete chaotic dynamics; symmetrical images; genetic algorithm; inverse problem; artificial brain.

1. Introduction

Developments in the last decades in the area of fractals, L-systems, and cellular automata have led to the emergence of a new field of computer science, "mathematical imaging". Computer generated images are finding applications in the modeling and investigation of complex natural objects, natural processes, and their evolution [Mandelbrot, 1977; Peitgen *et al.*, 2004; Prusinkiewicz & Hanan, 1989; Wolfram, 2002; Ilachinski, 2001]. The best known and most widespread examples of computer generated images are fractals. Fractals are complex self-similar patterns with noninteger dimensions that can be generated by simple geometrical reduplication, as well as using different sets of difference equations.

Another engine for image generation, L-systems, was introduced by Lindenmayer [Prusinkiewicz & Hanan, 1989]. An L-system is a formal grammar for generating strings, i.e. it is a collection of rules. By recursively applying the rules of the L-system to an initial string, a string with fractal structure can be created. Interpreting this string as a set of graphical commands allows the fractal to be displayed. L-systems are very useful for generating realistic plant structures.

The cellular automata approach is also used for artificial image creation. Cellular automata (CA) are a class of deterministic computer models characterized by a discrete lattice of homogeneous cells with a finite number of possible discrete states and rules governing interactions between the cells. Such a system evolves by an iterative process for updating the cells using logical transition rules that take into account neighborhood impact. Visualization of CA can provide a variety of visual patterns. In [Nowak & May, 1993; Nowak *et al.*, 1994; Wiederien & Udwadia, 2004], deterministic rules for CA are derived from evolutionary game theory (Prisoner's Dilemma), and various 2D images are presented.

In spite of the fact that images generated by the methods described above sometimes resemble naturally observed or created patterns, the purely mathematical character of these models (they are not related to physical first principles or to fundamental laws of nature) makes it difficult to establish relations between the model's parameters and physical characteristics of real experiments.

In this paper we introduce an automatic system for determining the parameters of the model and generating images resulting from the basic equations of chemical reactions discrete chaotic dynamics (CRDCD), formulated in accordance with fundamental physicochemical principles and laws of nature [Gontar, 1981; Gontar & Il'in, 1991; Gontar, 1997].

The problem of computer imaging that we examine below is known as the inverse problem. This could be formulated for our case as identification of the specific type of CRDCD difference equations and parameter values that correspond to the desired image(s). Among the heuristic optimization and search methods available, the genetic algorithm (GA) is often the tool of choice in the difficult field of discrete inverse problems. For example, in [Shonkwiler *et al.*, 1991] a genetic algorithm was used to solve the "1-D fractal inverse problem". The attractor was viewed by Shonkwiler *et al.* as a fractal set, and the inverse problem consisted in finding the specific set of difference equations whose attractor was given. The GA proposed by Shonkwiler *et al.* was tested on 2- and 3-map attractors, and the inverse problem was successfully solved.

In [Vences & Rudomin, 1997] the use of GA was extended to the 2D fractal inverse problem for the purpose of compressing a single image and image sequences. GA-based searching for a particular map for image encoding was introduced, tested, and shown to be robust.

Mitchell *et al.* [1994] applied a GA for evolving CA to perform a particular computational task (one-dimensional density classification). There the inverse problem consisted of finding CA rules capable of generating a specific class of images. Based on this task, the behavior of the GA was analyzed in depth, and the GA was recommended for use for the given class of problems.

The application of GAs to solve an inverse problem in the field of finite automata similar to L-systems was considered by Leblanc *et al.* [1998]. The inverse problem was to find the type of automaton (rules) by which a given string was generated. The performance of the GA was tested on two target words prefixed at fixed points on two different automata using six symbols, with the maximal length of words limited to six.

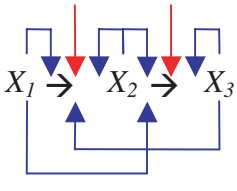
An example of application of the GA to search for a rewriting expression describing leaf shapes or L-systems is discussed by Rodkaew *et al.* [2004]. An L-system was used to construct the shape of a given rewriting expression, and a GA to search for the rewriting expression's fitting parameters. The authors reported that after 200 GA generations, the resulting shape is better than that produced by manual adjustment of the parameters.

Based on these reports regarding the use of different modifications of the GA for similar inverse problems, we chose to use a real coded genetic algorithm (RCGA) [Eshelman & Schaffler, 1993; Deb & Agrawal, 1995] to search for CRDCD difference equations parameters corresponding to particular type of images (for example, symmetrical).

2. Background and Mathematical Model

According to CRDCD theory, the initial hypothesis about the mechanism of transformation of the constituents (agents) X_i is given by the molecular and stoichiometric matrices and includes some mechanism of "information exchange" between the agents [Gontar, 1997]. For our purposes, we will consider agent transformations at a discrete time t_q, spatially distributed on the discrete 2D square lattice. Each cell of the lattice has coordinates x and y which are specified by the integers $(R_x, R_y), R_x, R_y = 1, 2, \ldots, R$.

In discussing the agents' transformations, designated by $(\rightarrow)$, we use blue arrows to denote the local influence ("information exchange" within each cell of the lattice) of the corresponding agents X_i whose concentrations were defined at the previous moment of discrete time t_{q-1} on the characteristics (for example, the rate constants) of the process of transformation at the discrete time t_q. Red arrows denote "information exchange" emanating from the concentrations of cells adjacent to the cell under consideration. Here we will consider one of the possible mechanisms of transformation of the agents:



(1)

with the corresponding molecular and stoichiometric matrices $\|a_{ij}\|$ and $\|\nu_{li}\|$:

$$\|a_{ij}\| = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix}; \quad \|\nu_{li}\| = \begin{vmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{vmatrix} \tag{2}$$

where $i = 1, 2, 3,\ j = 1,\ l = 1, 2$.

According to CRDCD, for our mechanism of transformations (1) we will have the following system of equations:

$$\frac{X_2^{t_q}(R_x, R_y)}{X_1^{t_q}(R_x, R_y)} = k_1 \exp\left\{-\left[\sum_{i=1}^{3} \alpha_{1i} X_i^{t_{(q-1)}}(R_x, R_y) + S_1\right]\right\} \tag{3}$$

$$\frac{X_3^{t_q}(R_x, R_y)}{X_2^{t_q}(R_x, R_y)} = k_2 \exp\left\{-\left[\sum_{i=1}^{3} \alpha_{2i} X_i^{t_{(q-1)}}(R_x, R_y) + S_2\right]\right\} \tag{4}$$

$$X_1^{t_q}(R_x, R_y) + X_2^{t_q}(R_x, R_y) + X_3^{t_q}(R_x, R_y) = b \tag{5}$$

$$\begin{aligned} S_l = \sum_{i=1}^{3} [&\beta_{li}^1 X_i^{t_{(q-1)}}(R_x - 1, R_y - 1) \\ &+ \beta_{li}^2 X_i^{t_{(q-1)}}(R_x - 1, R_y) \\ &+ \beta_{li}^3 X_i^{t_{(q-1)}}(R_x - 1, R_y + 1) \\ &+ \beta_{li}^4 X_i^{t_{(q-1)}}(R_x, R_y - 1) \\ &+ \beta_{li}^5 X_i^{t_{(q-1)}}(R_x, R_y) \\ &+ \beta_{li}^6 X_i^{t_{(q-1)}}(R_x, R_y + 1) \\ &+ \beta_{li}^7 X_i^{t_{(q-1)}}(R_x + 1, R_y - 1) \\ &+ \beta_{li}^8 X_i^{t_{(q-1)}}(R_x + 1, R_y) \\ &+ \beta_{li}^9 X_i^{t_{(q-1)}}(R_x + 1, R_y + 1)] \end{aligned} \tag{6}$$

$X_i^{t_q}(R_x, R_y)$ is the concentration of the ith agent calculated in each cell of the lattice with coordinates (R_x, R_y) and characterizing the system's particular state at the discrete time $t_q\,(q = 1, 2, \ldots, Q)$. Here $\|a_{ij}\|$ is a molecular matrix defining the number of components of type "j" in the ith constituent, $\|\nu_{li}\|$ is a matrix of stoichiometric coefficients reflecting the mechanism of agents chemical transformations, b is the total concentration of the main constituent, k_l-rate constant of lth reaction, α_{li} are empirical parameters characterizing the local "information exchange" taking place between the constituents inside the considered cell, β_{li}^r are empirical parameters characterizing the "information exchange" emanating from the constituents in eight closest neighboring cells including considered cell $(r = 1, 2, \ldots, 9)$ (Fig. 1).

