

CELLULAR AUTOMATA: STRUCTURES AND SOME APPLICATIONS

MARCIN BURZYŃSKI

Institute of Environmental Mechanics and Applied Computer Science, Bydgoszcz University
e-mail: marcinb@ito.pl

WALDEMAR CUDNY

Institute of Environmental Mechanics and Applied Computer Science, Bydgoszcz University
Institute of Fundamental Technological Research, IPPT PAN, Warsaw
e-mail: wcudny@ippt.gov.pl

WITOLD KOSIŃSKI

Research Center, Department of Artificial Intelligence
Polish-Japanese Institute of Information Technology, Warsaw
e-mail: wkos@pwstk.edu.pl

A new approach to the modelling of various nature phenomena such as predator and prey ecological system, heat transport, spreading of oil slick and traffic flow is introduced. Cellular automata (CA) are discrete dynamical systems whose behaviour is completely specified in terms of simple local relations. They are mathematical models of spatially distributed processes; however they can lead to an appropriate simulation of complex dynamic processes. Applications to heat transfer and problems of environmental simulations are done. A discrete automaton model with fuzzy rules to simulate one-way traffic flow is also described. Results of simulations are consistent with phenomena observed in reality. It gives a base to propose the cellular automata tool as an option in modelling and solving problems of complex (and some times, not completely known) nature.

Key words: cellular automata, traffic flow, fuzzy rules

1. Introduction

Differential equations form a mathematical basis for most current models of natural systems. Cellular automata (CA) may be considered as an alternative

approach, and in some respect, they can play a role of complementary models of nature. Whereas ordinary differential equations are suitable for systems with a small number of degrees of freedom, evolving in a continuous manner, cellular automata describe the behaviour of systems with large numbers of discrete degrees of freedom (Wolfram, 1984b).

Modelling complex physical phenomena through computer simulation has become a common tool for understanding the natural world. Realistic models based on physical laws generally result in large systems of non-linear integral or partial-differential equations (PDEs). Such a modelling approach has many advantages, e.g. an ability to reproduce in quantitative details results of experiments. However, there are some disadvantages: it is difficult to extrapolate the behaviour of "realistic" models to different experimental and natural conditions. Those models sometimes obscure basic physical principles underlying the modelled phenomena. Complex models lead often to formidable numerical problems. One technique for simplifying these often numerically intractable systems is to mimic the physical laws by a series of simple rules that are easy to compute (Ermentrout and Edelstein-Keshet, 1993). This kind approach is called Cellular Automata. The fundamental observation which forms the basis of the CA approach is: do not describe complex systems with the help of complex equations, but let the complexity emerge by interaction of simple individuals following simple rules.

In the paper, this new approach to the modelling of various nature phenomena such as predator and prey ecological system, heat transport, spreading of oil slick is applied. A discrete automaton model with fuzzy rules to simulate one-way traffic flow is also described. Results of simulations are consistent with phenomena observed in reality. It gives a base to propose the cellular automata tool as an option in modelling and solving problems of complex (and some times, not completely known) nature. All presented applications of CA have been developed by the present authors.

1.1. History of cellular automata

The first cellular automata was due to John von Neumann (1903-1957) and Stanisław Ulam (1909-1984) during their common works in Los Alamos. The 80's was a decade of these models that based upon parallel computation methods. Stephen Wolfram exhaustively explored one-dimensional cellular automata and introduced for the first time a classification of CA depending on the system behaviour, known as Wolfram Classes (Wolfram, 1984b). In 1970 American mathematician John Conway developed the *Game of Life* which popularized CA among the scientists. *Life* is a cellular automaton operating on

a two-dimensional grid of cells, each of which can have only two states: dead or alive (coded as value 0 or 1). The rules of *Life* are also very simple:

- If a cell has exactly two living neighbours, it stays as it is. Live cells stay alive and dead cells stay dead.
- If a cell has exactly three living neighbors a dead cell becomes alive and a live cells stay alive.
- If a cell has any other number of living neighbours a dead cell stays dead and a live cell becomes a dead cell.

These very simple rules can lead to extremely complex behaviour giving the right arrangement of live cells. *Life* can be considered as a model of bacteria population dynamics.

2. Definitions

There are many non formal cellular automata definitions introduced by authors. We mention two slightly different definitions that reflect the origin of CA and which mimic the same features of cellular automata.

Definition 1. Cellular automata are dynamical systems in which space and time are discrete. A cellular automaton consists of a regular grid of cells, each of which can be in a finite number of k possible states, updated synchronously in discrete time steps according to local, identical interaction rules. The state of a cell is determined by the previous states of surrounding neighborhood of the cell.

Definition 2. A cellular automaton is an array (linear array) of cells that interact with their neighbours. The array can take on any number of dimension, starting from one dimensional string of cells. Each cell has its set of states. By receiving the input from connected cells (or generally, a message from its neighbours) a cell uses its own sets of rules to determine what its reaction should be. This reaction is manifested as a change of own state and can also be a trigger to send out a message to the neighbours. The message is passed to other selected cells which makes them act likewise.

From the definitions above, Wolfram (1984b) has specified five fundamental characteristics of cellular automata:

1. They consist of a discrete lattice of sites.
2. They evolve in discrete time steps.
3. Each site attains a finite set of possible values.
4. The value of each site evolves according to the same rules.
5. The rules for the evolution of a site depend only on a local neighbourhood of sites around it.

Hence we can write main cellular automata features:

- *Parallelism* means that individual cell updates are performed independently of each other; it means that all of the updates are being done at once.
- *Locality* means that when a cell is updated, its state is based solely on the previous state of the cell and of its nearest neighbours.
- *Homogeneity* means that each cell is updated according to the same rules. Typically, states of the cell and its nearest neighbours are related by some algebraic (or logic) formulas.

These characteristics determine the structure and behaviour of particular automata, and are discussed in a further part of the paper.

2.1. Cell space

First, we define the cell space for 2D cellular automata

$$L = \{(i, j) | i, j \in \mathbb{N} : 0 \leq i < n, 0 \leq j < m\} \quad (2.1)$$

where i, j are the number of the column/row of the lattice with the maximum extent of n columns and m rows. Equation (2.1) can be extended to n dimensions.

