

Spatial Pattern Formation in Asynchronous Cellular Automata with Mass Conservation

Tomoaki Suzudo ¹

Japan Atomic Energy Research Institute, Tokai-mura, Japan 319-1195

Abstract

The pattern formation of asynchronous cellular automata with mass conservation is explored. A cellular automaton rule causing a specified kind of pattern was manually designed, and the mechanism of pattern formation was found similar to real chemical process. In addition, a methodology to automatically search for pattern-forming cellular automata is proposed, and its performance is validated. The searching technique can be applied to a wide range of potential studies related to self-organization.

Key words: asynchronous cellular automata, mass conservation, pattern formation, crystallization, genetic algorithm, automatic cellular automaton rule search

PACS: 05.65.+b

1 Introduction

Self-organization is an essential phenomenon caused by complex systems and cellular automata have intensively been exploited for such studies. Because this mathematical tool is suitable for visual analyses of the calculation results, the self-organization of spatial patterns is a most common topic amongst the cellular-automaton-based complex system studies. For instance, The Game of Life[1], the hodgepodge machine[2], and the cyclic cellular automata[3], they all give intricate patterns and look like real phenomena in nature. Another stream of the study is one related to self-replication, which is started by von Neumann[4] and taken over by Codd[5], Langton[6] and others[7–9] who advanced the notable cellular automaton called the Langton's Loop.

In common cellular automata, all the cell states are updated synchronously and the pattern formations of the above examples are heavily dependent

¹ Email address: suzudo@clsu3a0.tokai.jaeri.go.jp

on this scheme. This means that such pattern formations are prone to tiny errors or perturbations that always exist in the natural environment. Self-organizations observed in real physical, chemical or biological systems are however robust and not dependent on such a synchronous timing. Several scientists already started questioning the assumption of the synchronous update of random boolean networks[10–12] and cellular automata[13].

Another major difference of common cellular automata from natural systems is the lack of conservation laws. This includes mass, momentum and energy conservations. Such conservation laws are adopted in the study of specified physical simulations (see for instance [14]), but these have not been generally adopted by studies of self-organization so far. Without mass conservation, for instance, generating new particles from nothing and annihilating existing particles are not prohibited, and therefore it is easier to design the cellular automata leading to some spatial order than those with mass conservation. The above examples of cellular automata, such as the Game of Life or the Langton's Loop, make the most of this advantage.

Applications of artificial dynamical systems such as random boolean networks or cellular automata to the study of spatial pattern formation are expected to disclose the mechanism of advanced or intelligent level of self-organization, which are frequently seen in biochemical systems, such as self-protection, self-replication, self-maintenance, etc.. Some believes crystal-growth is a primitive form of self-replication and a possible candidate for the origin of life and life-could-be[15]. However, the discrepancies discussed above between artificial and natural systems may spoil the fruit given by such studies.

In this paper, we will consider asynchronous cellular automata leading to a certain spatial order under mass-conservative restriction, although other conservation laws are ignored this time and left for possible future extensions. In the following, the similarity between the pattern formation of such cellular automata and those in real chemical systems is discussed. We will then establish a base for mass production of such cellular automata applying genetic algorithm[16], so that their statistical analyses may enable some advanced studies of self-organization.

2 Asynchronous cellular automata with mass conservation

Cellular automata are dynamical systems such that the current state of each cell is determined by its neighborhood state at the previous time step. We consider in this paper an interactive particle system modeled by a cellular automaton having two kinds of cells, occupied and empty cells. In common cellular automaton scheme each cell has its own neighborhood and its state

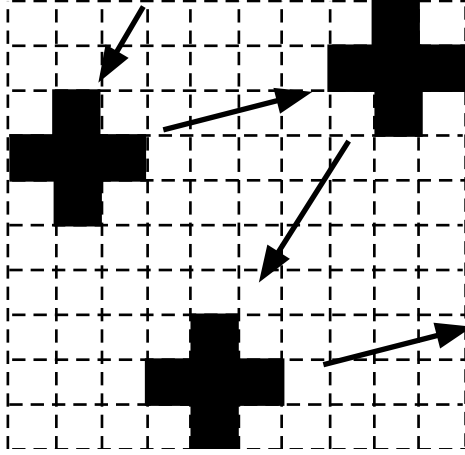


Fig. 1. Random partitioning cellular automata: Randomly selected five black cells are updated in the order of the arrows' direction.

is determined regardless of what states its neighboring cells take. In other words the update rule of a cell is a many-to-one mapping, which is why the total mass is not conserved in such scheme. One technique to realize mass-conservative dynamics with cellular automata is to divide the whole cellular space into many blocks composed of several cells, and the states of each block are updated as the total mass in the block is conserved. In this case the rule becomes many-to-many mapping. This technique is called partitioning. Margonous neighborhood [17] is a typical partitioning method and has been applied to various cellular automata with the conservation law.