For any given X_i, at iteration $q - 1$ and for any set of parameters $b, k_l, \alpha_{li}, \beta_{li}^r\,(b, k_l > 0, r = 1, 2, \ldots, 9)$, system (3)–(5) becomes a system of nonlinear algebraic equations and has a unique positive solution $(X_i > 0)$.

Obviously, Eqs. (3)–(5) could be written as

$$X_i^{t_q}(R_x, R_y) = F(X_i^{t_{q-1}}(R_x, R_y), \theta), \tag{7}$$

where the function F is defined by Eqs. (3)–(6) and $\theta(b, k_l, \alpha_{li}, \beta_{li}^r)$ is a vector of the system's parameters. In order to calculate $X_i^{t_q}(R_x, R_y)$ $(q = 1, 2, \ldots, Q)$, we need to solve the system of nonlinear algebraic equations [Eqs. (3)–(5)] by one of the numerical methods for any given $\theta(b, k_l, \alpha_{li}, \beta_{li}^r)$ starting from the initial and boundary conditions.

The initial conditions are:

$$\begin{aligned} &X_1^{t_0}(R_x, R_y) = b, \quad X_2^{t_0}(R_x, R_y) = 0, \\ &X_3^{t_0}(R_x, R_y) = 0, \qquad R_x, R_y = 1, 2, \ldots, R. \end{aligned} \tag{8}$$

The boundary conditions are:

$$X_i^{t_q}(R_x, R_y) = \begin{cases} X_i^{t_q}(R_x, R_y), & 1 \le R_x, R_y \le R \\ & \text{(inside the lattice)} \\ 0, & R_x, R_y < 1; R_x, R_y > R \\ & \text{(outside the lattice)} \end{cases} \tag{9}$$

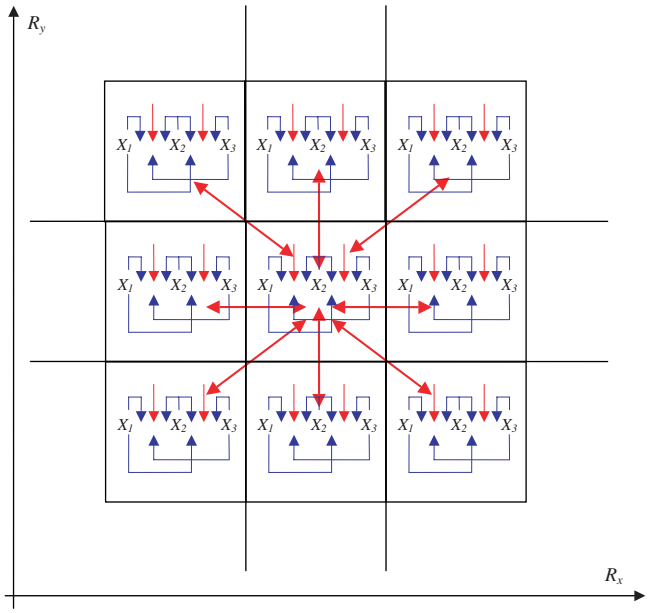


Fig. 1. Nine cells of the discrete square lattice $(R_x, R_y = 1, 2, \ldots, R)$. Blue arrows denote local "information exchange" within the cell and red arrows denote "information exchange" with the closest neighbors.

We define our direct task as that of finding the solution $(X_i^{t_q}(R_x, R_y))$ of Eq. (7) for the given parameters $\theta(b, k_l, \alpha_{li}, \beta_{li}^r)$, with presentation of the solutions obtained as a 2D image for each $X_i^{t_q}(R_x, R_y), R_x, R_y = 1, 2, \ldots, R$. The solutions $X_i^{t_q}(R_x, R_y)$, calculated at the discrete time t_q $(q = 1, 2, \ldots, Q)$ and distributed on a discrete square lattice of size $R \times R$, will be visualized in the form of sequences of 2D images (the total number of images in a sequence is equal to Q). Each sequence of images will correspond to the calculated concentrations of one of the agents $X_1^{t_q}(R_x, R_y), X_2^{t_q}(R_x, R_y)$ or $X_3^{t_q}(R_x, R_y)$. The values of the concentration, for example $0 < X_1^{t_q}(R_x, R_y) < b$, will be encoded by a colored palette consisting of N different colors taken in an arbitrarily chosen order and therefore providing a resolution of b/N in each cell of the lattice. Each cell within the lattice presented on a computer screen will be filled with M pixels (1, 4 or 9).

At the discrete time t_0 the concentration $X_1^{t_0}(R_x, R_y)$ is b and concentrations $X_2^{t_0}(R_x, R_y) = 0, X_3^{t_0}(R_x, R_y) = 0$ in each cell of the lattice in accordance with the initial and boundary conditions [Eqs. (8) and (9)]. In order to generate a sequence of Q images where each q-image corresponds to the concentration distribution $X_1^{t_q}(R_x, R_y)$ calculated according to the Eqs. (3)–(5) at discrete time t_q $(q = 1, 2, \ldots, Q)$ we used the following procedure. Calculations begin for the cell in the upper left corner of the lattice, then proceed along the first row from left to right, continue along the second row from left to right, and so on. In principle, all cells of the lattice could be updated simultaneously for each t_q if parallel computing for each cell would be applied. The "information exchange" at any discrete time t_q is introduced through the concentrations calculated at the previous moment of time t_{q-1}. For example, for the cell with coordinates $(2, 2)$ at time t_2, we take the values of the concentrations $X_i^{t_1}(R_x, R_y)$ in the nine cells closest to it (including the cell under consideration), calculated at the previous moment of time t_1: $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 1)$, $(2, 2)$, $(2, 3)$, $(3, 1)$, $(3, 2)$, $(3, 3)$.

The CRDCD mathematical model [Eqs. (3)–(5)] has 63 independent parameters and therefore could generate numerous different solutions $X_i^{t_q}(R_x, R_y)$ which could be presented as different types of 2D images [Gontar, 1997, 2000]. Provided the parameters are known, the fact that they are numerous does not pose a problem for the direct task. However, when solving the inverse problem — estimation of the parameters corresponding to the desired images — a multiplicity of parameters entails extremely complex computations.

Let us begin by investigating the use of GA search procedures to estimate the parameters of our model for specific images with a reduced number of parameters. We will assume the following:

$$\begin{aligned}
&\alpha_{li} = \alpha_{l'i}, && l \neq l', \quad l = 1, 2 \quad l' = 1, 2\\
&S_l = S_{l'}, && l \neq l', \quad l = 1, 2 \quad l' = 1, 2\\
&\beta_{li}^r = \beta_{l'i}^r, && l \neq l', \quad l = 1, 2 \quad l' = 1, 2\\
&\beta_{li}^r = \beta_{li}^{r'}, && r \neq r', \quad r = 1, 2, 3, \ldots, 9\\
& && \qquad\quad r' = 1, 2, 3, \ldots, 9
\end{aligned}$$

Hence the number of parameters is reduced to nine: $b, k_1, k_2, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3$.

3. RCGA for Automatic Retrieval of Desired Images

Genetic algorithms are inspired by biological processes of inheritance, mutation and natural selection, and the genetic crossover that occurs when parents mate to produce offspring [Goldberg, 1989]. GAs are typically implemented as a computer simulation in which a population of abstract representations of candidate solutions to an optimization or search problem are stochastically selected, recombined, mutated, and then either eliminated or retained, based on their relative fitnesses as given by a fitness function. The candidate solution is called a chromosome and consists of genes, which are the parameters being sought. A fixed number of chromosomes are joined into a population. Genetic operators are applied to the population in order to produce the next generation of the population. The process repeats itself iteratively till the stop criterion is reached.

Faced with a discrete system of nonlinear difference equations [Eqs. (3)–(5)] with a large number of model parameters and the need to design an automatic search procedure that could give us the parameters corresponding to the desired types of images (patterns) within a reasonable computational time, we examined the possibility of using real coded genetic algorithms — RCGAs. The main difference between the real coded GA method [Eshelman & Schaffler, 1993] and the classical GA method is that in the former the search parameters are encoded in chromosomes as floating point numbers. Real encoding of chromosomes is recommended in cases of continuous parameters

and when each of the parameters has an allowed interval.

Let us consider the following type of encoding of our model's parameters [Eqs. (3)–(5)] into chromosomes. Chromosome P is a row vector of real numbers corresponding to the parameters (genes), and the size of the vector is equal to the number of parameters (nine in our case: $b, k_1, k_2, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3$). Each parameter varies within the intervals, which in our case were arbitrarily chosen under the physical constraints: $b, k_1, k_2 > 0$, and other parameters could be also negative. Let us start with a population consisting of 20 chromosomes. We are supposing that we have no initial information regarding the values of the searching parameters, and therefore that each chromosome in the population is initialized by applying uniform random number distribution for each parameter within its given interval.