2.2. Local neighbourhood

The neighborhood is a local space in which cells can interact. The nearest neighbourhood is described by a set of cells that neighbour the given cell in position (i, j) . Equation (2.2)₁ defines the so-called von Neumann neighbourhood of radius r . The Moore neighbourhood of radius r is described by (2.2)₂. Other neighbourhoods can be defined as long as they are uniform and finite

$$N_{i,j} = \{(k, l) \in L : |k - i| + |l - j| \leq r\} \quad (2.2)$$

$$N_{i,j} = \{(k, l) \in L : |k - i| \leq r \text{ and } |l - j| \leq r\}$$

One should point out that the extension of the neighbourhood leads sometimes to some improvement of the isotropy property, and therefore is often used when natural phenomena are to be modelled. However, a large neighbourhood is usually very inefficient to simulate.

2.3. Lattice geometry

The definition of cellular automata requires the lattice to be regular. We consider different possibilities for 1D, 2D or 3D models. For 1D automata there is only one possibility: a linear array of cells. In 2D there are three regular lattices, namely triangular, square, and hexagonal ones:

Triangular lattice – small numbers of nearest neighbours

Square lattice – the most popular lattice, easy to implement

Hexagonal lattice – the lowest anisotropy which makes simulations more natural (in some cases it is absolutely necessary to use this lattice to model the phenomena correctly (Wolfram, 1984a; Frisch *et al.*, 1985).

In three dimensions there are many possible regular lattices but the cubic lattice is mostly used.

2.4. Boundary conditions

In the formal definition of cellular automata, it is usually required that the lattice is infinite. For computability and complexity points of view, this is reasonable and necessary, but it is impossible to implement and simulate an infinite lattice on a computer. Therefore, we have to introduce some boundary conditions. Moreover, the problem we would like to simulate has some boundaries itself. Mostly, we deal with three types of boundary conditions: periodic, reflective and fixed valued.

In the *periodic boundary condition* one sticks together opposite sides of the lattice. It causes a loop for a 1D lattice and leads to the topology of a torus for a two-dimensional lattice. Periodic boundary conditions are the closest to simulate an infinite lattice, and for this reason they are very often used.

The *reflective boundary condition* is obtained by reflecting the lattice at the boundary. It is a suitable approach where the system to be simulated has a boundary but the values of physical variables are not fixed (i.e. diffusive system).

Fixed-value boundary conditions are obtained by simply prescribing a fixed value for the cell on the boundary. This boundary condition corresponds to Dirichlet's boundary condition for PDEs. Note, that all three types of boundary conditions can be combined, except for the case when one side of the

lattice has a periodic boundary condition. Then the opposite side must also have periodic boundary condition as well.

It is possible to introduce one more class when a modification of transition rules for some group of cells is allowed. Then an *absorbing boundary condition* can be defined for which cells at the borders have no neighbours beyond the limits of the lattice.

2.5. Initial condition

In most cases initial conditions determine the evolution of a considered model. Initial conditions are arbitrarily constructed on the base of the problem considered or are generated randomly. Many cellular automata rules conserve some quantities, for example the total number of particles, the total momentum, energy. This must be taken into account hence while generating initial conditions.

2.6. Transition rules

The most important aspect of cellular automata is the problem of *transition rules*. The rules determine the evolution of a model. They depend somehow on the lattice geometry, the neighborhood, and the state set. There are two approaches towards defining the transition rules. One is based on intuitive knowledge of an expert and leads to a set of *if-then rules* the outcome of which is the final transition rule for the considered cell and its neighbourhood. For example, they can be written as follows: IF *conditions* N_1, \dots, N_n hold THEN *state of cell is* S_m . This can be modified by introducing the probability into transition rules: IF *conditions* N_1, \dots, N_n hold AND *probability of* $P = l$ THEN *state of cell is* S_m . The probabilistic rules have their importance because natural systems are noisy.

The second approach is based on an implicit(functional type) specification in which the detailed transition rule has to be evaluated from some mathematical formulas (Toffoli, 1984; Abelson *et al.* [1]; Chopard and Droz, 1993). This approach is common when modelling physical phenomena (Vichniac, 1984). In the further part it will be shown how to develop the transition rules from differential equations.

2.7. Time evolution

The need for a proper time evolution of the modelled system is obvious. However, sometimes there is a lack of knowledge about the system, e.g. we do not know which parts of the system perform first, which perform later. In

result, one has to make a compromise and introduce such a time evolution approach which does not favour any part of the system nor a process. Then we say about *synchronous time evolution* which is original and comes from the definition of cellular automata. The change of cell states is due to the previous state and previous state of its neighbour

$$a_{n,t=m+1} = f(a_{n,t=m}, a \in N_{t=m}) \quad (2.3)$$

where $a_{n,t}$ is the state of the cell in time t , N is the neighbourhood of the considered cell. Note that there is a question that needs to be addressed, namely the cellular automata's synchronicity. Hogeweg (1988) and others claimed that the simultaneous updating of all cells is at odds with the localness of interaction that is one of the strength of cellular automata. It was shown by Packard and Wolfram (1985) that some synchronous automaton results are artifact patterns which do not reflect real phenomena. By changing the definition of a CA we allow for asynchronous updating of cells. From the numerical point of view it is very similar to the Monte Carlo simulation. The asynchronous time evolution can dramatically alter the behaviour of CA (Hogeweg, 1988). In some situations the conclusion is that the interesting structures seen in the evolution of cellular automata are, in fact, artifacts of the synchronous updating. On the other hand some 1D asynchronous automata can produce patterns independently of the update approach (Kanada, 1994). During implementation of synchronous cellular automata one should also take into consideration the so-called *collision problem*. It means that when we apply a rule that causes an immediate movement of objects in the automaton, there is always a possibility that two or more objects could occupy the same cell in the same time. Then we have to implement special complicated procedures that split the automaton rule into several sub-steps to avoid such a situation.

Note that if one needs to introduce the time domain into asynchronous cellular automata, one iteration is done when some number of drawings is done. Mostly, the number of drawings is equal to the number of cells. We assume that in one iteration approximately all cells are drawn, which is similar to the synchronous approach.

