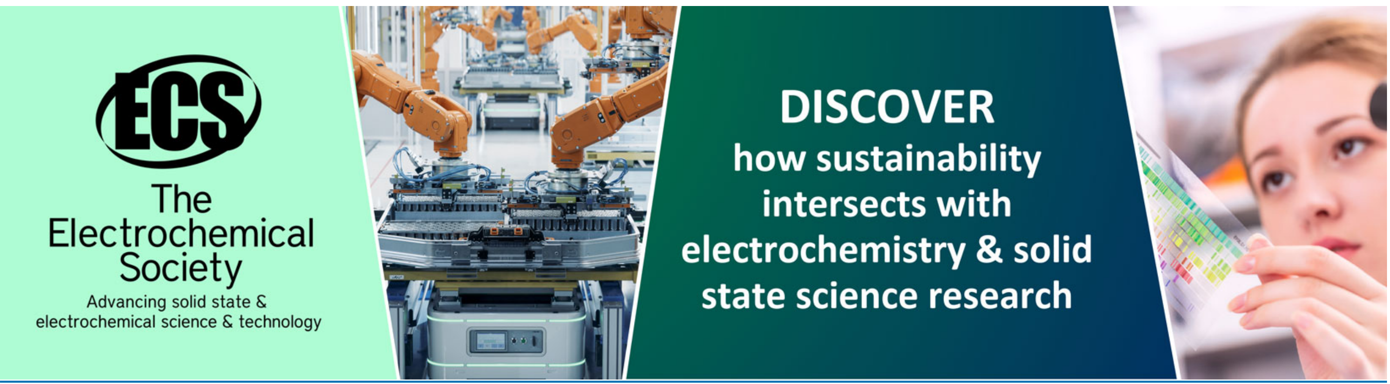
IOP Conference Series: Materials
Science and Engineering

A survey on the modeling and applications of cellular automata theory

To cite this article: Yimin Gong 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **242** 012106



IOP Conf. Series: Materials Science and Engineering **242** (2017) 012106 doi:10.1088/1757-899X/242/1/012106

A survey on the modeling and applications of cellular automata theory

Yimin Gong[1, a)]

[1] Jiangsu Tianyi High School, Wuxi 214101, China

[a)] gongyimin1704@163.com

Abstract . The Cellular Automata Theory is a discrete model which is now widely used in scientific researches and simulations. The model is comprised of some cells which changes according to a specific rule over time. This paper provides a survey of the Modeling and Applications of Cellular Automata Theory, which focus on the program realization of Cellular Automata Theory and the application of Cellular Automata in each field, such as road traffic, land use, and cutting machines. Each application is further explained, and several related main models are briefly introduced. This research aims to help decision-makers formulate appropriate development plans.

1. Introduction

Cellular automata were first developed by John von Neumann as formal models of self-reproducing organisms [1-3]. An original two-dimensional cellular automata, which each cell is a small square in a large grid paper, is developed afterwards. Each cell has two possible states, black and white, which are determined by its neighborhood. In John von Neumann's theory, the neighborhood of a cell is four adjacent squares, as the gray squares in the figure shown below, which is called Von Neumann neighborhood, shown in Fig. 1 and addressed by Z^2.

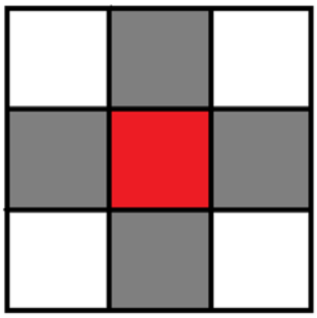


Figure 1. Von Neumann neighborhood

In 1969, computer scientist Alvy Ray Smith completed a Stanford PhD dissertation on cellular automata theory [3]. The author proved that two-dimensional CA were computation universal, introduced 1-dimensional CA, and showed that they too were computation universal, even with simple neighborhoods. It is also showed how to subsume the complex von Neumann proof of construction universality (and hence self-reproducing machines) into a consequence of computation universality in a 1-dimensional CA. In 1970s, a two-state, two-dimensional cellular automaton named Game of Life became widely known [4]. In this cellular automaton, it use a neighborhood named Moore neighborhood (shown in Fig. 2), which is the surrounding eight squares, as the gray squares in the figure shown below [5].

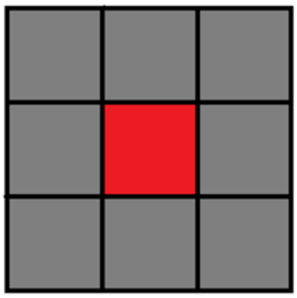


Figure 2. Moore neighborhood

Later the cellular automaton had been extended to a d-dimensional board, addressed Z^d . Recently, in modern cellular automaton theory, the lattice structure is provided by any group. This new algorithm has been referred to as the classical case in the rest of the present paper. Therefore, cellular automata have been used in various topics like group theory, but also recognition, decidability questions, computational universality, dynamical systems. The composition and rules of the game of life are as follows: cellular distribution rules in the grid division; cellular with 0 and 1 of two states, 0 represents "death", 1 represents "live"; at the present time, if a cellular state is "alive", and the eight adjacent cell in two or three the status of "alive", the next time the cellular remain "alive" , otherwise it will turn to "dead"; at the present time, if a cellular state is "dead", and the eight adjacent cell in exactly three "students", the cell will "born"(turn to “alive”. Although the rules are very simple, the game of life is a cellular automaton model with the ability to generate dynamic structure. The distribution of the game of life and the initial state of the cell value is related to the initial state, given an arbitrary distribution, after several step operation, some patterns will soon disappear, but some patterns are fixed, and some repeat two or several patterns. The most famous is the glider, it will move to a fixed direction.

