Parallel Composition of Asynchronous Cellular Automata Simulating Reaction Diffusion Processes *

Olga Bandman

Supercomputer Software Department
ICM&MG, Siberian Branch, Russian Academy of Sciences
Pr. Lavrentieva, 6, Novosibirsk, 630090, Russia
E-mail: bandman@ssd.sscc.ru

Abstract. A method of constructing asynchronous cellular automata (ACA model) as a parallel composition of two interacting ACA is presented. The resulting ACA is intended to simulate a process with more than one species being involved in it. Two cases of such a composition are considered: (1) when one component of the composition is functioning independently affecting the evolution of the other, and (2) when both components evolute interacting at each iteration. The method is illustrated by simulation of two typical examples: (1) pattern formation process with inhibitor strength changing in time and space, and (2) prey-predator interactions proceeding to a stable spacial distribution.

1 Introduction

Physico-chemical spacial processes are asynchronous by nature: there is no natural clock synchronizing movements and interactions between molecules, atoms or other kind of particles. Meanwhile, the majority of mathematical models intended for simulation spacial dynamics in physics are synchronous. This contradiction exists both in traditional continuous mathematical modeling and in cellular automata (CA) simulation. In continuous case numerical solution of differential equations (PDE) are based on the time layers concept, performing computations of function values for the whole space for each time layer. As for CA simulation, the most popular CA-models in physics, such as based on Lattice-Gas idea [1], or classical CA [2] operate in synchronous mode. Three main reasons of synchronous CA predominance are as follows.

- The origin of the CA-modeling, i.e. the first von-Neumann's CA is a synchronous one.
- Synchronous mode corresponds on one hand to numerical analysis principle, on the other hand – to hardware mode of operation.

* Supported by (1)Presidium of Russian Academy of Sciences, Basic Research Program N 2 (2009), (2)Siberian Branch of Russian Academy of Sciences, SBRAS Interdisciplinary Project 32 (2009)

– Synchronous computation process is more deterministic, and, hence, easier for programming and parallel implementing.

Recently, the scientific interest has been moved towards nonlinear self organizing processes, which are hard to be simulated by PDE or cannot at all be expressed in continuous form. So, great attention is given to CA-modeling of complex phenomena such as prey-predator systems [4], pattern formation [3] and phase-transition [5] processes. Some of them (prey-predator, Belousov-Zhabotynsky, oreganator) describe reaction-diffusion asynchronous interactions between two or more species, and, hence, may be represented as composition of several CA. In [6] a method for two species reaction-diffusion processes is classified as local parallel composition of two CA, each simulating a self diffusion and mutual reaction [7]. Several attempts have been made to construct CA models of self organizing patterns formation using CA with weighted templates similar to Cellular Neural Networks (CNN)[8]. All of them deal with synchronous CA. The fact is in contradiction with the above thesis about asynchronicity of real life processes. To eliminate the contradiction parallel composition method of asynchronous CA has been investigated in detail, the results being presented in the paper.

The paper is organized as follows. The second section is devoted to formal statement of the problem. In the third section one way composition method is shown and illustrated by an example. In the fourth section the same is given for the two way composition. Some considerations about the results and future work are presented in Conclusion

2 Formal Problem Statement

Parallel composition suggests functioning of n interacting CA, each processing its own cellular array. Taking into account that the number of possible interactions in the composition exponentially increases with n, and for clearness of presentation, the composition $\aleph = \Upsilon(\aleph_1, \aleph_2)$ of two CA is further considered. Each component CA is determined by three sets $\aleph_k = \langle A_k, M_k, \Theta_k \rangle, k = 1, 2$, where A_k is a state alphabet, M_k - is a set of cell names (coordinates in finite discrete space), and Θ_k is a set of local operators. The alphabets A_1, A_2 may be different and of any type (Boolean, real, symbolic). Between $M_1 = \{(i,j)_1\}$, and $M_2 = \{(i,j)_2\}$, $i = 1, 2 \ldots, I$, there exists an one-to one correspondence $\xi : M_1 \to M_2$,

$$\begin{array}{ll} (i,j)_2 = \xi((i,j)_1), & \forall (i,j)_2 \in M_2, \\ (i,j)_1 = \xi^{-1}((i,j)_2), & \forall (i,j)_1 \in M_1. \end{array} \tag{1}$$