3. Application of cellular automata

In the forthcoming sections, some applications of cellular automata to biological systems, heat transfer and environmental simulation problems will be

made. A discrete automaton model with fuzzy rules to simulate one-way traffic flow will also be described.

3.1. Cellular automata model of predator and prey system

Many of the most interesting dynamics in the biological world have to do with interactions between species. Mathematical models which incorporate such interactions are required if we want to simulate their dynamics. One of the first models which incorporated the interactions between the predator and prey was proposed in 1925 by American biophysicist Alfred Lotka (Lotka, 1925), and independently by Italian mathematician Vito Volterra (Volterra, 1926). The Lotka-Volterra model is based on differential equations

$$\begin{aligned}\frac{dx}{dt} &= a_1x + a_2xy \\ \frac{dy}{dt} &= -b_1y + b_2xy\end{aligned}\tag{3.1}$$

where x and y denote the space averaged number of prey and predator population, respectively; a_1 is the natural growth rate of the prey in the absence of the predator; a_2 is the death per encounter of the prey due to predator; b_1 is the natural death of the predator in the absence of food (prey); b_2 is the efficiency of turning the predated prey into a predator.

The numerical solution (Runge Kutty 4th order) of equations (3.1) shows three characteristics of the Lotka-Volterra system: periodic dynamics, amplitudes interdependence and phase shifts.

Burzyński (2001), the first author of this paper, developed a cellular automata model of the predator-prey system. The model used a rectangular two-dimensional lattice with the periodic boundary condition. Each site can be: empty, prey or predator. The time evolution was synchronous and the Moore neighbourhood of radius 1 was used. The rules mimic some experts knowledge and apply the same set of state for each cell:

- Cell is empty:
 - if $N_l \neq 0$ and $N_r \neq 0$ then S_{t+1} is a predator or prey under probability proportional to N_l and N_r
 - if $N_l = 0$ and $N_r \neq 0$ then S_{t+1} is a prey under probability proportional to N_r
 - if $N_l = 0$ and $N_r = 0$ then S_{t+1} remains an empty cell

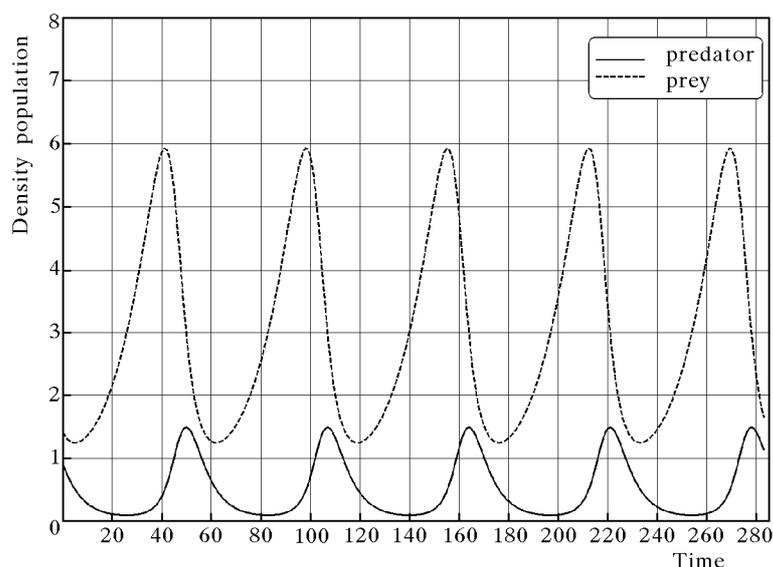


Fig. 1. Numerical solution of Lotka-Volterra model

- Cell is occupied by a predator, then S_{t+1} becomes empty under constant probability P_l
- Cell is occupied by a prey, then S_{t+1} becomes empty under constant probability P_r if $N_r > 0$.

The following denote: N_l – number of predators in the neighbourhood; N_r – number of preys in the neighbourhood; P_l – probability of predator's death; P_r – probability of hunting success (can depend on N_l); l – initial number of predators; r – initial number of preys.

One can notice that the parameters P_l , P_r are indirectly equivalent to the parameters b_1 , a_2 from equations (3.1). These parameters define the behaviour of the system, and one can obtain similar dynamics to that observed in the Lotka-Volterra model.

The cellular automata model of the predator and prey system has some advantages. We obtained an interesting spatial distribution of the populations density (Fig. 3), which is unknown in the Lotka-Volterra differential equation model. In Droz and Pekalski (2001) a cluster formation process was shown which can be observed in the natural environment. The population dynamics observed from simulations of cellular automata leads to a bigger diversity of the system behaviour (Burzyński, 2001; Lipowski, 1999; Lipowski and Lipowska, 2000).