While the partitioning technique has mainly been applied to synchronous cellular automata, such as Fredkin's Billiard Ball Machine[18], we will in this study apply the technique to asynchronous updating scheme: The position of the block to be updated is randomly selected, and the state of selected block is updated while the remaining cells are left unchanged. The block is composed of several cells. We will hereafter call this block simply 'neighborhood'. The rule applied to neighborhood is constant over the time just as it is for the common synchronous scheme. See fig. 1 for a schematic picture of the random partitioning in the case von-Neumann-type neighborhood is adopted.

Because the influence of a single update to the dynamics is much smaller than that of the common synchronous method, the unit time is newly defined as it is composed of $\frac{x_{max} \cdot y_{max}}{N_{neighbor}}$ asynchronous updates, where $N_{neighbor}$ is the total number of neighbor cells, i.e. five for the von Neumann neighborhood; x_{max} and y_{max} are the size of the cellular space. With this new definition a cell is on average updated once in a unit time step, which is consistent to the common synchronous updating scheme. Note that some cells may not be updated in a unit time step, while others may be done more than once.

We adopt in this study commonly-used torus-type boundary condition in

Table 1

The properties of the cellular automata studied.

type of property	property used
update scheme	random partitioning, see fig. 1
neighborhood	von Neumann type
size of space	60×60 grid
boundary condition	torus
cell's state	'0'(white cell) or '1'(black cell)
initial configuration	random

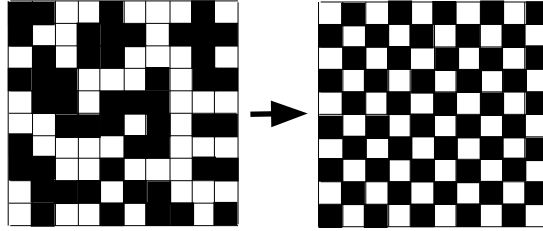


Fig. 2. A simple example of crystalline pattern in two-dimensional cellular automata.

which the top and left edges of the cellular space are connected to the bottom and right ones, respectively. We also adopt a square cellular space of $x_{max} = y_{max} = 60$ throughout this paper, because this size was confirmed to be large enough to overcome size-effects through the priori numerical studies. Each cell's state is assumed to take an integer equal either to nil or unity. The nil state denotes an empty cell having white color, and the other state is one occupied by a particle having black color; i.e. this is a mono-particle artificial chemical system. The particles merely move around the cellular space, but they never separate into two new pieces, nor do two particles merge to form a new particle. To satisfy this mass conservation law, the update rule must be defined so that the total number of particles in neighborhood does not vary. The initial global configuration of the cellular space is randomly set with the specified density. For this purpose, a unique randomizing seed is used for each run, so that the initial configurations are always different. Table 1 summarize the properties of the cellular automata used.

3 Searching for cellular automaton rules forming “crystalline” pattern

This section considers designing cellular automata that leads random configuration to a spatial pattern, especially one that each particle is diagonally linked to others(see fig. 2).

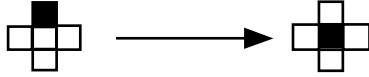


Fig. 3. A transition to cause isolated particles to move: Three rotationally symmetric entries are similarly defined.

The first step of the designing process is to select the neighborhood type. Generally speaking, small neighborhood limits the freedom of rule designing, because the varieties of transitions are limited. Ones with large neighborhood, on the other hand, also has a difficulty; the number of transition entries to define is large and it becomes messy when defining all of the transitions. Through the studies in try-and-error manner, von Neumann neighborhood proved to have the enough freedom of designing to produce rules forming the pattern in fig. 2 without much difficulty, and adopted in the following discussion.