On M_1 and M_2 naming functions $\phi(i,j)$ are defined. They have the form of shifts, i.e. $\phi(i,j) = (i + \alpha, j + \beta)$, $\alpha, \beta \in \{-r, \ldots, 0, \ldots, r+1\}$, $r << |M|$.

A pair of cells in Ω_1 and in Ω_2, such that $(i,j)_1 = \xi^{-1}((i,j)_2)$ are further referred to as twins. When a cell of any component CA is meant, it is named simply as (i,j). Similarly, in all expressions valid for both CA components the bottom indices are removed.

Elementary entity of an ACA is a *cell* represented by a pair $(v,(i,j))$, where $v \in A$, $(i,j) \in M$. The set $\Omega = \{(v,(i,j))\}$ containing $|M|$ cells with different names form a *cellular array*. In a composed ACA $\aleph = \Upsilon(\aleph_1, \aleph_2)$ two cellular arrays are processed in parallel by application Θ_1 to Ω_1, and Θ_2 – to Ω_2. Each Θ_k is a set of *local transition rules*, i.e.

$$\Theta_k = \{\theta_{k1}(i,j), \ldots, \theta_{k2}(i,j), \ldots, \theta_{kn_k}(i,j)\}, \quad k = 1,2,$$

where each $\theta_{kl}(i,j)$ is expressed by a substitution [9] as follows.

$$\theta_{kl}(i,j) : S_{kl}(i,j) \star\ S''_{kl}(i,j) \rightarrow S'_{kl}(i,j), \quad k = 1,2, \quad l = 1, \ldots, n_k, \qquad (2)$$

where each term is a set of closely located cells referred to as a *local configuration*. The first local configuration $S_{kl}(i,j) \subseteq \Omega_k$

$$S_{kl}(i,j) = \{(v_m, \phi_m((i,j)_k) : m = 0,1,\ldots,m_{kl}\}, \quad k = 1,2, \quad l = 1,\ldots,n_k. \quad (3)$$

is called a *base* of $\theta_{kl}(i,j)$. The second local configuration is called a *contexts* of $\theta_{kl}(i,j)$. It contains cells from both cellular arrays $S''_{kl}(i,j) \subseteq \Omega_1 \cup \Omega_2$, so it may be partitioned into two subsets

$$S''_{kl}(i,j) = S''_{1l}((i,j)_1) \cup S''_{2l}((i,j)_2), \quad k = 1,2, \quad l = 1,\ldots,n_k, \qquad (4)$$

where $S''_{1l}((i,j)_1) \subseteq \Omega_1$ and $S''_{2l}((i,j)_2) \subseteq \Omega_2$, comprising cells determined by naming functions as follows.

$$S''_{k1}((i,j)_k) = \{(v''_p, \psi_p((i,j)_k) : p = 0,1,\ldots,p_{kl}\}, \quad k = 1,2, \quad l = 1,\ldots,n_k. \quad (5)$$

The right-hand side of (2) is a next-state local configuration

$$S'_{kl}(i,j) = \{(v'_m, \phi_m((i,j)_k) : m = 0,1,\ldots,m_{kl}\}, \quad k = 1,2, \quad l = 1,\ldots,n_k. \quad (6)$$

