Determining the Critical Temperature of the Continuous-State Game of Life

Susumu Adachi[1], Jia Lee[1,2], Ferdinand Peper[1],
Teijiro Isokawa[3], and Katsunobu Imai[4]

[1] Nano ICT Group, National Institute of Information and Communications Technology, Japan
[2] College of Computer Science, Chong Qing University, China
[3] Division of Computer Engineering, University of Hyogo, Japan
[4] Faculty of Engineering, Hiroshima University, Japan

Abstract. This paper proposes a simple algorithm to find the critical temperature of the continuous-state Game of Life (GoL). The algorithm conducts the transitions of cells and the update of the temperature parameter alternatingly. The temperature starts from a low value and it increases gradually, while a fixed GoL pattern evolves. This process continues, but before the temperature exceeds the critical temperature, the update algorithm acts to decrease it, so as to prevent overshoot of the temperature, which would make the cell states deviate from the normal GoL behavior. An oscillatory value of the temperature can be observed, but it converges towards a fixed value, indicating that its critical point is being approached.

1 Introduction

Cell states in conventional Cellular Automata (CA) assume discrete values, but when continuous states are allowed [9], it becomes possible to investigate dynamics that is chaotic or at the edge of chaos [1,10,11,12,13] or to apply it to biological modeling [2], the coupled map lattice [4,5], etc.

One model with continuous cell states is the so-called Game of Life at finite temperature [3]. A cell state in this model lies in the range $[0 - 1]$ and it is updated in a synchronous way at discrete time steps, such that a cell's state after update is determined by its own state and the sum of the neighbouring cell states at the previous time, as well as by a temperature parameter.

It has been observed [3] that the temperature parameter plays an important role in the behavior of the model. When it is below a certain value, which we call the *critical value*, the model behaves like the traditional (discrete-state) GoL [6,7,14], but once this critical value is exceeded, different behavior emerges. The degree at which the behavior differs from the traditional GoL tends to be proportional with the difference of the actual temperature and the critical temperature below it. Though the use of the term *critical temperature* implies that it is at a unique point, the reality is more complicated. It turns out that the critical temperature is somewhat dependent on the actual configurations

G.C. Sirakoulis and S. Bandini (Eds.): ACRI 2012, LNCS 7495, pp. 83–90, 2012.

in cell space, and in fact it is a range of values. The minimum of these values represents a critical point below which every configuration behaves as if it were in the traditional GoL, and we are interested in this particular value. We call this the *critical point* of the model.

In this paper we formulate an algorithm that determines this critical point by adapting the temperature over the course of iterations. The algorithm employs two phases in alternation: cell update and temperature update. The temperature update is based on a steepest descent algorithm, whereby the gradient of the transition function is updated with a limiting value of 1. The algorithm then selects a target cell in the cell space, which distinguishes itself from other cells by having the largest gradient. The meaning of a cell's gradient being the largest is that that cell will start to decay first, i.e., behave in a different way than it would in the traditional GoL. It is demonstrated that when the temperature starts from a low value and increases gradually, a fixed GoL pattern evolves. This process continues, but before the temperature exceeds the critical temperature, the update algorithm acts to decrease it, so as to prevent overshoot of the temperature, which would make the cell states deviate from the normal GoL behavior.

This paper is organized as follows. In Section 2, the continuous-state GoL is explained. The methods and the results of the algorithm used to find the critical temperature are shown in Section 3. Conclusions are drawn in Section 4.

2 Continuous State Game of Life

The transition rule of the continuous state Game of Life [3] is described by

$$f(C_i) = 1/\left[1 + \exp(-2E(C_i)/T)\right] \tag{1}$$

$$E(C_i) = E_0 - \left(C_i + 2\sum_{j \neq i} C_j - x_0\right)^2 \tag{2}$$

where, C_i is the continuous state ($0 \leq C_i \leq 1$) of the i-th cell, j varies over the Moore neighbourhood, T is the temperature, $E(C_i)$ is the local energy, and E_0 and x_0 are constants. All cells are selected in an iteration and their states undergo transitions according to this function f synchronously, i.e., $C_i(t+1) = f(C_i(t))$. Note that the transition function has 3 variables, the center cell's state, the summation of the neighbouring cells' states, and the temperature. If the constants are $E_0 = 2.25$ and $x_0 = 6$, this function is identical to the rule of the Game of Life at zero temperature limit, as can be shown. For this reason, these typical values are used in this paper.