The second step is the determination of each transition that is a mapping from a neighborhood pattern to another neighborhood pattern. There are 32 transitions for each rule in this case. We at first set all of the transitions to be “inactive”: Any patterns of neighborhood are not changed by the update. In the following we will replace some inactive transitions with “active” ones that change the pattern of neighborhood by the update. Note that, for the simplicity sake, we will in this section consider only rotationally symmetric rules.

If isolated particles are still, they can never be a part of the pattern formed. Thus we must adopt the following principle.

Principle I: Isolated (free) particles move around to search for other particles.

If a neighborhood pattern includes only one particle, such particles may be isolated. In order to satisfy the above principle, some transitions from such neighborhood patterns must be active. There are only five entries of this kind, that is, one entry with a particle at the center cell and the four entries with a particle at one of the four “wings”. Actually, the entry with a particle at the center cannot be active, otherwise rotational symmetry would not be satisfied. The remaining four transitions can be active with keeping the rotational symmetry of the rule satisfied. We adopt the transitions that move the particle from the wing to the center, that is, one shown in fig. 3 and the other three rotationally symmetric transitions.

Through the transitions above the particles can be gathered, but not necessarily in the form of fig. 2. Therefore we need

Principle II: Particles nearby shall be rearranged as they become diagonally linked to other particles.

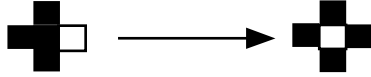


Fig. 4. An example of a transition to cause the particles to become diagonally linked.



Fig. 5. An example of a global configuration produced by the basic rule.

To support Principle II, we give a minimum set of active transitions which seems natural. These are the transitions shown in fig. 4 and three other rotationally symmetric transitions.

So far we adopted eight active transitions to rearrange the particles. We will hereafter call the rule composed of these eight active and the remaining 24 inactive transitions the basic rule.

Figure 5 shows a snapshot of cellular automaton evolved by the basic rule at the 500 unit time. We set the density at 0.5 for this case, simply because the specified pattern needs the density of 0.5 when the whole space is perfectly covered by the intended pattern. Some parts in fig.5 form the intended pattern, but there are also many errors. These errors are not improved by further time development.

Empirical studies were conducted by adding some other active transitions to the basic rule, to try to improve the fidelity of the pattern. However, it was not possible to make the situation drastically better. Besides, the effect of changing one transition was not always clear. For example, we could find another active transition to improve the fidelity in some case, but this transition may make the situation worse if it is combined with other active transitions. This is actually not surprising, because the effect of rule transitions is interrelated to other transitions.

Through the discussion so far, it seems that making a pattern-forming cellular automata is a messy optimization problem, and that we have no guiding

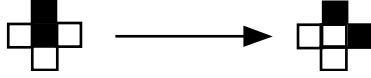


Fig. 6. Adding this and other three rotationally symmetric transitions to the basic rule gives chaotic behavior.

principle to tune the rule. In the next section, we will introduce a new parameter to characterize cellular automaton rules and obtain a better view of relationship between these rules and the pattern formation.

4 Parameterization of crystal growth cellular automata

In the real chemical process we would tune the speed of crystal growth to maximize the correctness of the microscopic pattern: The slower the speed is, the more exact is the pattern created, because erroneous bindings are given more opportunities to be corrected. This speed is related to the saturation degree of solution. If this degree is too low, molecules tend to move around the space and never stay as a part of solidified structure. If it is too high, on the other hand, the crystal grows too fast and imperfect bindings are not likely corrected. The value of saturation degree is determined by the combination of temperature and density of the solution. Therefore, to maximize the correctness of the intended spatial pattern in real chemical systems, it is necessary to tune the two parameters so that the saturation degree becomes close to the critical value between undersaturated and supersaturated, or more precisely to the value slightly supersaturated so that the solution is solidified in the long run. In this section, we would like to know whether or not a similar mechanism governs the physics of pattern formations of cellular automata.

The question is then what kind of characteristic quantities in our cellular automata correspond to density and temperature of the solution. The counterpart for density is easily defined, because there is a similar concept in the cellular automata with mass conservation. For example, the ratio of the total number of particles in the cellular space against the total number of cells would be adequate. Note that this ratio is invariant over the temporal development of mass-conservative cellular automaton dynamics. This ratio will simply be called density hereafter.