The states of the cells $(v'_m, \phi_m(i,j)) \in S'_{kl}(i,j)$ in (6) are values of *transition functions*

$$v'_m = f_m(V_{kl}), \qquad (7)$$

where V_{kl} is a set of states in the cells of $S_{kl}(i,j) \star\ S''_{kl}(i,j)$. The sets of cell names in the local configurations of (2) are referred to as *underlying templates* $T_{kl}(i,j)$, $T''_{kl}(i,j)$, and $T'_{kl}(i,j)$, $k = 1,2$, $l = 1,\ldots,n_k$. The relations between them are as follows.

$$T_{kl}(i,j) \cap T''_{kl}(i,j) = \emptyset, \qquad T_{kl}(i,j) = T'_{kl}(i,j), \quad k = 1,2,\ l = 1,\ldots,n_k. \quad (8)$$

An attempt of θ_{kl} application to a cell $(i,j)_k$ is successful if $S_{kl}(i,j) \cap S''_{kl}(i,j) \subseteq \Omega_k$, otherwise it fails and nothing happens. In case of a success $S_{kl}((i,j)_k)$ is replaced by $S'_{kl}((i,j)_k)$, the next states v'_m being substituted for v_m in all cells of $S_{kl}((i,j)_k)$.

An attractive property of asynchronicity is absence of simultaneous operations, that makes the computation free of dangerous collisions. So, as distinct to synchronous case, no constraint is imposed to intercell interactions.

By condition, let us consider that application of all $\theta_{1l} \in \Theta_1$ to all cells in Ω_1 and all $\theta_{2l} \in \Theta_2$ to all cells of Ω_2 comprise an *iteration*, let us number the iterations as $0, 1, \dots t, \dots T$, and assume that at each t global transitions $\Omega_k(t) \to \Omega_k(t+1)$, $k = 1, 2$ are performed. Although global transitions are not synchronized, such an assumption allows to define the evolution of the composed CA as a sequences $\Omega(0), \Omega(1), \dots, \Omega(t), \dots, \Omega(T)$, where $\Omega(t) = \Omega_1(t) \cup \Omega_2(t)$.

Functioning of the composed ACA proceeds in Ω_k, $k = 1, 2$, according to the following algorithm.

(1) a cell name $((i,j)_k) \in M_k$ is chosen randomly,

(2) a local transition rule $\theta_{kl} \in \Theta_k$ is chosen randomly,

(3) an attempt is made to apply θ_{kl} to $((i,j)_k)$,

(4) if application is successful, next states $v'_m = f_m(V_{kl})$ are computed for all $m = 1, \dots m_{kl}|$, and the next state local configuration $S'_{kl}((i,j)_k)$ is substituted for $S_{kl}(i,j)$ in Ω_k.

The above four steps are repeated in turn for $k = 1$ and $k = 2$ until the simulation terminates either when it achieves a stable state or when the the prescribed number of iterations T is exhausted.

3 One Way Parallel Composition of Asynchronous CA

In one-way composition $\aleph = \Upsilon(\aleph_1, \aleph_2)$ the roles of the CA components are different. Let $\aleph_1 = \langle A_1, M_1, \Theta_1 \rangle$ be the basic ACA, representing the process under investigation, and $\aleph_2 = \langle A_2, M_2, \Theta_2 \rangle$ – be the secondary ACA, which operates autonomously playing a controlling role. Then the transition rules $\theta_{1l} \in \Theta_1$, use the states from both cellular arrays, while $\theta_{2l} \in \Theta_2$ depend only from call states in its own Ω_2. Hence,

$$\begin{array}{ll} S''_{1l}(i,j) \in \Omega_1 \cup \Omega_2, & l = 1, \dots, l_1 \\ S''_{2l}(i,j) \in \Omega_2, & l = 1, \dots, l_2. \end{array} \tag{9}$$

Example 1. Pattern formation process on a surface with a heated area is simulated by a composition of two ACA $\aleph_v = \langle A_v, M_v, \theta_v \rangle$, and $\aleph_u = \langle A_u, M_u, \theta_u \rangle$. Both ACA have Boolean alphabet $A = \{0, 1\}$, their naming sets are isomorphic according to (1), $|M_v| = |M_u| = \{(i,j) : i, j = 0, \dots, 300\}$.