After that, the cellular automata theory is further developed and used in a wide range of areas. Its applications appear in biology, ecology, physics, chemistry, transportation science, computer science, information science, geography, environment science, sociology, military science and complexity science. This paper presents the state-of-the-art modeling and applications of cellular automata theory in recent years. The rest of this paper is organized as follows. In section 2, the datasets employed in experiments are reviewed briefly. Section 3 investigates the properties of the methods on modeling and applications in detail. The conclusion is presented in section 4.

2. Modeling of Cellular Automata Theory

2.1 Definition of Cellular Automata

A cellular automata is a tuple (d, S, N, f) where S is a finite set of states, $N \subseteq_{finite} Z^d$ is the finite neighborhood and $f : S^N \to S$ is the local rule of the cellular automaton. A configuration $c \in S^{Z^d}$ is a coloring Z^d of by S, shown in Fig. 3. The global map $G : S^Z \to S^{Z^d}$ applies uniformly and locally as:

$$\forall c \in S^Z,\ \forall z \in Z^d$$

$$F(c)(z) = f\left(c_{|z+N}\right). \tag{1}$$

IOP Conf. Series: Materials Science and Engineering **242** (2017) 012106 doi:10.1088/1757-899X/242/1/012106

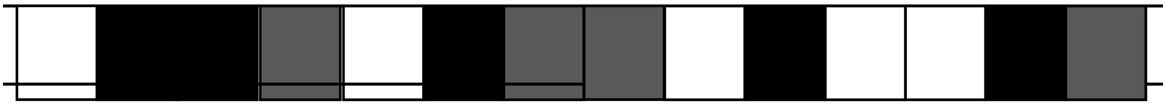


Figure 3. A configuration of cellular automata.

A space-time diagram $\Delta \in S^{Z^d \times N}$ satisfies, for all $t \in N$, there is $\Delta(t+1) = F(\Delta(t))$. Then, the von Neumann neighborhood and the Moore neighborhood can be represented as Eq. 2 and Eq. 3, respectively. Moreover, the set of configurations, S^{Z^d} is uncountable. We can consider a reasonable countable subset, for recursive configurations are useless, undecidability is everywhere (Rice theorem) [4]. First, finite configurations are with a quiescent state. Second, periodic configurations are ultimately periodic. Last, ultimately periodic configurations compromise. It is locality that can also consider partial space-time diagrams to study all configurations, used in next section.

$$N_{vN} = \{0\} \times \{-1,0,1\} \cup \{-1,0,1\} \times \{0\} \quad (2)$$

$$N_{Moore} = \{-1,0,1\} \times \{-1,0,1\} \quad (3)$$

2.2 Modeling of Elementary Cellular Automata

Use a two-dimensional array of $m(n,t)$ in the design of a cellular automaton data storage. In a Matlab simulation environment, put an axis into $n \times t$ box, where t means an element of each cell corresponds to the array, and n is the cellular space. I cellular in the state at t with $m(I,t)$ to represent the 0 indicates death, 1 indicates that the cell is alive. Besides, blue is also used to represent the cell of the 1 state, and white is used to represent the cell. The boundary condition of the cell is periodic, and the neighbor is a cell of the I cell, which is the neighbor length of 1, and the transformation rules are determined for a given number of elementary cellular automata. The algorithm is illustrated in Table 1.

Table 1. The algorithm of elementary cellular automata.

Steps	**Definitions**
1.	Initialize cell states with random numbers.
2.	Enter the conversion rules from the user interface (0-255).
3.	Process the conversion rule number makes him an algorithm subroutine. *Specific algorithm*: read the initial value from the *t* time as $m(1:n,t)$, by the rules to determine and generate the next moment the state of each cell $m(1:n,t+1)$.
4.	Call the subroutine from the 1-*T* loop.
5.	Print the user interface output.

2.3 Modeling of Game of Life

Design of "game of life", we will divide the plane into square, each square represents a cell, cellular state of 0 deaths, 1 said live, the neighborhood radius is 1, the neighborhood was of the Moose type, evolution rules:

$$S(t)=1,\ S(t+1)=\begin{cases}1 & S'=2,3\\0 & S'\neq 2,3\end{cases};$$
$$S(t)=0,\ S(t+1)=\begin{cases}1 & S'=3\\0 & S'\neq 3\end{cases}. \qquad (4)$$

Among them, $S(t)$ represents the state of t cell, and S' is the number of viable cells in 8 adjacent cells. In the program, the user uses the mouse to enter the initial state of the cell through the graphic field. After giving the definition of the neighbor and the local rules, the program can run automatically, and produce a variety of evolutionary patterns. We use two-dimensional matrix $X(m,m)$ to define the state of the cell at the moment t. The specific algorithms are illustrated as follows.