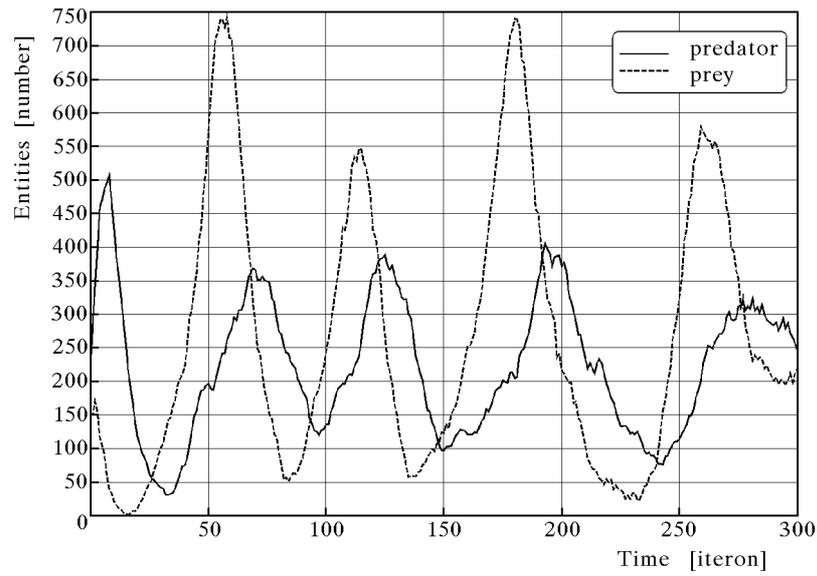


Fig. 2. Population dynamics in cellular automata model of predator-prey system

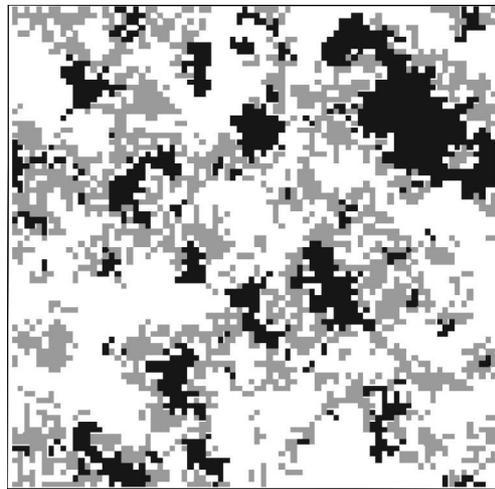


Fig. 3. Spatial distribution of predator and prey population

3.2. Heat transfer problem

In dealing with heat equation (3.2), different methods are used to solve it. We are referring here to the finite difference method in which the solution is searched at a discrete number of points which are grid points or sites of a

cellular automaton (i.e. cells). The governing equation of heat transfer in the plane state is

$$\frac{\partial T(x, y, t)}{\partial t} = a \left(\frac{\partial^2 T(x, y, t)}{\partial x^2} + \frac{\partial^2 T(x, y, t)}{\partial y^2} + \frac{\dot{q}_V(x, y, t)}{\lambda} \right) \quad (3.2)$$

for the temperature field $T(x, y, t)$, where λ is the heat conduction coefficient, c_p – heat capacity, ρ – mass density, $a = \lambda/(c_p\rho)$, and \dot{q}_V denotes the volumetric heat supply source.

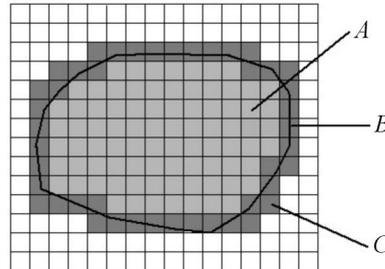


Fig. 4. Domain and its discretization

Let a finite domain \mathcal{C} bounded by a closed curve will be substituted by a set of inner cells \mathcal{A} and boundary cells \mathcal{B} , see Fig. 4. In the *finite difference method*, Eq. (3.2) is approximated by a system of recursive equations

$$\frac{T_S^{n+1} - T_S^n}{\Delta t} = \frac{a}{h^2} (T_A^n - 2T_S^n + T_C^n + T_B^n - 2T_S^n + T_D^n) + a \frac{\dot{q}_V(x, y, t)}{\lambda} \quad (3.3)$$

where T_K^n with $K = S, A, B, C, D$ denotes the temperature value at the time step n at the grid node K . Here $K = S$ denotes the current node, while $K = A, B, C, D$ are the neighbouring ones. We have assumed the finite difference approximation of the second order derivative y'' of a typical function y to be $y'' = (y_{-1} - 2y + y_{+1})/h^2$, where h is the increment of independent variable (grid size) and the elementary grid size $\Delta x = \Delta y = h$. If no heat supply source is present, i.e. $\dot{q}_V = 0$, then after some transformations of (3.3) we get

$$T_S^{n+1} = T_S^n (1 - 4F_0) + F_0 (T_A^n + T_B^n + T_C^n + T_D^n) \quad (3.4)$$

where $F_0 = a\Delta t/(\Delta x)^2$ is the Fourier coefficient which characterizes the rate of heat transfer inside the body.

To solve Eq. (3.2), initial and boundary conditions are needed. In each discretization method one needs to determine which cells (elements of the discretized domain) are treated as inner elements, which are treated as boundary elements, and how the boundary conditions are realized.

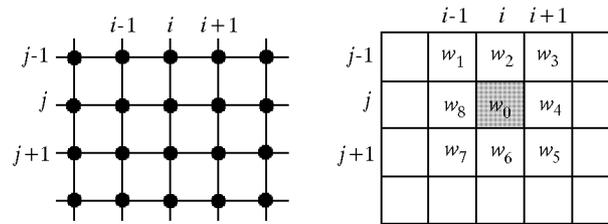


Fig. 5. Numeration of cells in the grid

In the further part, we have assumed a rectangular (square) domain with the boundary shown in Fig. 7a. The domain is divided into (16×16) smaller squares. The boundary is represented by a single layer of cells for which a constant state is assumed. It means that the Dirichlet boundary condition (for temperature) has been assumed. As the initial condition we assume a constant (independent of the position) temperature equal to the reference temperature. Since we are working with the increment of temperature above the reference level (the latter measured as the absolute temperature), as an independent variable, the initial condition means that initially the temperature (increment) is equal to zero. On two neighbouring sides of the boundary the temperature is constant and equal to 50 degrees, while on the remaining sides it is a linear distribution which starts with the value 50 and ends with 16 degrees. To satisfy the numerical stability condition, the following inequality must hold

$$1 - 4F_0 \geq 0 \quad (3.5)$$

In further investigations, the value of F_0 will be assumed arbitrary, satisfying constraint (3.5). The numerical scheme represented by Eq. (3.4) can be regarded as a convolution of the elements of the grid domain with a particular set, called a filter. This analogy comes from processes known in 2D signal analysis. Hence, we can write

$$\begin{aligned} T_{i,j}^{n+1} = & T_{i,j}^n w_0 + T_{i-1,j-1}^n w_1 + T_{i,j-1}^n w_2 + T_{i+1,j-1}^n w_3 + T_{i-1,j}^n w_8 + \\ & + T_{i+1,j}^n w_4 + T_{i-1,j+1}^n w_7 + T_{i,j+1}^n w_6 + T_{i+1,j+1}^n w_5 \end{aligned} \quad (3.6)$$

Solving Eq. (3.4) in an iterative way is the same as the realization of the convolution of the grid domain with the filter presented on Fig. 6, as it is shown in Eq. (3.6), where

$$\begin{aligned}
 w_0 &= 1 - 4F_0 & w_1 = w_3 = w_5 = w_7 &= 0 \\
 w_2 &= w_4 = w_6 = w_8 & &= F_0
 \end{aligned}
 \tag{3.7}$$

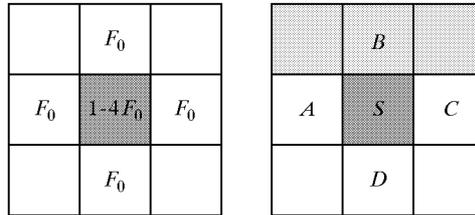


Fig. 6. General filter, filter in (3.7), boundary cell B

In the cells neighbouring the boundary the same filter is used in the convolution operation, since the isothermal boundary condition has been assumed. It is the place to write explicitly the solution algorithm:

1. determine the domain of the solution as a discrete set of cells
2. determine the boundary cells
3. determine the initial state of the cells
4. determine the evolution rules according to which the state will change, cf. Eq. (3.6)
5. apply the evolution rules m times sufficient to fulfill the stop condition (e.g. realize the iterative process by Eq. (3.6)).