The fitness function should reflect our main goal, which is to generate images of a specific type. The fitness function f_{obj} should evaluate the 2D images generated by the difference equations [Eqs. (3)–(5)] in order to direct automatic RCGA search to parameters that will correspond to the desired images. Below we use RCGA to search for symmetrical images.

We propose the following fitness function, which is algorithmically defined by a four-step procedure:

Step 1. Calculate Eqs. (3)–(5) for the given size of lattice, $R \times R$, and for the given number of iterations $t_q, q = 1, 2, \ldots, Q$. In the work presented here, we set $R = 40, Q = 100$ (to reduce computer time). At the last iteration ($Q = 100$) we obtained the values of $X_1^{t_{100}}(R_x, R_y)$ on the two diagonals inside the 40×40 square lattice for purposes of further examination in Steps 2–4 (an example of lattice diagonals for $R = 40$ is shown in Fig. 2).

Step 2. Determine the differences between the three values of $X_1^{t_{100}}(R_x, R_y)$ closest to the lattice corner along the main diagonal (cells with coordinates $X_1(1,1), X_1(2,2), X_1(3,3)$ in Fig. 2). The pattern begins to emerge from the corners and frame of the lattice, because of the boundary conditions: they are nonsymmetrical as between the corners and the borders (frame). Hence the appearance of differences in color in the corner indicates that the current parameters have a greater chance of corresponding to the desired pattern. If there are no differences between these values, the fitness function procedure is stopped at $f_{\text{obj}} = 0$ (the pattern obtained corresponds to a lattice filled with just one color, as shown in Fig. 3(a)). If three different colors (X_1 values) are revealed, we proceed to Step 3.

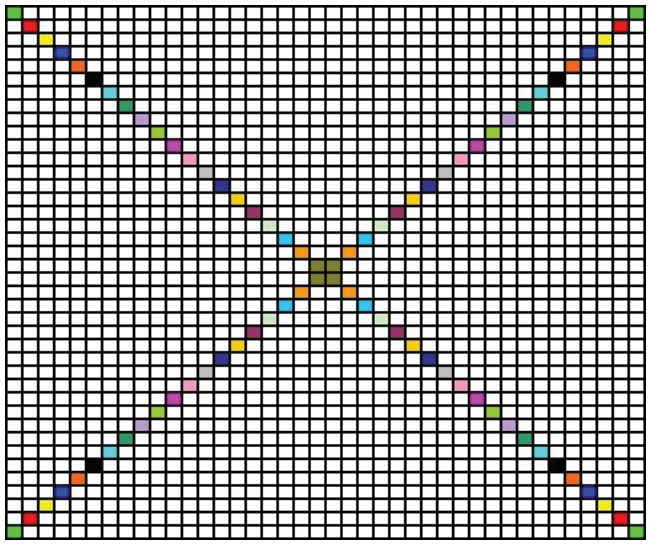
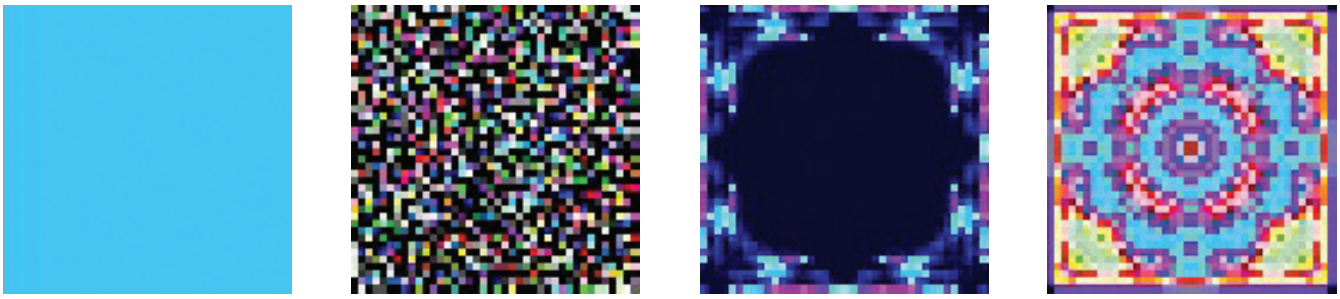


Fig. 2. Example of a 40×40 square lattice with calculated diagonal values of $X_1^{t_{100}}(R_x, R_y)\,(R_x, R_y = 1, 2, \ldots, 40)$ in colors (case of symmetry).

Step 3. Determine the symmetry of $X_1^{t_{100}}(R_x, R_y)$ values taken on two diagonals in relation to the center point of the lattice. For example, in Fig. 2 the same colors appear on the first diagonal ($X_1(1,1) = X_1(40,40), X_1(2,2) = X_1(39,39), \ldots, X_1(20,20) = X_1(21,21)$) and on the second diagonal ($X_1(1, 40) = X_1(40,1), X_1(2,39) = X_1(39,2), \ldots, X_1(20, 21) = X_1(21,20)$). If nonequal values are found on the two diagonals, the fitness check is stopped at the fitness value $f_{\text{obj}} = 1$ (the image obtained is disordered, as shown in Fig. 3(b)); if not, proceed to Step 4.

Step 4. Check three values of $X_1^{t_{100}}(R_x, R_y)$ around the center of the main diagonal (cells with coordinates $X_1(20,20), X_1(19,19), X_1(18,18)$ in Fig. 2). If they are equal, then $f_{\text{obj}} = 2$ and the obtained image will be symmetrical, with different colors appearing only along the frame,while the center will be filled with one color [see Fig. 3(c)]; if not, then the pattern obtained may be considered to be a symmetrical pattern [see Fig. 3(d)] and $f_{\text{obj}} = 3$.

Thus in our case the fitness function can have four discrete integer values $f_{\text{obj}} = \{0, 1, 2, 3\}$, which



(a) (b) (c) (d)

Fig. 3. Example of images (40×40 lattice) with: (a) $f_{\text{obj}} = 0$, (b) $f_{\text{obj}} = 1$, (c) $f_{\text{obj}} = 2$, (d) $f_{\text{obj}} = 3$.

should reflect the subtype of pattern that was generated by the concrete set of parameters embedded into the chromosomes:

$f_{\text{obj}} = 0$ — corresponds to a lattice filled with just one color;
$f_{\text{obj}} = 1$ — corresponds to a disordered pattern;
$f_{\text{obj}} = 2$ — corresponds to a symmetrical pattern with different colors confined to the frame;
$f_{\text{obj}} = 3$ — corresponds to a symmetrical pattern.

The RCGA is considered to have converged if at least one chromosome with fitness value $f_{\text{obj}} = 3$ is found, which is why the stopping criterion is $f_{\text{obj}} = 3$.

As is well known, the main operators of GAs are selection, crossover, and mutation [Goldberg, 1989]. For every given type of optimization problem, we have to decide what particular internal procedures for main genetic operators to use. Selection, which determines which chromosomes belonging to the current population will participate in the production of the next generation, is biased towards chromosomes exhibiting better fitness. The fact that we are using a specific type of fitness function (defined algorithmically) determines our choice of selection operator. Here we used binary tournament selection adjusted to our case: chromosomes of the previous generation are divided into two groups according to our fitness function values ($f_{\text{obj}} = \{0, 1, 2\}$). The first group will include chromosomes with $f_{\text{obj}} = \{1, 2\}$, because such chromosomes have a higher potential for generating the desired image in the next generation. Therefore, we want to ensure selection of at least one of these chromosomes. The second group will include chromosomes with $f_{\text{obj}} = \{0\}$. First and second parent chromosomes, $P^{(1)}$ and $P^{(2)}$, are selected randomly from the first and second groups, respectively, with equal selection probability. If all the chromosomes of the previous generation prove to belong exclusively to one group (the first or the second), then both parent chromosomes $P^{(1)}$ and $P^{(2)}$ are selected randomly with equal selection probability from the whole collection of chromosomes of the previous generation.