Defining the “temperature” is much more difficult because there is no explicit counterpart. A characteristic quantity of cellular automaton rules like λ -parameter [19] or μ -parameter [20] seems a possibility. Such concepts are however merely meaningful in the sense of statistics, and to adopt them as a tuning parameter is not realistic. We want a parameter that can change the saturation degree more smoothly.

As mentioned above, the basic rule we reached in the previous section can be improved by adopting some additional active transitions. Therefore the “temperature” of cellular automaton given by the basic rule is less than the critical value, i.e. the solution is supersaturated. We also recognized that adding other active transitions to the basic rule led the dynamics to chaotic evolution; the temperature of this cellular automaton is more than the critical value, i.e. the solution is undersaturated. For example, the rule with four extra active transitions from the neighborhood including two particles, one of which is shown in fig.6, caused chaotic evolution. Thus a rule holding the critical temperature is “between” the two rules or a “mixture” of them. In order to create the “mixed” rules, we will consider a class of rules which always apply the active transitions in the basic rule and stochastically applies the additional four active transitions introduced above with a constant probability, P_a : At each time when the updating process shown in fig. 1 encounters one of these neighboring patterns, the transition is applied with the probability of P_a and keep the pattern unchanged with the probability of $1 - P_a$. We again set the density at 0.5 in the following, and conducted the numerical studies with varying P_a , hoping that the quantity would work as the temperature.

To analyze the given numerical results, we now introduce a new quantity called an “active ratio” to measure the dynamics of cellular automata. This is the ratio of total moving particles observed between every two consecutive time steps against the all particles in the cellular space. This quantity is usually large (i.e. ~ 1) at the beginning of the time development of cellular automata, and is stabilized to a certain steady value as the time passes. The value is therefore expected to illustrate how static or chaotic the dynamics are. The solid line in fig.7 denotes the stabilized active ratio varying with P_a . To make the quantity statistically meaningful, values averaged over the time steps of 2001 – 2500 were calculated and such averaged values are calculated 100 times for each case. The values given in the figure are the values averaged over these 100 cases. The graph indicates that the particles are hardly active at $P_a = 0$, the active ratio simply increases as P_a does so, and it becomes about one fourth at $P_a = 1$: P_a works as a temperature in effect. The graph of the active ratio (the solid line) has the steepest slope at $P_a \cong 0.3$, which is the estimated critical value between undersaturated and supersaturated.

The broken line in fig. 7 shows the fidelity of the pattern formation. This value is calculated by counting the number of the local pattern shown in fig. 8 found in a snapshot of the global cellular space. The snapshots are taken after the transient period (2500 time steps in this case). The value is normalized as it is unity when the whole space is completely covered by the pattern. In the figure are the fidelity values averaged over 100 cases. The graph indicates that the fidelity has the maximum value around at $P_a = 0.1$.