In the iterative formula, we have proposed above, the Dirichlet boundary condition is assumed and no heat source is included. If a uniform heat source \dot{q}_V is present, then the central element in the filter will be modified to the form

$$w_0 = (1 - 4F_0) + \frac{\dot{q}_V}{\lambda} (\Delta x)^2
 \tag{3.8}$$

Moreover, in the case of the Neumann boundary condition $\dot{q}_s = \text{const}$, the boundary cells in the filter (Fig. 6) will assume the value $\dot{q}_s \Delta x / \lambda$ in the place of F_0 . In Nagórski (2001) one can find different filters for a number of boundary conditions. That approach, however, is no more a cellular automata one.

Let us consider stability condition (3.5) and the form of the filter value w_i , $i = 1, \dots, 8$ from Eq. (3.7). Notice that the following equality is satisfied

$$\sum_i w_i = 1
 \tag{3.9}$$

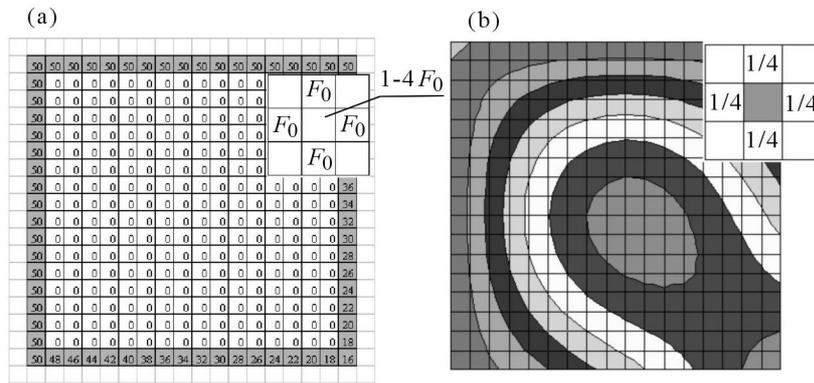


Fig. 7. Grid and filter (3.7) for classical finite difference scheme (3.4)

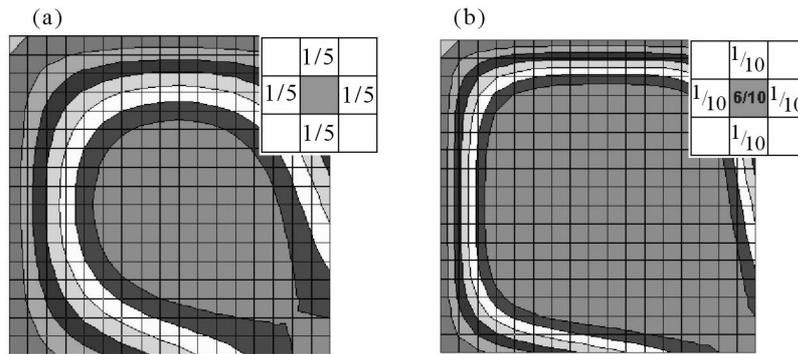


Fig. 8. Simulation results with different filters

In the course of implementation of the filter, (3.7), satisfying the stability condition with $F_0 = 1/4$, the realization time was the shortest. The questions are: whether it is possible to obtain reasonable results of numerical implementation if constraint (3.9) is only assumed? What is the relation of such a solution scheme to stability condition (3.5)? Is it possible to find an apparent value of F_0 corresponding to this scheme? In Fig.7 some simulation results are presented corresponding to solutions to Eq. (3.6) where the iterative expression (3.6) has been applied. The boundary and initial conditions are the same in all cases. For better comparisons and presentations, the temperature interval $[15, 50]$, to which values of the temperature at the boundary belong, has been divided into 7 subregions with different gray levels. Referring to the temperature at the boundary, the regions inside of the domain with the same temperature values are easy to recognize. Results in Fig.7 and Fig.8

after 30 iterations and with the use of (3.7) are rather compatible with our expectations.

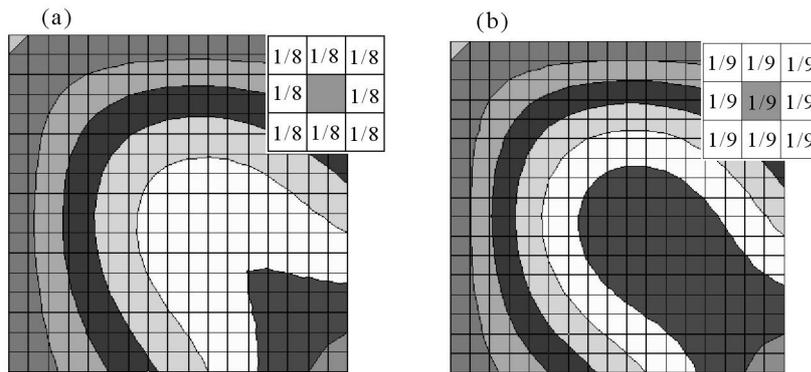


Fig. 9. Simulation results with different filters

In Fig. 10, however, different iteration time has been obtained for different F_0 . What we can see is that the smaller value of F_0 the shorter time step is, and more iteration steps must be performed in order to reach the same temperature level at a given point (cell). Figure 9 presents simulation results for Eq. (3.6) but with a more general form of the filter than that in (3.7), however with condition (3.9). Notice that the number of iteration steps to reach the same temperature value as in Fig. 8 is 3 times smaller than in Fig. 7; similar acceleration is obtained with the filter applied in calculations reported in Fig. 9.