After the selection has been made, the crossover (or recombination) operation is performed upon the selected chromosomes. Following [Deb & Agrawal, 1995], we use a simulated binary crossover (SBX) operator with crossover parameter $\eta = 5$. The SBX algorithm for producing two offspring from two parents for the kth gene ($k = 1, 2, \ldots, 9$) of chromosomes $P^{(1)}$ and $P^{(2)}$ (the current and the next generations are denoted respectively by g and $g+1$) is as follows:

1. Generate a random number u between 0 and 1.
2. Calculate the spread factor γ_k:

$$\gamma_k = \begin{cases} (2u)^{\frac{1}{\eta+1}} & \text{if } u \le 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \tag{10}$$

3. Calculate the kth gene of offspring chromosomes:

$$P_k^{(1,g+1)} = 0.5[(1+\gamma_k)P_k^{(1,g)} + (1-\gamma_k)P_k^{(2,g)}]$$
$$P_k^{(2,g+1)} = 0.5[(1-\gamma_k)P_k^{(1,g)} + (1+\gamma_k)P_k^{(2,g)}] \tag{11}$$

For example, let us consider two parent chromosomes consisting of nine genes:

$$P^{(1,g)} = (4.5, 3, 1, -4, -4, 2, 0.5, 0.25, 0.7)$$
$$P^{(2,g)} = (1, 2.8, 2, 3, -7, -6, 8.8, -1.2, 5.5)$$

Next, we generate the random number "u" (say, $u = 0.3$). According to Eq. (10), $\gamma_1 = (2 \cdot 0.3)^{\frac{1}{5+1}} = 0.9184$. Applying Eq. (11), we obtain the following values for the first parameter (gene) for the two

offspring chromosomes:

$$P_1^{(1,g+1)} = 0.5[(1 + 0.9184) \cdot 4.5 + (1 - 0.9184) \cdot 1] = 4.3572,$$

$$P_1^{(2,g+1)} = 0.5[(1 - 0.9184) \cdot 4.5 + (1 + 0.9184) \cdot 1] = 1.1428.$$

Therefore the gene value 4.5 in $P^{(1,g)}$ should be replaced by 4.3572 and the gene value 1 in $P^{(2,g)}$ by 1.1428. Equations (10) and (11) are used nine times to create offspring chromosomes.

The application of selection and crossover should be repeated ten times (half the population size) to create the new generation of chromosomes. Thus, a single generation of the RCGA for a population of twenty chromosomes will include twenty applications of the selection operator, ten applications of the crossover operator, and twenty applications of the fitness function.

If for the first fifty generations the algorithm does not converge ($f_{\text{obj}} = 3$ is not found), random mutation is applied to all chromosomes of the current population in order to move to a different area of parametric space. This means that all the parameters in the population are randomly renewed by uniform random number distribution (within the allowed interval for each parameter). This mutation procedure is repeated every fifty generations until the algorithms converge. The set of parameters corresponding to the desired image — in the form of a symmetrical pattern for the 40×40 lattice — may then be used to generate a symmetrical image on a lattice of any size.

4. Results and Discussion

Application of RCGA to the CRDCD inverse problem formulated above resulted in sets of parameters corresponding to different patterns (images) with different evolutionary scenarios when $q = 1, 2, \ldots, Q$.

Having obtained the parameters corresponding to the symmetrical images by RCGA for a 40×40 lattice, we can then use these parameters to generate an image in any size of the lattice (here we are presenting images in a 100×100 lattice). Each cell of the lattice was drawn on the computer screen using four pixels and was assigned a color from a palette of 128 colors (Fig. 4) corresponding to the value of $X_1^{tq}(R_x, R_y)$ calculated with Eqs. (3)–(5). As an example, twelve images taken from the sequence of $Q = 100$ images ($q = 1, 10, 20, \ldots, 80, 90, 95, 100$) and corresponding to the set of parameters defined by RCGA are shown in Fig. 5.

It is interesting to look at the structure of the parameter search space and evaluate the regions that contain symmetrical patterns. Accordingly, we present cross-sections of our nine-dimensional parameter space for the two parameters b, α_1 (Fig. 6) and for the three parameters b, α_1, β_1 (Fig. 7) within the intervals $(0, 1]$. The coordinates of the blue points correspond to the parameter values of symmetrical patterns (the remaining seven and six parameters, respectively, are fixed).

Based on the examples shown here, we can form an initial conception of the searching space. The cross-sections reveal that the areas of parameters corresponding to symmetrical images have complex, nonhomogeneous structures, as well as much larger areas of "empty space" (no symmetrical patterns).

Let us look at the results of applying RCGA to a search for symmetrical images. The allowed intervals for RCGA initialization were set as (0, 10] for the parameters b, k_1, k_2 and $[-10, 10]$ for the parameters $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3$. In order to evaluate the

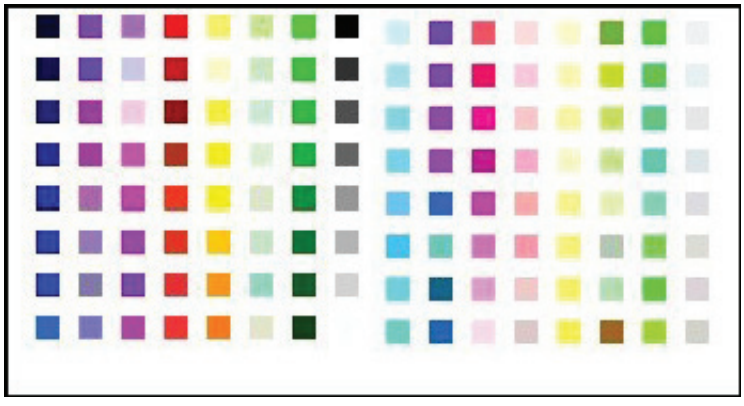


Fig. 4. Palette of 128 colors arrayed in arbitrary order.

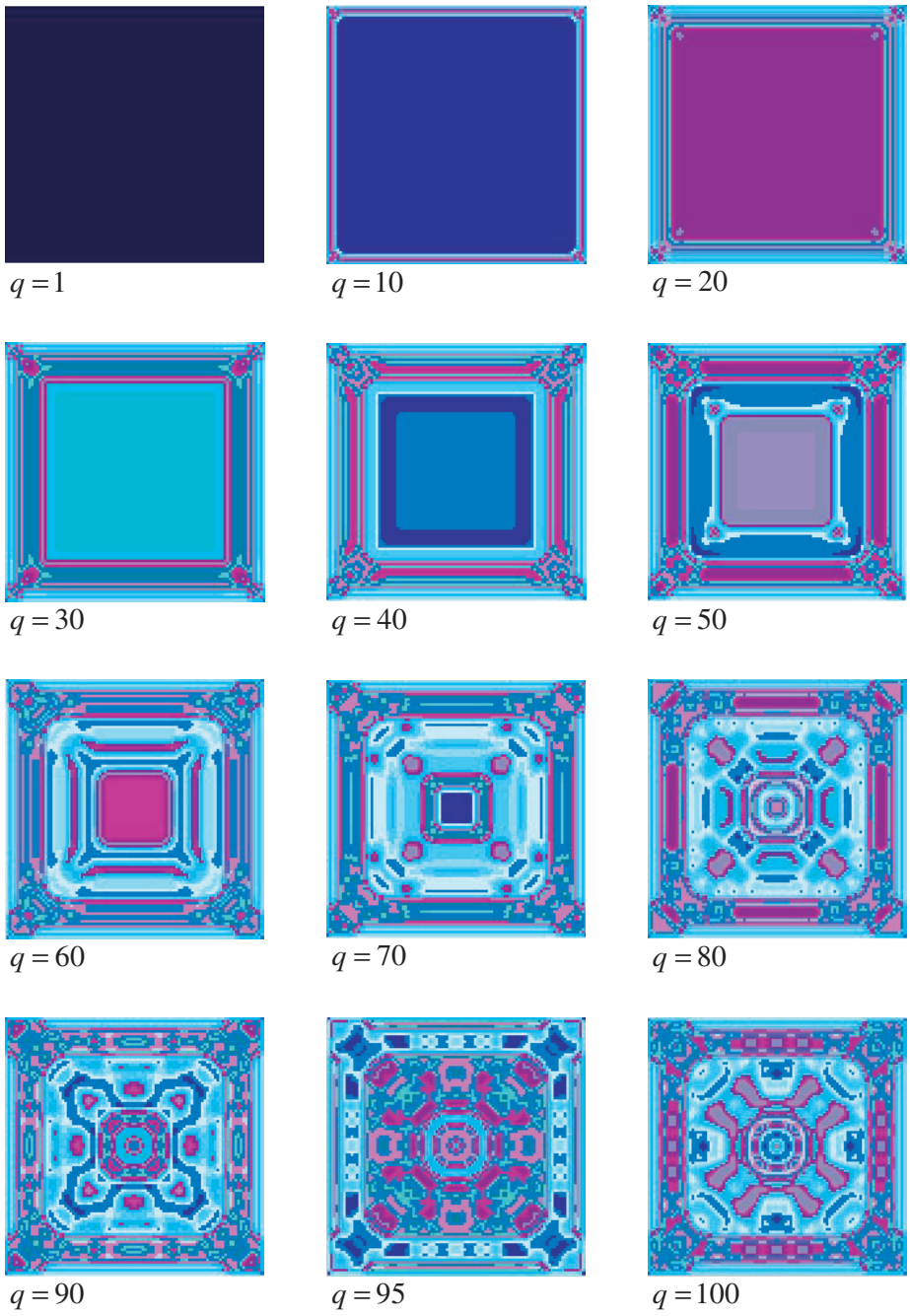


Fig. 5. Evolution of the image generated by Eqs. (3)–(5) starting from $q = 1$ for the parameters found with real coded genetic algorithm: $b = 0.2553$, $k_1 = 13.5293$, $k_2 = 4.7829$, $\alpha_1 = 6.5459$, $\alpha_2 = -5.5308$, $\alpha_3 = 2.5765$, $\beta_1 = -7.5475$, $\beta_2 = 5.7321$, $\beta_3 = -1.0422$.

average rate of RCGA convergence, we performed 130 runs. RCGA demonstrated convergence to the desired symmetrical image in all the cases. The number of generations produced by the RCGA procedure before convergence varied, as shown in Fig. 8. The average time required by a PC computer (Intel Pentium 4, 2.4 GHz) to find an image by RCGA from a random initial set of parameters was about 60 min on the average (∼50 RCGA generations). The fact that the RCGA procedure we used always converged during our trials confirms its robustness.