1. Define four direction vectors named adjacency matrix.

$$n=[1 \quad 1:m-1]$$
$$e=[1:m-1 \quad m]$$
$$s=[m \quad 2:m]$$
$$w=[2:m \quad 1] \qquad (5)$$

2. Sum the neighborhood matrix according to the Moore neighbor model and matrix N for east, south, west, north, southeast, southwest, northeast, northwest and the eight neighbors.

$$N=X(n,:)+X(s,:)+X(:,e)+X(:,w)+X(n,e)+ X(n,w)+X(s,e)+X(s,w) \qquad (6)$$

3. Get the algorithm of computing the state matrix of t+1 at the next t moment according to the rule of Comvay.

$$X=\left(X\ \&\left(N=2\right)\right)\mid\left(N=3\right) \qquad (7)$$

3. Applications of Cellular Automata Theory

In this section, it provides the description of Applications of Cellular Automata Theory that are regarded as a benchmark in the current research.

3.1 Traffic Flow Based on Cellular Automaton

A traffic flow based on cellular automaton model named NaSch is a basic model describing road traffic [7]. The model is a discrete dynamical model. The Space is discrete and consists of an infinite regular grid of cells. Each cell is described by a state among a common finite set. Similarly, time is discrete, and at each clock tick cells change their state deterministically, synchronously and uniformly according to a common local update rule. This paper illustrates that the model cannot explain some phenomena because the rules of increasing and decreasing speed of cars is not similar to that in the real world. It should include the influence of the nearest car and the second nearest car. The author introduced some new variables such as the sensitivity coefficient of nearest car and the second nearest car.

3.2 Simulation of Complex Changes Based on Neural-network-based Cellular Automaton

A simulation of complex changes Based on Neural-network-based Cellular Automaton named ANN-CA model is composed of simple networks [9]. The model consists of two relatively independent modules: model correction (training) and simulation. The two modules use the same neural network. In the model correction module, the parameters of the model are obtained automatically by the training data. The structure of the entire model is very simple- the user does not have to define their own rules and parameters, which are suitable for simulating complex land use system. The network has 3 layers, the first layer is the data input layer, where the neurons correspond to the effects of land use change variables; the second layer is the hidden layer; the third layer is the output layer, which is composed of a number (N) of output neurons, and get the probability of changing between each type of land use.

3.3 Temperature Field Simulation in Cutting Process Based on Cellular Automaton

The study on temperature field simulation in cutting process based on cellular automaton takes the temperature field [8]. The field simulation milling insert produced in cutting process as the research object, carries on the grid division of the cutting tool entity with the cellular automata, establishes the cellular automata's model of the milling insert temperature field and studies the optimized mechanism of the milling insert in cutting process, it can enable the cutting value of the milling insert arrive the optimization. In this example, there is a one-to-one correspondence between the equivalence classes of construction triples and the cellular automaton. Note that it is quite common to define a cellular automaton as an equivalence class of construction triples. Many papers use this same definition without mentioning it, for it is supposed to be known, but you may still see that the map in this case is a global transition map.

4. Conclusion

The question arises whether other classical theorems on cellular automata are also true for different models of cellular automata. As the application of cellular automata theory becomes more and more widespread, the modelling of cellular automata on computers became more and more useful. In different fields, the cellular automata are used differently to suit the need. By using specially designed cellular automata, such as Neural-network-based cellular automata and traffic flow cellular automata, we can solve problems with complexity, which are hardly solved by other methods.

References

[1] Sarkar, P. (2000). A brief history of cellular automata. Acm Computing Surveys, 32(1), 80-107.
[2] T. Ceccherini-Silberstein, M. Coornaert, Cellular automata and groups, Springer Monographs in Mathematics, Springer-Verlag, Berlin (2010).
[3] E.R. Berlekamp, J.H. Conway, R.K. Guy, Winning Ways for Your Mathematical Plays, Academic Press (1982).
[4] M. Margenstern, on a characterization of cellular automata in tilings of the hyperbolic plane, Internat. J. Found. Comput. Sci. 19 (2008), no. 5, 1235–1257.
[5] J. Kari, Theory of cellular automata: A survey, Theoretical Computer Science 334 (2005), 3–33.
[6] Shun Wai Tsang, Yee Leung. A Theory-Based Cellular Automata for the Simulation of Land-Use Change. Geographical Analysis 43(2011), no. 2, 142-171.
[7] Xianjuan Kong. Research on Modelling and Characteristics Analysis of Traffic Flow Based on Cellular Automaton. Beijing Jiaotong University, 2007.
[8] Xingqin Cao. The Cellular Automata Studying of Complex System. Huazhong University of Science and Technology, 2006.
[9] Xia Li, Jiaan Ye. Neural network based cellular automata and simulating complex land use system. Geography Research (2005), no. 24, 19-27.