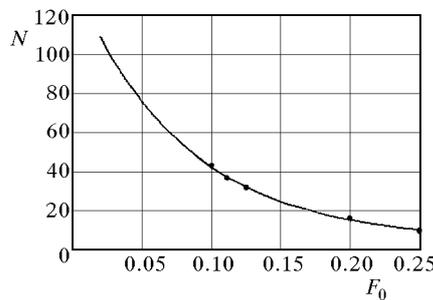


Fig. 10. Number of iteration for different values of F_0

The well-posedness of the construction of the filters from Fig. 9a can be explained with the help of the mean value theorem and two filters from Fig. 7 in which the latter is turned by $\pi/4$ with respect to the former. Moreover,

for the central cell it corresponds to the apparent value $F_0^p = 3/8$, which is greater than $1/4$, for which the shortest calculation time has been proved with classical iterative scheme (3.4). It can be proved that all filters satisfying Eq. (3.9) lead to the same final state. What is important for our analysis is that the solutions to differential equations can yield some numerical schemes in which more complex filters can be obtained, i.e. 5×5 . In the case of the heat conduction problem independent of the shape of the boundary and the type of boundary conditions, we can design evolution rules in the form of appropriate filters satisfying condition (3.9), which lead to a better approximation of the Laplace operator (cf. Collatz, 1960) in Eq. (3.2) than the scheme presented by Eq. (3.3).

In our opinion the designing of filters, having in mind a phenomenon of our interest, can lead to appropriate analysis of the phenomenon and then to its reasonable description. It can be fruitful even in the case when the phenomena cannot be described by means of known (one can say, classical, with the use of PDEs) methods. What we should do is to use alternative methods on the assumption that the phenomenon is similar to other known and already described with the help of appropriate tools, phenomena. The cellular automata approach can be of great help, in our opinion, especially because of its simplicity and demonstrative character.

3.3. Movement of an oil slick

In this section, an ecological catastrophe will be discussed in which an oil slick spreads over the ocean and moves toward seacoast. We assume that the oil slick emerges in a vicinity of two islands. To describe this in a grid, three domains are determined: the oil and two islands. Then initial and boundary conditions are assumed for the diffusion equation, similar to the heat conduction equation. The islands coasts are assumed in the form of a fixed boundary condition for the temperature (or a fixed pollution concentration). We assume, however, that the water around the islands moves and influences (together with the wind) movement of the oil slick. We assume that the movement directions are different in different regions.

The relevance of obtained simulation results strongly depends on the assumed hypothesis, which contains the initial value of the state, model of oil spreading, assumed boundary conditions – the most interesting, in our opinion. We assume that the evolution rules of the automaton is similar to that applied in the heat equation, Eq. (3.6), in which absorption at a typical boun-

dary cell B , neighbouring the cell T_S , appears, and the evolution at a typical grid cell T_S , are governed by

$$B^i = \sum_{j=1}^i \alpha T_S^j \quad (3.10)$$

$$T_S^{i+1} = \frac{1}{4}[T_A^i + T_B^i + T_C^i + (1 - \alpha)T_S^i]$$

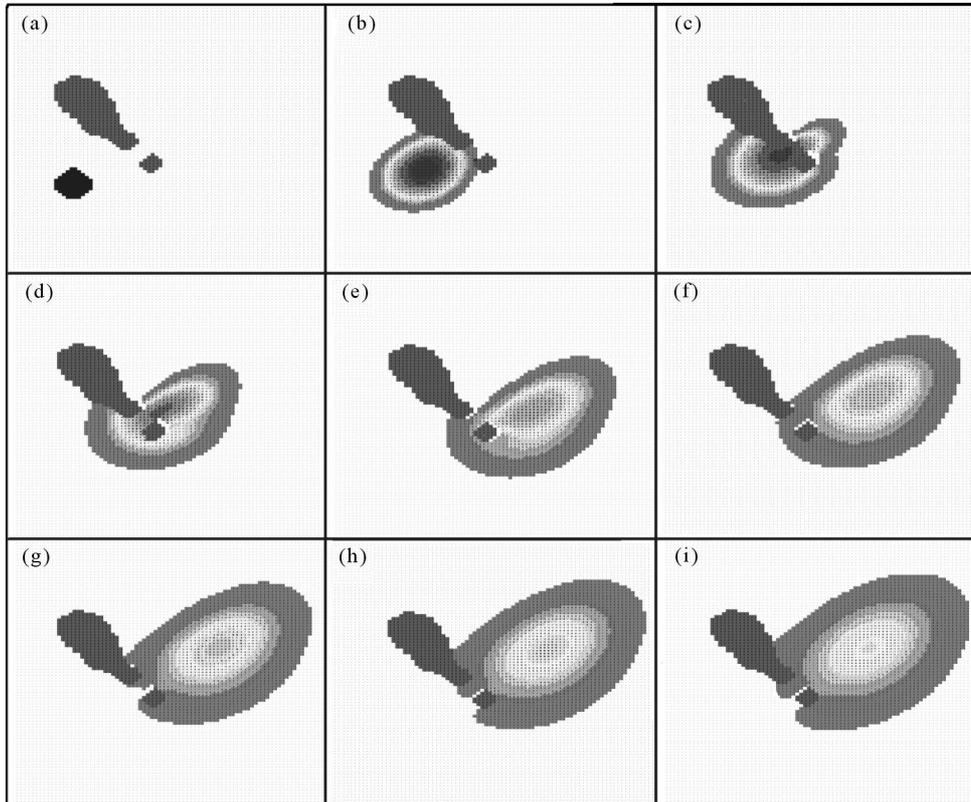


Fig. 11. Movement of oil slick: (a) initial state, (b) state after 20 and more iterations

Results of the numerical implementation of the simulation are presented in Fig. 11, where the subsequent figures present states after subsequent 20 iterations. In Fig. 11a the initial state is shown. It is important in cellular automata models that we have a possibility to determine transition rules when two (or more) states are known at the distance dt . Then, one can approximate states in further instants without complete knowledge about the process that

takes place and governs the system evolution. On the other hand, from the knowledge of the filter one can try to determine a type of equations that govern those changes.