A *pattern* is defined as a set of living cells that is a subset of the cell space, whereby a cell is called *alive* when its state is larger than 0.5, and dead otherwise. The patterns of the model decay above the critical temperature T_c, and sustain their typical Game of Life forms below T_c. The critical temperatures are different for different patterns, for example, 0.53 is the critical temperature for a 'beehive',

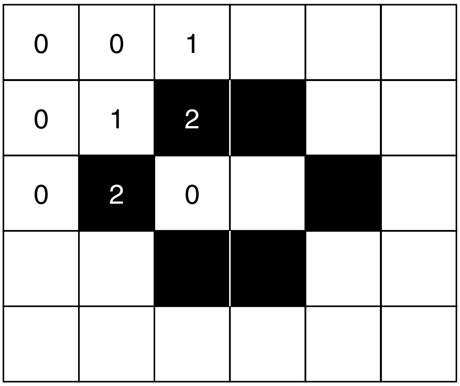


Fig. 1. A beehive pattern. Cells with the same indexes take about the same value at a stable situation.

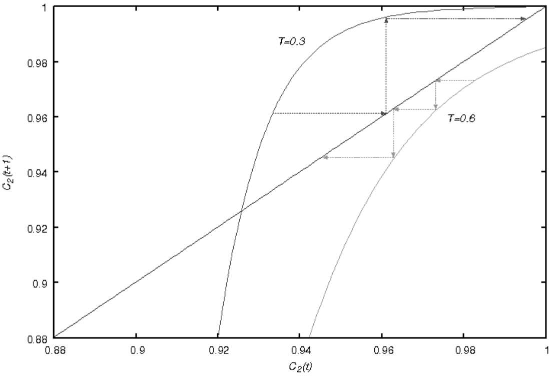


Fig. 2. A part of the map of the cell state C_2 at time t and $t+1$

0.61 for a 'glider gun', 0.66 for an 'eater', 0.7 for a 'glider', and 0.8 for a 'block', respectively. Since glider guns, eaters, and gliders are required as logic gates for computing [7,8], this model is computational universal at $T < 0.61$.

We show the dynamics of how a pattern decays above the critical temperature T_c. Below the critical temperature, the beehive pattern is stable as shown in Fig.1. In the figure, we assume that the cells with the same indexes take about the same value.

Relations of the states of cells denoted by 1 and 2 at the stable situation can be expressed as follows,

$$E_1 = E_0 - (3C_1 + 4C_2 - x_0)^2 \tag{3}$$

$$C_1 = 1/\left[1 + \exp(-2E_1/T)\right] \tag{4}$$

$$E_2 = E_0 - (5C_2 + 6C_1 - x_0)^2 \tag{5}$$

$$C_2 = 1/\left[1 + \exp(-2E_2/T)\right] \tag{6}$$

where, $C_0 \sim 0$ is assumed for simplicity. Assuming $T \sim 0.5$, then we obtain $C_1 \sim 0.0005$ and $C_2 \sim 0.99$ numerically by an iterative computation. Based on this value of C_1, a map of $C_2(t+1) = f(C_2(t))$ at T is obtained that is shown in Fig.2.

If the temperature is smaller than the critical temperature ($T < T_c$), there are three intersection points, and C_2 converges towards the largest of them. Otherwise ($T > T_c$), there is only one intersection around 0, and C_2 converges towards it. If $T = T_c$, there are two intersections at 0 and 1, and the gradient of the function is 1 around the upper intersection.

Generally the mechanism of pattern decay is based on the disappearance of bistability. However the detailed situation of this model is slightly different from the Ising model in physics (or Hopfield Neural Network). Since the magnetization of the Ising model is equal to the solutions of an equation like $x = \tanh(x/T)$, which is derived by using mean field approximation, the disappearance of the bistability occurs continuously towards $x \to 0$ at $T \to 1$, which is a so-called phase transition.

In this model, since the intersection points of $y = x$ and $y = f(x)$ around 1 disappear at $T > T_c$, the cell state jumps to the other intersection point around 0. In general, this highly non-linear characteristic of the transition function of the model guarantees the survival of life patterns at low temperatures.