The two graphs in the figure claim that the optimal probability for making

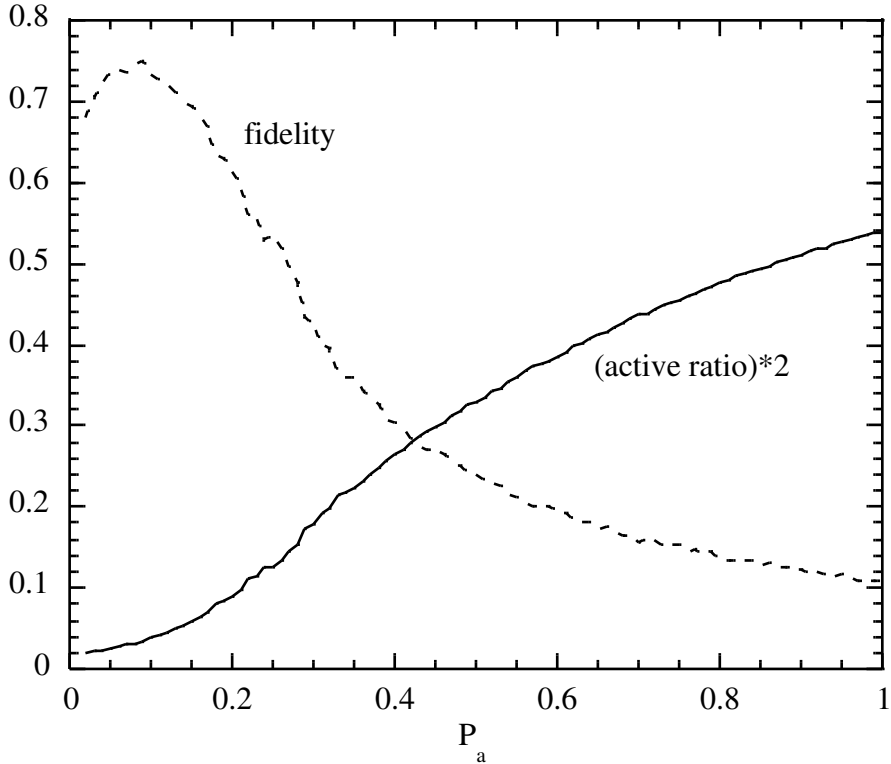


Fig. 7. The active ratio (the solid line) and the fidelity (the broken line) change according to P_a .



Fig. 8. The local pattern used for the calculation of fidelity.

the pattern is slightly less than the critical value. This result matches the fact that the crystallization is the most likely to occur when the solution is slightly supersaturated. The discussion above implies that the mechanism of pattern formation for the class of cellular automata is similar to real chemical processes.

Because the two parameters that govern the pattern formation are found, it is possible to view how the fidelity varies on the two parameters' plane. Figure 9 shows the fidelity's variation on the P_a -density plane. The peak appears at the density of ~ 0.5 , but it is stretched in the wide range of $P_a = 0.1 - 0.4$. This broad peak is not visible at fig.7, because the tail of the peak at the large P_a 's side have the density slightly less than 0.5, see fig. ??.

The surface of the spatial entropy H_s was also calculated. (Rigorous definition of H_s is located at Appendix A.) Figure 10 displays it on the P_a -density plane. Naturally the entropy is small when the density is very low or very high, and it is large when the density is ~ 0.5 . Notice that a trough can be seen at the exactly same place of the fidelity's peak, which ensures the trough is caused

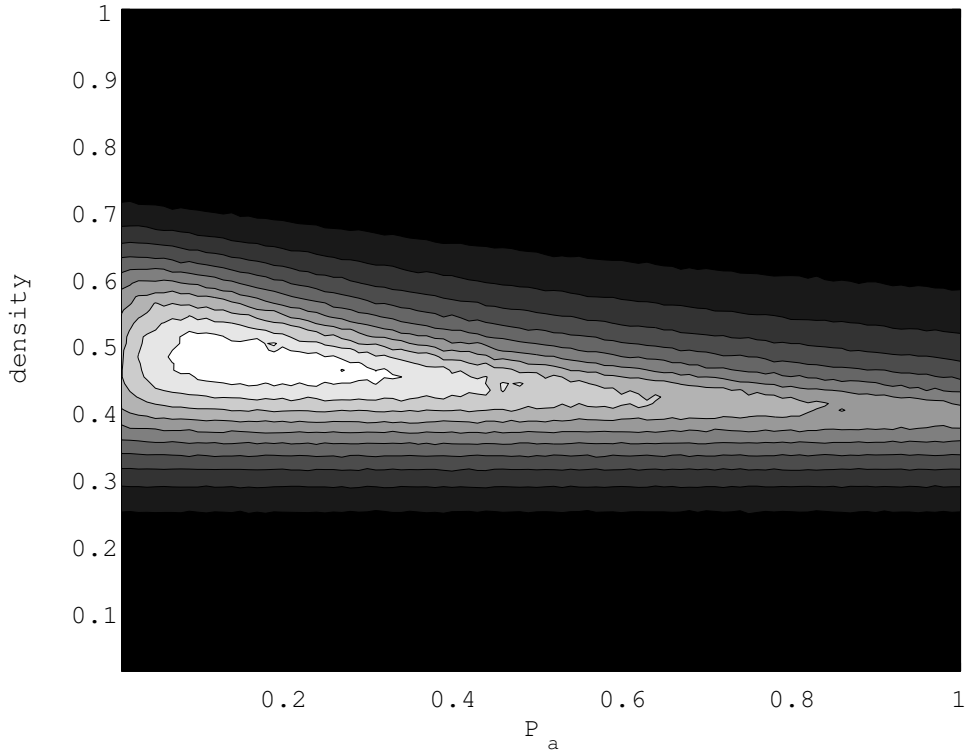


Fig. 9. Contour map of fidelity calculated with various P_a -and-density combinations. The darker the color is, the smaller the fidelity is.

by the pattern formation. The calculation of spatial entropy does not need the prior information on the pattern type, and is generally applied to pattern formations. This is thus useful when the type of pattern is not specified, and will be exploited in the next section.