3.4. Modelling of traffic flow

The modelling of the traffic transport problem is very interesting and important for its dynamics and serious dramatic consequences in real life. We focus on the cellular automata approach instead of classical ones, e.g. the kinetic theory of vehicular traffic (Prigogine and Herman, 1971) or fluid-dynamics approach (Lighthill and Whitman, 1995). We want to exploit one important property of cellular automata, namely the lack of stability, which means that very small changes in transition rules or states can have very dramatic consequences (Kulakowski, 2000). In reality, a single driver can dramatically change the flow dynamics as well.

The used model is similar to that introduced by Nagel and Schreckenberger (1992), Schadschneider and Schreckenberg (1993), however, some crucial details are changed. We use rules based on a fuzzy controller instead of deterministic and probabilistic rules, and some driver's individual characteristics are included in our model.

The computational model of the traffic flow is defined on a 1D Cartesian lattice of the linear size L and with periodic boundary conditions. Each cell can be empty or occupied by a single car. Each cell is described by the following set of parameters: $F = 0$ or 1 , which determine whether the cell is empty or occupied by a car. $V = 0$ or 1 upto 7 , is the speed of the car. $V_{max} = 0$ or 1 upto 7 is the maximum speed the car can develop. N is the individual number of a car (we need the car to be numbered to perform some statistic). The neighbourhood is asymmetric: $N = \{a_{i+1}, \dots, a_{a+7}\}$, it reflects the space ahead observed by the driver. Cells are updated synchronously by each car translation based on the distance S proportional to the calculated velocity V ; due to the unit time interval, in most cases $S = V$.

To mimic the process of making decision by a driver on the road, we decided to use fuzzy rules based on fuzzy control instead of deterministic or probabilistic rules. Fuzzy approach gives us a possibility to describe local behaviour of the system, mainly the driver's behaviour. It can be introduced by the expert knowledge, which includes fuzzy (non-crisp) variable values, i.e. *high speed*, *low speed*, *do brake immediately*, and so on. We used Mamdani Controller (Piegat, 1999) to control the output variable A , which is the acceleration deceleration (negative acceleration) of a car. There are two input variables: the distance D to the nearest car and the speed V . All three variables are regarded as lingu-

istic variables defined by fuzzy sets (Łachwa, 2001). Parameters of triangular membership functions of the sets A , D , V (Łachwa, 2001) and the rule matrix have been chosen arbitrarily to avoid a car crash (see Burzyński *et al.* (2003) for details). In our previous research, we decided to apply the same fuzzy set parameters to each car. We are going to distinguish characteristics of drivers by using independent sets of fuzzy parameters for each driver in our future works.

As it was said, our model based on an 1D automaton with periodic boundary conditions subject to simulations was in a long closed loop and on a single lane. The total number N of cars in the automaton cannot change during one simulation, thus it is possible to define a constant system density. This is a drastic simplification of the problem, because a constant number of cars is not usually possible in the real traffic flow. Thus, to reflect real conditions, the number of cars should be measured and averaged in a given site over period of time.

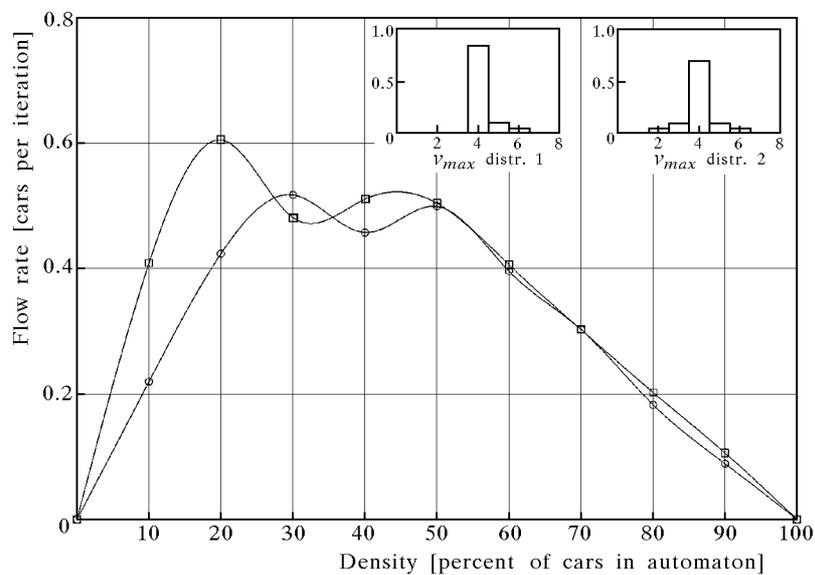


Fig. 12. Traffic flow simulation – fundamental diagram

The main object of our study was to obtain the fundamental diagram (see Fig. 12). The distribution of the maximum speed V_{max} can change the fundamental diagram. As it is shown in Fig. 12, there are two maxima instead of one. It is evident that there is no common idea for the same maximum speed among the drivers. A number of drivers moving fast or idlers can change dramatically

the traffic flow dynamics. Therefore, we noticed the need to examine relations between the preferable maximum speed and the fundamental diagram.

The results of carried out simulations are consistent with Schreckenberger's model (Nagel and Schreckenberger, 1992) and real life measurements (Hall *et al.*, 1986).