$\aleph_v$ is a diffusion ACA which is called in [10] as *naive diffusion*. The automaton operates independently simulating the propagation of heat from a hot area ($v = 1$) towards the cold center($v = 0$) and towards the borders through the warm ($\langle v \rangle = 0.5$) area (Fig.1). The local operator $\Theta_v = \{\theta_v\}$.

$$\begin{array}{l} \theta_v : \{(v_0, (i,j)), (v_1, (i-1,j)), (v_2, (i,j+1)), (v_3, (i+1,j)), (v_4, (i-1,j))\} \\ \to \quad \{(v'_0, (i,j)), (v'_1, (i-1,j)), (v'_2, (i,j+1)), (v'_3, (i+1,j)), (v'_4, (i-1,j))\} \end{array} \tag{10}$$

with the transition functions

$$\begin{array}{l} v'_0 = \quad v_k, \quad \text{if} \quad 0.25k < rand < 0.25(k+1), \\ v'_k = \begin{cases} v_0 & \text{if} \quad 0.25k < rand < 0.25(k+1), \\ v_k & \text{otherwise.} \end{cases} \end{array} \qquad k = 0, \dots, 3. \tag{11}$$

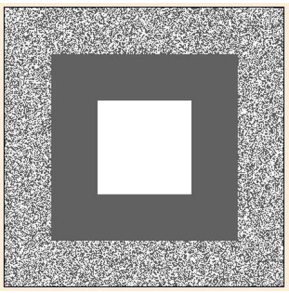

Fig. 1. Initial state of the context cellular array $\Omega_v(0)$.

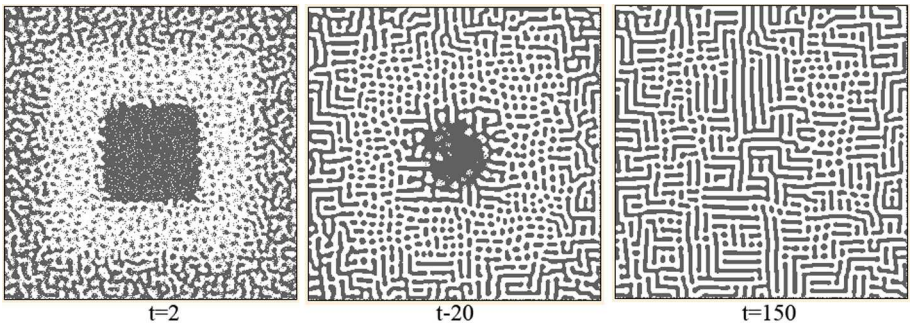

Fig. 2. Three snapshots of the evolution of a pattern formation ACA $\aleph_u$, whose inhibitor values are controlled by the heat propagation $\aleph_v$ with initial state $\Omega_v(0)$ is shown in Fig.1

The basic local operator $\aleph_u$, operating in $\Omega_u = \{u, (i,j) : i,j \in M_u\}$ simulating pattern formation process is as follows.

$$\theta_u : (u_0,(i,j)) \star \{(u_{i+g,j+h},(i+g,j+h)) : g,h = -3,\ldots,0,\ldots,3\} \rightarrow (u'_0,(i,j)), \tag{12}$$

$$u'_0 = \begin{cases} 1, & \text{if} \quad \sum_{g=-r}^{r} \sum_{h=-r}^{r} w_{gh} u_{i+g,j+h} > 0.1 \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where

$$w_{gh} = \begin{cases} 1, & \text{if} \quad |g| \leq 1 \;\&\; |h| \leq 1, \\ -\langle v_{i+g,j+h} \rangle & \text{otherwise.} \end{cases}$$

In Fig.2 three snapshots of pattern formation process in Ω_u are shown.