Image sequences generated for five different sets of parameters found by RCGA, and demonstrating three types of image evolution are shown in Figs. 9–13. Each sequence consists of twenty images. Figure 9 shows "convergence" of the sequence of different symmetrical images to one ("steady state")

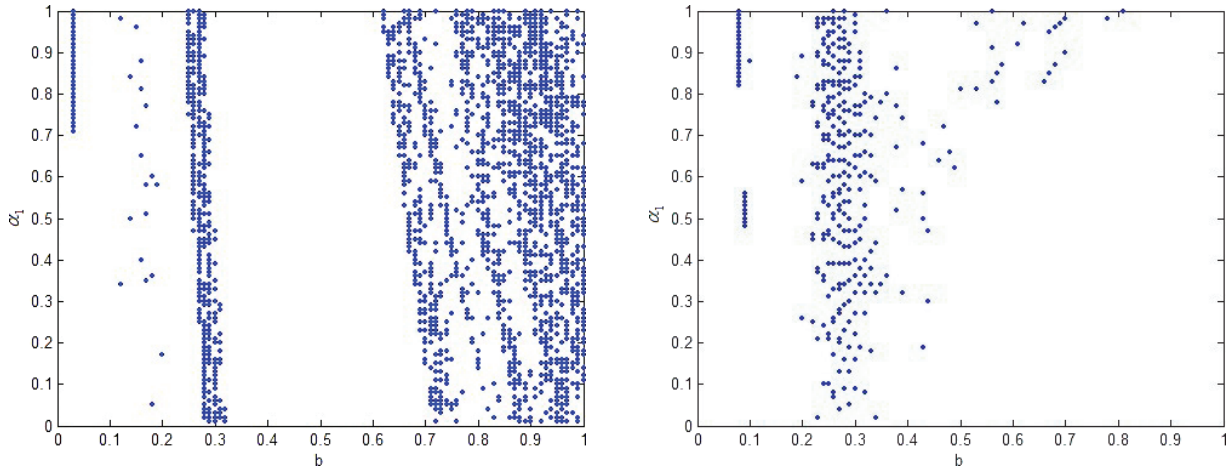


Fig. 6. Examples of two parameters (b, α_1) cross-section of the searching space in the interval $(0, 1]$. Blue points correspond to parameters that engendered symmetrical images.

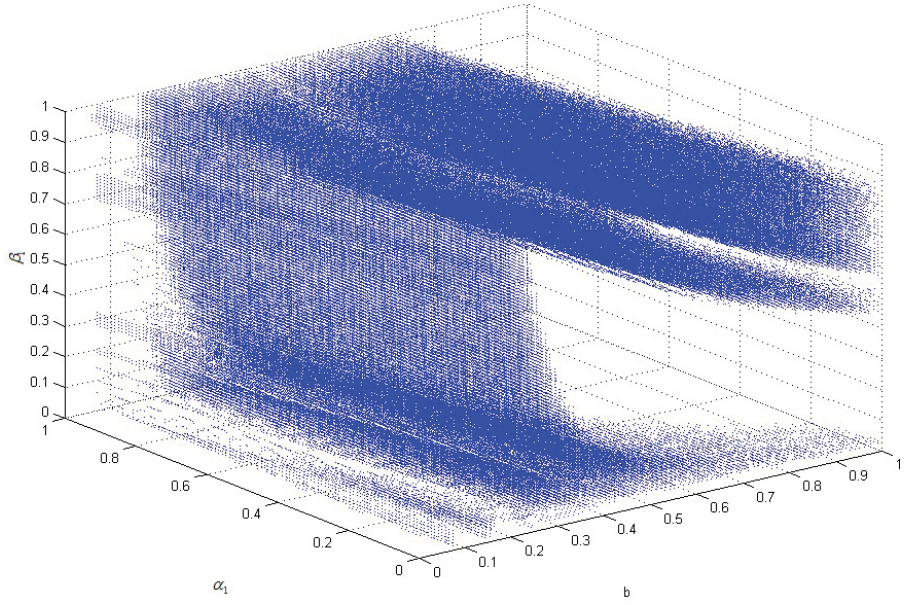


Fig. 7. Example of three parameters (b, α_1, β_1) cross-section of the searching space in the interval $(0, 1]$. Blue points correspond to parameters that engendered symmetrical images.

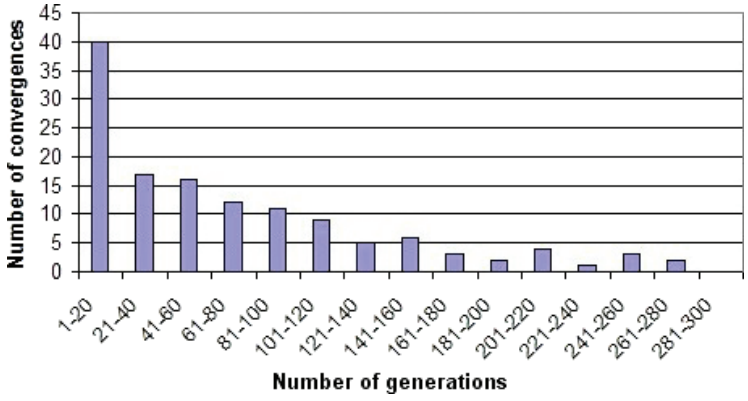


Fig. 8. Histogram of real coded genetic algorithm's convergence to symmetrical images (130 runs).