References

1. ABELSON, ADAMS, COORE, HANSON, NAGPAL, SUSSMAN, *Gray Scott Model of reaction diffusion*,
<http://www.swiss.ai.mit.edu/projects/amorphous/GrayScott>
2. BURZYŃSKI M., 2001, Symulowanie układu drapieżca-ofiara za pomocą automatu komórkowego, M.Sc. Thesis, Akademia Bydgoska im. Kazimierza Wielkiego, Wydział Matematyki, Techniki i Nauk Przyrodniczych, Bydgoszcz
3. BURZYŃSKI M., CUDNY W., KOSIŃSKI W., 2003, Traffic flow simulation-cellular automata with fuzzy rules approach, In: *Advances in Soft Computing, Proc. of the Sixth Int. Conference on Neural Network and Soft Computing*, Zakopane, Poland, June 11-15, 2002, Rutkowski L., Kacprzyk J., (eds.), Physica-Verlag, Heidelberg, 808-813
4. CHOPARD B., DROZ M., 1993, Study of the A+B to C reaction-diffusion process, *International Journal of Modern Physics, C*, 4, 209-215
5. COLLATZ L., 1960, *Metody numeryczne rozwiązywania równań różniczkowych*, PWN, Warszawa
6. DROZ M., PEKALSKI A., 2001, Coexistence in a predator-prey system, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, **63**, 5, 051909-1-051909-6
7. ERMENTROUT G.B., EDELSTEIN-KESHET L., 1993, Cellular automata approaches to biological modeling, *J. Theor. Biol.*, **160**, 97-133
8. FERBER J., 1999, *Multi-Agent Systems: an Introduction to Distributed Artificial Intelligence*, Addison-Wesley, New York
9. FRISCH U., HASSLACHER B., POMEU Y., 1985, A lattice gas automaton for the Navier-Stokes equation, *Los Alamos Preprint LA-UR-85-3503*
10. HALL F.L., ALLEN B.L., GUNTER M.A., 1986, Empirical analysis of freeway flow - density relationships, *Transportation Research*, **A20**, 197ff
11. HOGEWEG P., 1988, Cellular automata as a paradigm for ecological modeling, *Applied Mathematics and Computation*, **27**, 81-100

12. KANADA Y., 1994, The effect of randomness in asynchronous 1D cellular automata, *Artificial Live IV*, Poster Session
13. KARAFYLLIDIS I., 1997, A model for the prediction of oil slick movement and spreading using cellular automata, *Environment International*, **23**, 6, 839-850
14. KUŁAKOWSKI K., 2000, *Automaty komórkowe, "Jak"*, Kraków
15. LAWSON B.G., PARKE S., 2000, Asynchronous time evolution in a artificial society mode, *JASSS*, **3**, 1
16. LIDTHILL M.J., WHITMAN G.B., 1995, On kinematics waves: a theory of traffic flow on long crowded roads, *Proc. R. Soc. Lond.*, **A 229**, 1178, 317-345
17. LIPOWSKI A., 1999, Oscillatory behavior in a lattice prey-predator system, *Physical Review*, **E, 60**, 5, 5179-5184
18. LIPOWSKI A., LIPOWSKA D., 2000, Nonequilibrium phase transition in a lattice prey-predator system, *Physica*, **A 276**, 456
19. LOTKA A.J., 1925, *Elements of Physical Biology*, Baltimore, Williams and Wilkins Co.
20. ŁACHWA A., 2001, *Rozmyty świat zbiorów, liczb, relacji, faktów, reguł i decyzji*, EXIT, Warszawa
21. NAGEL K., SCHRECKENBERGER M., 1992, A cellular automaton model for freeway traffic, *Journal de Physique*, **1, 2**, 2221-2229
22. NAGÓRSKI Z., 2001, *Modelowanie przewodnictwa ciepła za pomocą arkuszką kalkulacyjnego*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa
23. PACKARD N.H., WOLFRAM S., 1985, Two-dimensional cellular automata, *Journal of Statistical Physics*, **38**, 901-946
24. PIEGAT A., 1999, *Modelowanie i sterowanie rozmyte*, EXIT, Warszawa
25. PRIGOGINE I., HERMAN R., 1971, *Kinetic Theory of Vehicular Traffic*, American Elsevier Publishing Co., New York, 17-54
26. SCHADSCHNEIDER A., SCHRECKENBERG M., 1993, Cellular automaton models and traffic flow, *J. Phys. A: Math. Gen.*, **26**, L679-l683
27. SCHATTEN A., *Cellular Automata Tutorial*, <http://www.ifs.tuwien.ac.at/~aschatt/info/ca/ca.html>
28. TOFFOLI T., 1984, Cellular automata as an alternative (rather than an approximation of) differential equations in modeling physics, *Physica*, **D10**, 117-127
29. TOFFOLI T., MARGOLUS N., 1987, *Cellular Automata Machines: A New Environment for Modeling*, Cambridge, MA: Mit Press
30. VICHNIAC, 1984, Simulating physics with cellular automata, *Physica* **D**, 10, 96-115

31. VOLTERRA V., 1926, Variazioni e fluttuazioni del numero d'individui in specie animali conviventi, *Mem. R. Accad. Naz. dei Lincei*, **6**, 2, 31
32. WOLFRAM S., 1984a, Cellular automata as models of complexity, *Nature*, **311**, 419-424
33. WOLFRAM S., 1984b, Universality and complexity in cellular automata, *Physica*, **10D**, 1-35
34. WOLFRAM S., 1994, *Cellular Automata and Complexity*, Reading, MA: Addison Wesley,
35. WOLFRAM S., 2002, *A New Kind of Science*, Wolfram Media Inc.
36. WOLFRAM S., FALL, 1983, Cellular automata, *Los Alamos Science*, **9**, 2-21

Automaty komórkowe: struktura i pewne zastosowania

Streszczenie

W pracy zaprezentowano metodę modelowania układów i zjawisk obserwowanych w przyrodzie, takich jak dynamika systemu ekologicznego drapieżnik-ofiara, przewodzenie ciepła, rozprzestrzenianie się plamy ropy naftowej po wycieku na wodzie czy ruch strumienia pojazdów na drodze miejskiej. Metodę oparto na tzw. automatach komórkowych, które są układami dyskretnymi o zachowaniach ściśle zdeterminowanych prostymi relacjami o charakterze lokalnym. Automaty komórkowe to matematyczne modele procesów przestrzennych, mogące z powodzeniem opisywać złożone zjawiska dynamiczne. W pracy przedstawiono aplikację do zagadnienia przewodzenia ciepła oraz kilku symulacji środowiskowych. Przedstawiono także model automatowy z regułami rozmytymi opisujący jednokierunkowy ruch pojazdów na drodze. Wyniki symulacji okazały się zgodne z obserwacjami rzeczywistych układów. Zachęcające rezultaty badań skłaniają do postrzegania automatów komórkowych jako efektywnej opcji w modelowaniu i rozwiązywaniu problemów o złożonej (czasem nie całkiem rozpoznanej) naturze.

Manuscript received May 25, 2003; accepted for print June 14, 2003