4 Two Way Parallel Composition of Asynchronous CA

In a two-way composition the component CA have equal rights. They simulate evolution of two interdependent processes. It means, that in Θ_1 and in Θ_2 there

are transition rules whose contexts contain cells from both cellular arrays, i.e. there exists $\theta_{1l} \in \Theta_1$ and $\theta_{2n} \in \Theta_2$, such that

$$\begin{aligned} S''_{1l}(i,j) \cap \Omega_2 \neq \emptyset \\ S''_{2n}(i,j) \cap \Omega_1 \neq \emptyset. \end{aligned} \tag{14}$$

Example 2 A two-dimensional prey-predatory problem is a well known one in mathematical ecology. It is usually represented by a system of two PDEs.

$$\begin{aligned} u_{tt} = d_u u_{xx} + F_u(u,v), \\ v_{tt} = d_v v_{xx} + F_v(u,v), \end{aligned} \tag{15}$$

where d_u, d_v are diffusion coefficients for two species, functions $F_u(u,v)$ and $F_v(u,v)$ are usually given in the form of polynomials of both variables. Let us interpret the problem in such a way: some predator (fish, foxes) eat prey (plankton, hares). If there is enough of food, predator density increases (predator propagates) with the probability depending on satiated predator density. In case of food shortage predator density diminishes (predator dye of hunger). Prey always attempts to propagate, when not being eaten dy the predator. Since predator is usually more agile than prey, its diffusion is essential, as for prey diffusion – it is hardly observable, ($d_v >> d_u$).

In the composed ACA $\aleph = \Upsilon(\aleph_v, \aleph_u)$, $\aleph_v = \langle A_v, M_v, \Theta_v \rangle$ stands for predator, and $\aleph_u = \langle A_u, M_u, \Theta_u \rangle$ stands for prey. Both alphabets are Boolean $A_v = A_u = A$, $M = M_v \cup M_u$, $M_v = \{(i,j)_v\}$ and $M_u = \{(i,j)_u\}$ being isomorphic.

Local operators for both ACA consist each of two transition rules: $\Theta_v = \{\theta_{v1}, \theta_{v2}\}$, $\Theta_u = \{\theta_{u1}, \theta_{u2}\}$. The first ones θ_{v1}, θ_{u1} are transition rules of naive diffusion given as (10,11). The second transition rules: θ_{v2} and θ_{u2}, represent the behavior of predator and prey and depend on both local densities $\langle (v,(i,j)_v) \rangle$ and $\langle (u,(i,j)_u) \rangle$ in twin cells. For the predator it is as follows.

$$\theta_{v2}: \ (v_0,(i,j)_v) \star (S''_v((i,j)_v) \cup S''_u(i,j)_u) \rightarrow (v'_0,(i,j)_v), \tag{16}$$

where $S''_v(i,j)$ and $S''_u(i,j)$ are local configurations with underlying templates $T_u = T_v = \{(i+a, j+b) : a,b = -r,\ldots,0,\ldots,r+1\}$, and the next state value

$$v'_0 = \begin{cases} 0, & \text{if } U(i,j) > V(i,j) \quad \& \quad (rand) < p_{v\rightarrow 0}, \\ 1, & \text{if } V(i,j) > U(i,j) \quad \& \quad (rand) < p_{v\rightarrow 1}, \end{cases} \tag{17}$$

$V(i,j)$ and $U(i,j)$ being sums of cell states in the context configurations $S''_v(i,j)$ and $S''_u(i,j)$, respectively.