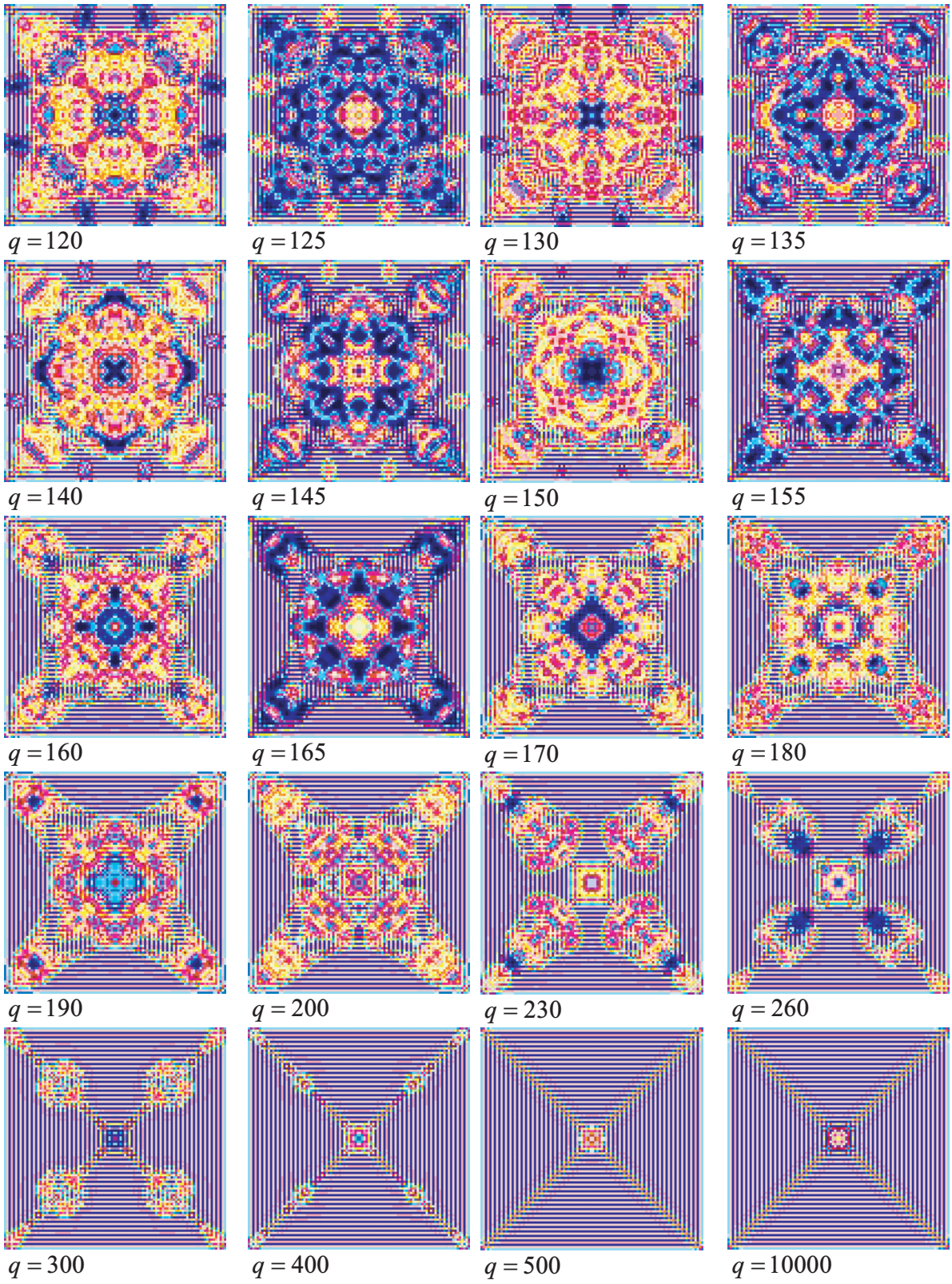


Fig. 9. Evolution of image (convergence to the "steady state") corresponding to the parameters found by real coded genetic algorithm: $b = 0.1400,\ k_1 = 9.2400,\ k_2 = 2.7700,\ \alpha_1 = 7.7200,\ \alpha_2 = 6.0000,\ \alpha_3 = -6.7800,\ \beta_1 = -6.0700,\ \beta_2 = 5.8500,\ \beta_3 = 1.0000$.

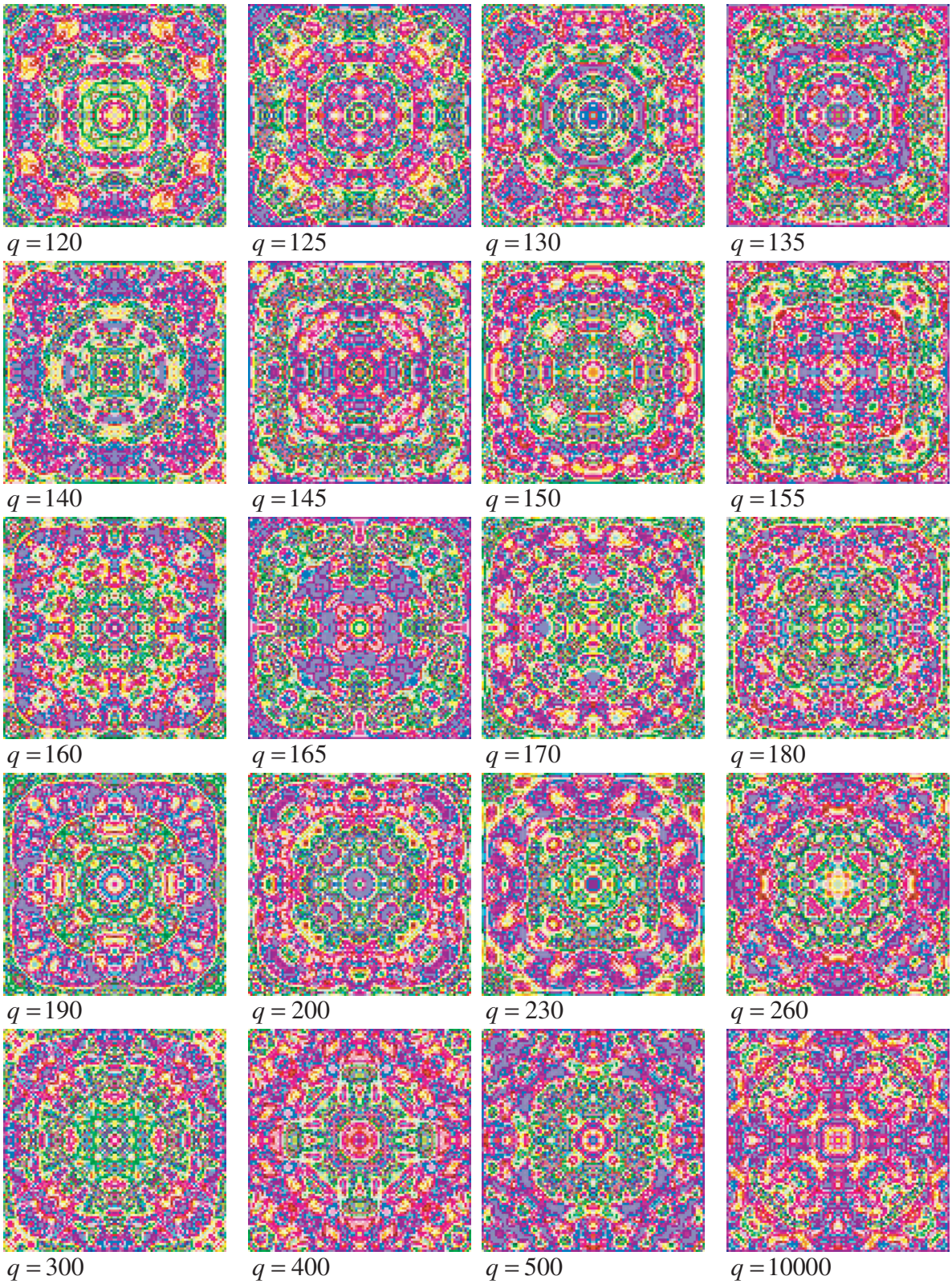


Fig. 10. Sequence of symmetrical images corresponding to the parameters found by real coded genetic algorithm: $b = 0.1500$, $k_1 = 3.7500$, $k_2 = 3.1300$, $\alpha_1 = 4.0500$, $\alpha_2 = -5.1000$, $\alpha_3 = 3.9600$, $\beta_1 = 2.6800$, $\beta_2 = -6.5500$, $\beta_3 = 6.2600$.