5 Searching automatically for pattern-forming cellular automata

So far we have conducted the numerical analyses based on a certain carefully designed cellular automaton. This section in turn deals with another approach, that is, the pattern-forming cellular automata are searched for automatically. As discussed above, on the P_a -density plane is a trough of the spatial entropy where the corresponding cellular automata form the intended regular pattern. Because the P_a -density plane is a special case of “rule-density plane”, troughs as many as regular pattern types must exist on the general rule-density plane. If we can find such troughs effectively, we are able to encounter many kinds of pattern formation even those we never expect.

As one can see from fig. 10, the basin of the trough is not global and minimizing the spatial entropy on the P_a -density plane mostly leads not to the

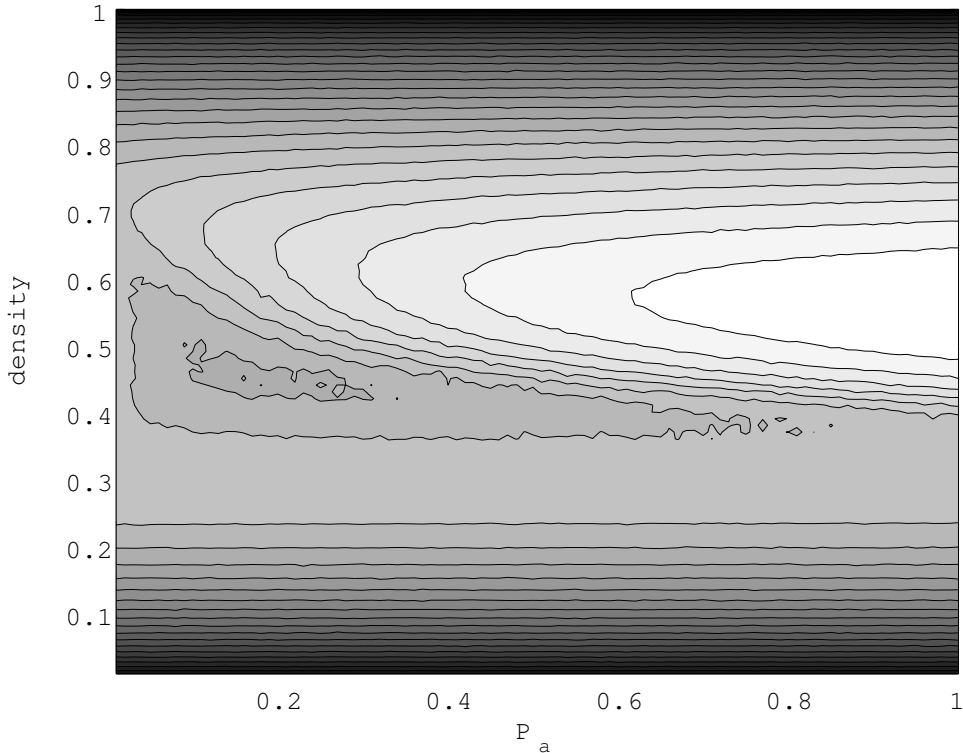


Fig. 10. Contour map of the spatial entropy calculated with various P_a -and-density combinations. The darker the color is, the smaller the spatial entropy is.

trough, but to the top ($density = 1$) or bottom line ($density = 0$). To escape these unwanted behaviors of optimization, the density is fixed to a certain constant value throughout the optimization process. In fact, being able to impose this restriction is a great advantage of mass-conservative cellular automata, because such trough can hardly be found if the density is time-variant. In the following examples the density is fixed to 0.43 which is approximately the lowest end of the trough (see fig 10).

Although the previous section introduced the stochastic rule, we will concentrate in this section on common deterministic ones for the sake of generality. The space of cellular automaton rules is in this case discrete and two close rules arbitrary selected do not necessarily produce a similar development, and the optimization of cellular automaton rules tends to become a messy problem as already mentioned before: Classical tools such as steepest descent method are not applicable.