The probabilities $p_{v\rightarrow 0}$ and $p_{v\rightarrow 1}$ are determined based on the following considerations. If $V(i,j) > U(i,j)$, a number of predators $(V(i,j) - U(i,j))$ in the local area $T((i,j)_v)$ has no food and may die. So, $V(i,j) - U(i,j)$ cell states in $T((i,j)_v)$ should be inverted into "zero", yielding in $p_{v\rightarrow 0}$ be equal to the ratio $(V(i,j) - U(i,j)/V(i,j)$ [11]. If $U(i,j) > V(i,j)$, then predator has plenty of food, it propagates increasing its density according to the propagation function of the form $F_v(\langle v(i,j) \rangle) = c * \langle v(i,j) \rangle (1 - \langle v(i,j) \rangle)$, where

$\langle v(i,j)\rangle = V(i,j)/|T(i,j)|$, c being a coefficient corresponding to the type of predator. So, the probabilities in (17) are as follows.

$$\begin{array}{ll} p_{v\to 0} = (V-U)/V, & \text{if} \quad V(i,j) > U(i,j), \\ p_{v\to 1} = 0.5\langle v(i,j)\rangle(1-\langle v(i,j)\rangle), & \text{if} \quad U(i,j) > V(i,j). \end{array} \tag{18}$$

The transition rule θ_{u2} looks like (17) differing in the probabilities $p_{u\to 0}$ and $p_{u\to 1}$, whose values are based on the following considerations. If $U(i,j) < V(i,j)$, prey is freely eaten. So, its density decreases with probability proportional to predatory density, Otherwise, if $U(i,j) > V(i,j)$, prey propagates with probability proportional to the number of the remainders $(U(i,j) - V(i,j))$.

$$\begin{array}{l} p_{u\to 0} = \langle v(i,j)\rangle), \quad \text{if} \quad V(i,j) > U(i,j), \\ p_{u\to 1} = 0.5(\langle u(i,j)\rangle - \langle v(i,j)\rangle)(1-(\langle u(i,j)\rangle - \langle v(i,j)\rangle)) \quad \text{if} \quad U(i,j) > V(i,j), \end{array} \tag{19}$$

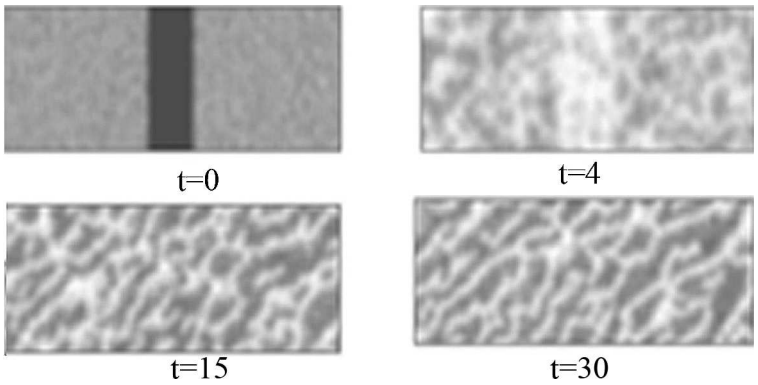

Fig. 3. Four snapshots of the evolution of the predator ACA

Let $M_k = \{((i,j)_k), i = 0,\ldots,399, j = 0,\ldots,799\}$. In the initial state prey is spread over the space with equal density, $\langle(u(i,j)_u)\rangle = 0.5$ for all $(u,(i,j)_u) \in \Omega_u(0)$. Predator has a maximum density $\langle(v(i,j)_v)\rangle = 1$ in the band $\{(i,j)_v : i = 0,\ldots,399, j = 369,\ldots,439\}$, and the same density $\langle(v(i,j)_v)\rangle = 0.5$ on the remaining area (Fig.3,t=0). Let $D_v = 10$ and $D_u = 1$ be ACA diffusion coefficients, that correspond to the difference of predator and prey agility. Then, the composed ACA operates according to the following iterative algorithm:

1) local operator θ_{v1} of asynchronous naive diffusion (10) is applied D_v times to Ω_v, each time at a randomly chosen cell $(v,(i,j)) \in \Omega_v$ (i,j),

2) local operator θ_{v1} (10) of asynchronous naive diffusion is applied D_u times to Ω_u in the similar way,

3) a cell $(i,j) \in M_u$ is randomly chosen and θ_{v2} (17,18) is applied to it.