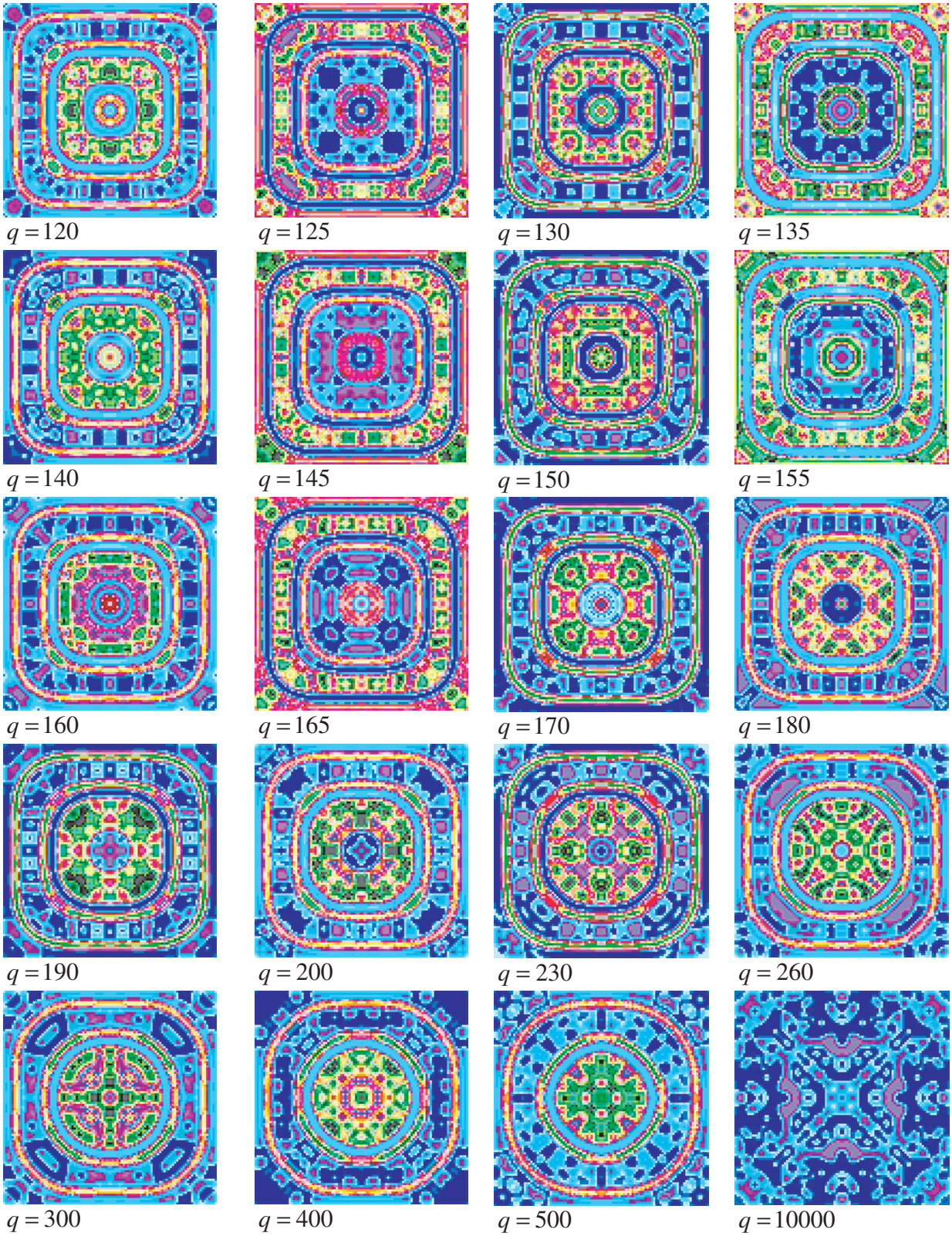


Fig. 11. Sequence of symmetrical images corresponding to the parameters found by real coded genetic algorithm: $b = 0.1900$, $k_1 = 9.0600$, $k_2 = 1.0300$, $\alpha_1 = 5.6500$, $\alpha_2 = -0.5600$, $\alpha_3 = 5.6400$, $\beta_1 = 0.2200$, $\beta_2 = -1.7500$, $\beta_3 = 5.5500$.

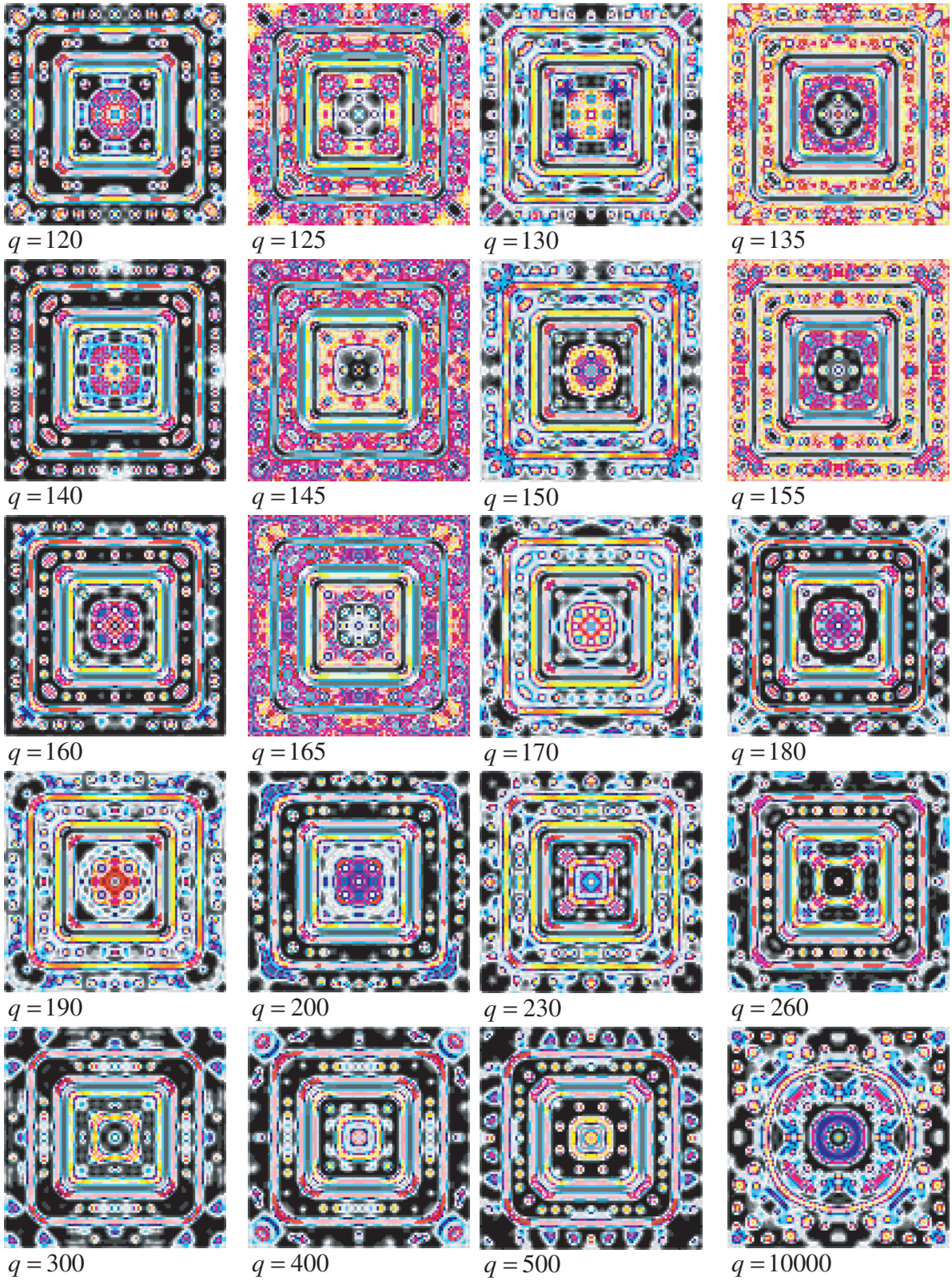


Fig. 12. Sequence of symmetrical images corresponding to the parameters found by real coded genetic algorithm: $b = 0.2100$, $k_1 = 6.0500$, $k_2 = 3.1400$, $\alpha_1 = 5.6000$, $\alpha_2 = -0.7300$, $\alpha_3 = 9.5800$, $\beta_1 = -9.6500$, $\beta_2 = 4.5600$, $\beta_3 = -0.8200$.