Genetic algorithm is however useful for such problems that the function to be minimized (or maximized) has a lagged surface, and we applied this technique to searching the rule space for pattern-forming cellular automata. More precisely, we searched for rules leading to some kind of pattern formation out of those with the properties listed in Table 1. Such rules are composed of $2^{N_{neighbor}} = 32$ integer values, and this array of integers became the “chro-

mosome” of this genetic algorithm. Searched rules were limited to the deterministic mass-conservative ones, but the condition of rotational symmetry previously applied was abandoned: The whole search space is then composed of $\sim 10^{14}$ rules.

The fitness function was selected as

$$F = \left(\frac{1}{H_s(t_c) - H_s^o}\right)^2 \cdot t_c^2, \quad (1)$$

where $H_s(t)$ is a spatial entropy of global cellular states at the time step t ; H_s^o is H_s 's target value of the optimization; t_c is the transient time, that is, the time when H_s 's time development becomes almost steady. The length of t_c is determined by the sequential tests of

$$|H_s(t+1) - H_s(t)| < \epsilon, \quad (2)$$

$$|H_s(t+2) - H_s(t+1)| < \epsilon, \text{ and} \quad (3)$$

$$|H_s(t+3) - H_s(t+2)| < \epsilon. \quad (4)$$

The smallest t satisfying all of eqs.(2)-(4) was considered as t_c , where ϵ was empirically set at 0.003. Recall the discussion at the beginning of the previous section saying that the speed of “solidification” is slow, i.e. the transient time (i.e. t_c) is long, in case a pattern formation occurs. For this reason, the multiplication by t_c^2 in eq.(1) was added to a simple square of $\frac{1}{H_s(t) - H_s^o}$, and it was empirically found that this multiplication accelerated the optimization speed.

The population of the genetic algorithm was set to 10 for each optimization run, and this initial set of rules was randomly selected. To evolve the chromosomes, five pairs of parents were randomly selected with the probability proportional to their fitness functions (this is called fitness-proportionate selection) and five pairs of infants were generated through crossover and mutation of the parents. The rules were evolved until at least one of the ten infants is found $H_s(t_c) < H_s^o$. The value of H_s^o was set at 0.6 taking the numerical studies in the previous section into consideration.

Many runs of the optimization trial were conducted and they all gave the spatial entropy of less than H_s^o . Generations needed to reach $H_s(t_c) < H_s^o$ varied between several hundreds and a few thousands, and it may be shortened by tuning. The spatial patterns the optimized rules gave can be classified into three types (see fig. 11), i.e. (a)diagonally-connected, (b)sand-like and (c)stripe patterns. The pattern in fig. 11a is the one we have been discussed in the

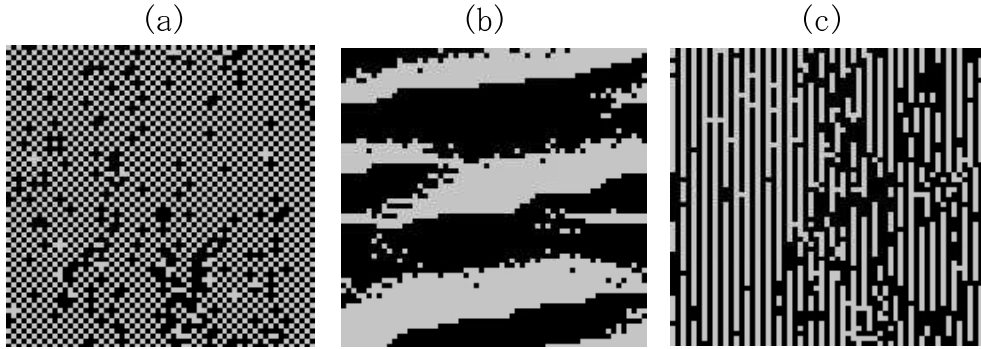


Fig. 11. The three kinds of patterns found by using genetic algorithm.

previous sections. Although these three patterns can be imagined in advance, it is interesting to confirm that there is a trough of H_s for each pattern and that they can really be found. The results suggests that the optimization technique using genetic algorithm can generally be applied to the study of pattern formation of mass-conservative cellular automata.