4) a cell $(i,j) \in M_u$ is randomly chosen and θ_{u2} (17,19) is applied to it.

In Fig.3 four snapshots of the evolution of the predator component are shown. In the initial state there is a dense band of predator. It is seen that the evolution

comes to to the stable state very fast: when $t > 20$ the changes of the patterns are hardly seen along the borders of predator bunches.

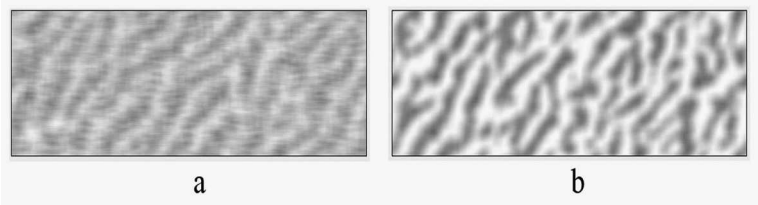

Fig. 4. Stable state of a composed prey-predatory CA evolution (t=30): a) averaged global configuration $\Omega_v(30)$(predator), b) Boolean global configuration $\Omega_u(30)$ (prey)

The important thing is that the composed CA is very stable. All initial configurations having any area (sometimes very small) with nonzero species densities, tend to the same stable pattern. In Fig.4 stable states of both components are shown, obtained with equal initial homogeneous distribution $\langle v(i,j)\rangle = \langle u\rangle = 0.2$. The stable states are identical to those of the case shown in Fig.3 and many other program runs with different initial conditions.

5 Conclusion

Parallel composition method of asynchronous cellular automata is presented. Two particular cases are considered in detail and experimentally tested: one-way composition allows to introduce a controlling effect in the evolution of a process under simulation; two-way composition allows to construct ACA, simulating self organizing behavior. The results allow to hope that the approach might be helpful to create some techniques for constructing ACA when some behavioral properties of its evolution are given.

References

1. Frish, U., d'Humieres, D., Hasslacher, B., Lallemand, P., Pomeau, Y., Rivet, J.P.: Lattice-Gas hydrodynamics in two and three dimensions. Complex Systems **1**. 649–707. (1987)
2. Wolfram, S.: A new kind of science. – Wolfram Media Inc., Champaign, Ill., USA (2002)
3. Deutsch A., Dorman S.: Cellular Automata Modeling of Biological Pattern Forfation. – Birkhäuser (2005)
4. Cataneo G., Dennunzio A., Farina F. :A Full Cellular Automaton to Simulate Predatory-Prey Systems. LNSC, 4273 (2006) pp.446-451
5. Kinzel W.: Phase Transition of Cellular Automata Zeitschrift Für Physik, Condenced Matter, 58, N3 (1985) pp.229-224.

6. Bandman O.: Cellular Automarta Composition Techniques for Spatial Automata Simulation. - In: Simulating Complex Systems by Cellular Automata (Eds.: J.Kroc, P.M.A.Sloot), Berlin: Springer (2010).
7. Nishyama A, Tokohiro T. Construction of isotropic cellular automaton for a real equation by means of random walk. arXiv: 1001.0505v1 [nlin.CG] 4 Jan 2010.
8. Chua L. CNN: a paradigm of complexity.– Singapore: World Scientiofic (2002).
9. Achasova S., Bandman O., Markova V., Piskunov S.: Parallel Substitution Algorithm. Theory and Application – Singapore: World Scientific (1994)
10. Toffolli T.,Margolus,N.: Cellular Automata Machine.– USA: MIT Press (1987)
11. Bandman O.: Simulating Spatial Dynamics by Pribabilistic Cellular Automata. Cellular Automata, 5th Int.Conf ACRI 2002. LNCS 2493 (S.Bandini, B.Chopard, M.Tomassini,eds.) Springer (2002) pp.10-16