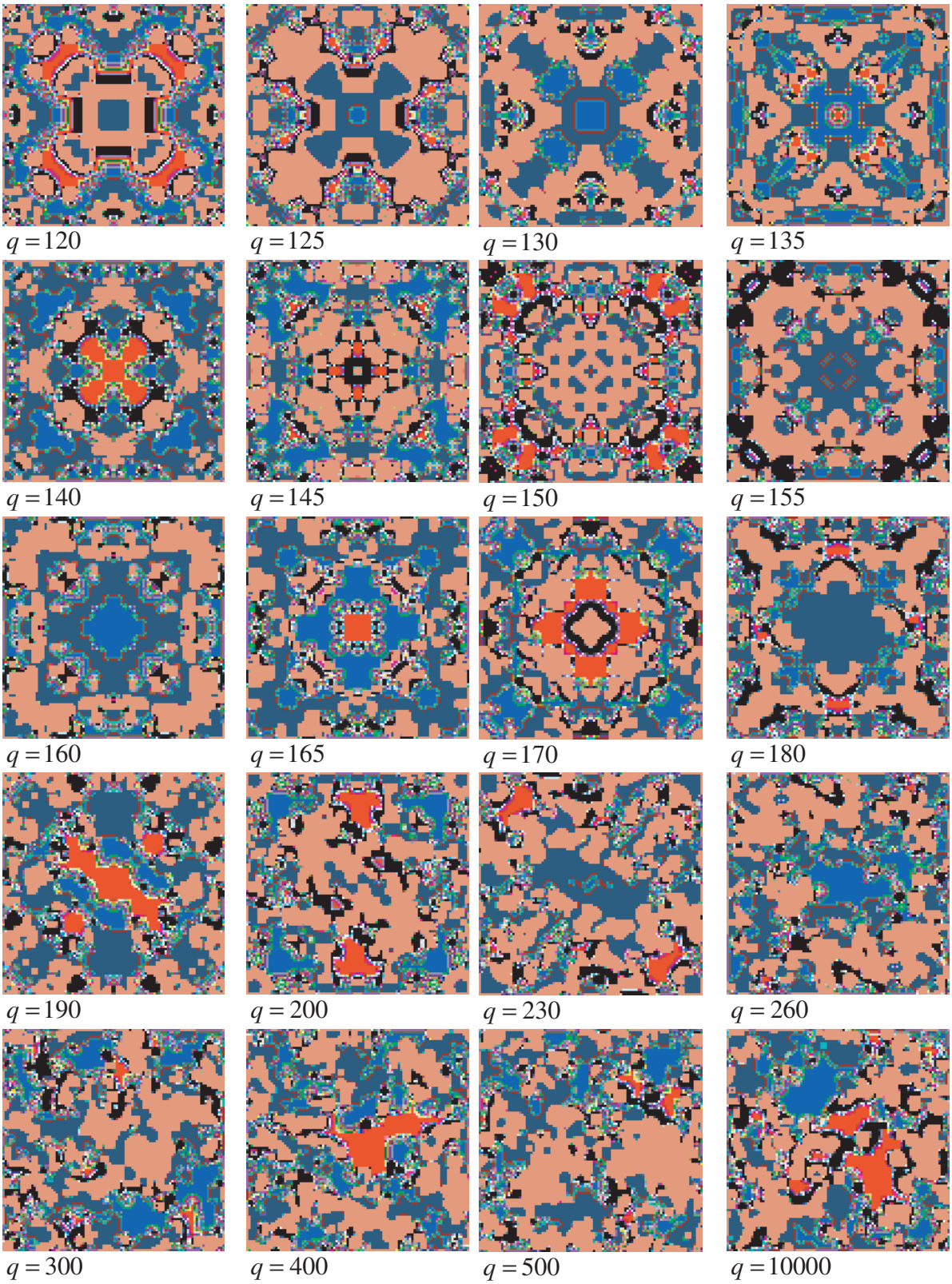


Fig. 13. Evolution of image (convergence to disordered images) corresponding to the parameters found by real coded genetic algorithm: $b = 3.6400$, $k_1 = 9.0100$, $k_2 = 4.0500$, $\alpha_1 = 2.8200$, $\alpha_2 = 3.4400$, $\alpha_3 = -1.4300$, $\beta_1 = -0.1000$, $\beta_2 = -5.3200$, $\beta_3 = 9.5400$.

image. Figures 10–12 show three image (pattern) sequences corresponding to the different sets of parameters, all of which remained symmetrical throughout their evolution ($Q = 10000$) while presented different geometrical forms. Figure 13 shows the evolution of initially symmetrical images into disordered patterns.

5. Conclusions

As was pointed out in the introduction, computer generated images have found applications in diverse areas of science and technology. In the present study we took a new type of mathematical model called chemical reactions discrete chaotic dynamics and used it to generate colored images of symmetrical forms. The concrete mathematical model presented here for illustration, namely the system of Eqs. (3)–(5), is just one possible initial hypothesis regarding mechanisms governing the agents' interactions.

We focused here exclusively on searching for symmetrical images using an RCGA special fitness function that we formulated. However, this fitness function could be adapted for an RCGA automatic search for parameters of CRDCD mathematical models corresponding to different types of patterns: rings, spirals and others (these patterns exist in CRDCD models [Gontar, 1997, 2004]).

The images generated here by CRDCD difference equations demonstrate their potential for generating complex symmetrical patterns similar to ornaments and mandalas designed by human beings [Jung, 1973].

The discrete chaotic dynamics of chemical reactions is proposed as a tool for exploring certain mechanisms of chemical transformation that could be related to real biochemical reactions taking place in the human brain. For example, ornaments or other artistic patterns may be presumed to appear in the human brain in the form of spatially distributed chemicals involved in the biochemical reactions before being "outputted" as concrete artistic patterns [Gontar, 2000].

What distinguishes the CRDCD mathematical model from other types of mathematical imaging methods and CA in particular is that the rules of interaction between the cells are based on concrete physicochemical laws and principles. In spite of the use by CRDCD of discrete space (lattice) and "discrete time", the states of each cell of the lattice are represented by continuous concentrations $X_i^{t_q}(R_x, R_y)$ and the corresponding parameters are given in terms of real numbers. Therefore each cell has continuous states with the given accuracy, which may naturally be represented by a colored palette of the corresponding resolution: $b/128, b/256, \ldots$.

It should be emphasized that from the standpoint of CRDCD principles, obtaining a symmetrical image does not presuppose symmetrical starting conditions: the symmetry results from the internal mathematical properties of the CRDCD model and reflects the mechanism of transformation and interaction of the agents, including "information exchange".

The proposed method will extend the current applications of mathematical imaging in the fields of computer graphics and artistic pattern design, evolutionary computation and parallel cellular machine design, human/robot visual interfaces, biofeedback systems, and artificial neural networks, as well as its application in the development of artificial brain systems [Gontar, 2003].

References

Deb, K. & Agrawal, R. B. [1995] "Simulated binary crossover for continuous search space," *Compl. Syst.* **9**, 115–148.

Eshelman, L. J. & Schaffler, J. D. [1993] "Real-coded genetic algorithms and interval schemata," *Foundations of Genetic Algorithms II*, ed. Whitley, D. (Morgan Kaufmann, San Mateo), pp. 187–202.

Goldberg, D. E. [1989] *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, MA).

Gontar, V. [1981] "New principle and mathematical model for formal chemical kinetics," *Russian J. Phys. Chem.* **55**, 2301–2308.

Gontar, V. & Il'in, A. V. [1991] "New dynamical model describing spatio-temporal behavior of chemical reactions," *Physica D* **52**, 528–531.

Gontar, V. [1997] "Theoretical foundation for the discrete chaotic dynamics of physicochemical systems: Chaos, self-organization, time and space in complex systems," *Discr. Dyn. Nature Soc.* **1**, 31–44.

Gontar, V. [2000] "Theoretical foundation of Jung's 'Mandala Symbolism' based on discrete chaotic dynamics of interacting neurons," *Discr. Dyn. Nature and Soc.* **5**, 19–28.

Gontar, V. [2003] "Artificial brain systems based on discrete chaotic dynamics of interacting intellectual agents," *Proc. Int. Conf. Computer Science and it's Applications*, National University, San-Diego, USA.

Gontar, V. [2004] "The dynamics of living and thinking systems, biological networks and the laws of physics," *Discr. Dyn. Nature Soc.* **8**, 101–111.

Ilachinski, A. [2001] *Cellular Automata*: *A Discrete Universe* (World Scientific, Singapore).

Jung, C. G. [1973] *Mandala Symbolism* (Princeton University Press).

Leblanc, B., Lutton, E. & Allouche, J.-P. [1998] *Inverse Problems for Finite Automata*: *A Solution Based on Genetic Algorithms*, Lecture Notes in Computer Science, Vol. 1363 (Springer-Verlag), pp. 157–166.

Mandelbrot, B. B. [1977] *Fractals*: *Form*, *Chance*, *and Dimension* (W. H. Freeman).

Mitchell, M., Crutchfield, J. P. & Hraber, P. T. [1994] "Evolving cellular automata to perform computations: Mechanisms and impediments," *Physica D* **75**, 361–391.

Nowak, M. A. & May, R. M. [1993] "The spatial dilemmas of evolution," *Int. J. Bifurcation and Chaos* **3**, 35–78.

Nowak, M. A., Bonhoeffer, S. & May, R. M. [1994] "More spatial games," *Int. J. Bifurcation and Chaos* **4**, 33–56.

Peitgen, H., Jurgens, H. & Saupe, D. [2004] *Chaos and Fractals* (Springer-Verlag).

Prusinkiewicz, P. & Hanan, J. [1989] *Lindenmayer Systems*, *Fractals*, *and Plants*, Lecture Notes in Biomathematics, Vol. 79 (Springer-Verlag).

Rodkaew, Y., Chuai-Aree, S., Siripant, S., Lursinsap, C. & Chongstitvatana, P. [2004] "Animating plant growth in L-system by parametric functional symbols," *Int. J. Intell. Syst.* **19**, 9–23.

Shonkwiler, R., Mendivil, F. & Deliu, A. [1991] "Genetic algorithms for 1-D fractal inverse problem," *Proc. Fourth Int. Conf. Genetic Algorithms,* eds. Belew, R. & Booker, L. (Morgan Kaufmann, San Mateo, CA), pp. 495–501.

Vences, L. & Rudomin, I. [1997] "Genetic algorithms for fractal image and image sequence compression," *Proc. Computacion Visual*, Universidad Nacional Autonoma de Mexico, pp. 35–44.

Wiederien, R. J. & Udwadia, F. E. [2004] "Global patterns from local interactions: A dynamical systems approach," *Int. J. Bifurcation and Chaos* **14**, 2555–2578.

Wolfram, S. [2002] *A New Kind of Science* (Wolfram Media).