6 Conclusion

As a study of self-organization, the spatial pattern formation of asynchronous cellular automata under the mass conservation law was explored, and the random partitioning scheme was applied for this purpose. A cellular automaton rule causing a specified pattern was step-by-step designed. A new continuous parameter was then introduced to give perturbation to the rule, and the relationship between the new parameter and the fidelity of the pattern was examined. The results indicated the fidelity have the maximum value at a critical point, which is similar to real chemical process: This promise the usefulness of this type of cellular automata to model complex systems in nature. A methodology to automatically search for the pattern-forming cellular automata was proposed, and its performance was validated. The methodology can easily be extended to more complicated situation such as mixture of many kinds of particles or adding energy and momentum conservation laws and etc.. We expect that these extensions could open a wide range of potential studies related to self-organization in nature, for example to a problem on what are the source of the order of living world[21].

References

- [1] Gardner, M., *Mathematical games*, Sci. Amer. **223 October**, 123(1970).
- [2] Dewdney, A.K., *Computer recreations*, Sci. Amer. **August**, 86(1988).

- [3] Dewdney, A.K., *Computer recreations*, Sci. Amer. **August**, 102(1989).
- [4] von Neumann, J., *Theory of self-reproducing automata* Urbana, Illinois: University of Illinois Press (1966). (edited and completed by A. W. Burks)
- [5] Codd, E. F., *Cellular Automata*, Academic Press, New York(1968).
- [6] Langton, C. G., *Self-reproduction in cellular automata*, Physica D **10**, 135(1984).
- [7] Reggia, J. A., Armentrout, S. L., Chou, H.-H. & Peng, Y. *Simple systems that exhibit self-directed replication*, Science **259**, 1282(1993).
- [8] Chou, H.-H., & Reggia, J. A., *Emergence of self-replicating structures in cellular automata space*, Physica D **110**, 252(1997).
- [9] Sayama, H., *Self-replicating worms that increase structural complexity through gene transmission*, Alife **VII**, The MIT Press, 21(2000).
- [10] Harvey, I., & Bossomaier, T., *Time out of joint: Attractors in asynchronous random boolean network*, Proceedings of the Fourth European Conference of Artificial Life, The MIT Press, 67(1997).
- [11] Di Paolo, E. A., *Searching for rhythms in asynchronous random boolean networks*, Alife **VII**, The MIT Press, 73(2000).
- [12] Di Paolo, E. A., *Rhythmic and non-rhythmic attractors in asynchronous random boolean networks*, BioSystems **59(3)**, 185(2001).
- [13] Nehaniv, C. L., *Self-reproduction in asynchronous cellular automata*, University of Herfordshire, Faculty of Engineering and Information Sciences, Technical Report **368**, 21(2002).
- [14] Creutz, M., *Microcanonical Monte Carlo simulation*, Phys. Rev. Lett. **50(19)**, 1411(1983).
- [15] Cairns-Smith A.G., *Seven Clues to the Origin of Life – A Scientific Detective Story*, Cambridge University Press (1985).
- [16] Mitchell, M., *An Introduction to Genetic Algorithms*, The MIT Press (1996).
- [17] Toffoli, T., & Margolus, N., *Cellular Automata Machines -A New Environment for Modeling*, The MIT Press (1987).
- [18] Fredkin, E., & Toffoli, T., *Conservative logic*, Int. J. Theor. Phys. **21** 219(1982).
- [19] Langton, C. G., *Computation at the edge of chaos: phase transition and emergent computation*, Physica D **42**, 12(1990).
- [20] Suzudo, T., *Crystallisation of 2-dimensional cellular automata*, Complexity International **6**, (1999) (online journal at <http://www.csu.edu.au/ci/>).
- [21] Kauffman, S. A., *The Origins of Order – Self-organization and Selection in Evolution*, Oxford University Press (1993).

A Definition of spatial entropy of cellular automata's global configuration

Consider four adjacent sites of the cellular automaton space such as (i, j) , $(i + 1, j)$, $(i, j + 1)$ and $(i + 1, j + 1)$, where i and j are the x-axis and y-axis position of an arbitrary cell, respectively. There are $2^4 = 16$ possible patterns for this local patch if each cell takes either '0' or '1' state. In this paper, we use

$$H_s(\tau) \equiv - \sum_k P_s^k(\tau) \log_{16}(P_s^k(\tau)), \quad (\text{A.1})$$

as the definition of the spatial entropy of global configuration at an arbitrary time step τ , where $P_s^k(\tau)$ is the probability for a particular pattern of the local patch at the time step τ . Note that a base of 16 is used for the logarithmic function as the entropy assumes a real number between nil and unity.