3 Approximating the Critical Temperature

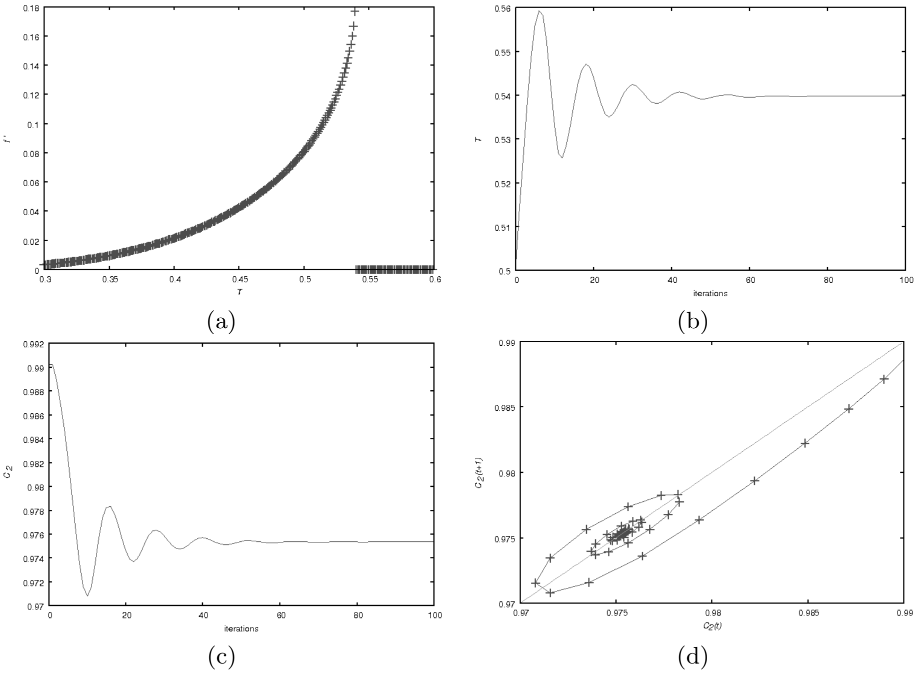
We introduce a simple framework to find the critical points, starting with a low temperature, and then carry out transitions of cell states and the update of the temperature in an alternating way. When the update is executed so that the gradient of the transition function is 1, then it can be expressed by the steepest descent as follows.

$$T \to T - \mu \frac{\partial g}{\partial T} \tag{7}$$

$$g = \left(1 - \frac{\partial f}{\partial C_i}\right)^2 \tag{8}$$

where μ is a constant and g is the error function. The index i of cell C_i is chosen as the point where the gradient $\partial f / \partial C_i$ takes its maximum value in the cell space. The reason of taking this i is that if the gradient of a cell is larger than 1, the cell starts to decay as shown in the previous section. In other words, a critical temperature is the maximum value at which the weakest cell does not decay. In the example of Fig. 1, the left cell of two cells C_2 labeled 2 is the weakest cell (since the pattern is symmetric, there are two of such weakest cells).

There is a problem in this framework that if the temperature becomes larger than the critical point at a certain time and it becomes smaller according to Eq. (7), it is too late to recover the cell state. This situation can be understood by investigating the gradient being dependent on the temperatures at the stable situation, as shown in Fig.3 (a). In the figure, the gradient is 0.177 at $T = 0.539$,



(a)

(b)

(c)

(d)

Fig. 3. Gradient of a living cell as a function of T (a), dynamics of temperature (b), dynamics of the cell state (c) and its map representation (d). The size of the cell space is 20×20, the initial configuration is beehive, the initial temperature is 0.5, and the constants are $\mu = 0.1$ and $\theta = 0.2$.

and ~ 0 at $T = 0.54$. Although the critical point must be between them, the gradient does not exceed 0.2 within the double precision of T due to sensitivity considerations. The problem of an overshooting temperature can be solved in the following ways:

1. replace double precision of T with quadruple precision.
2. asymmetric update (decrease the temperature more than in the original algorithm).
3. replace $g = (1 - f')^2$ with $g = (\theta - f')^2$, whereby θ is an appropriately chosen threshold.

We adopt option 3 above, because of reasons of simplicity: it only requires the change of the constant 1 into θ in Eq. (8). Since we have the knowledge that the gradient does not exceed 0.2, we can safely choose $\theta = 0.2$. Then the expression Eq. (8) can be rewritten as follows.

$$g = \left(\theta - \frac{\partial f}{\partial C_i}\right)^2 \tag{9}$$

The experiment is carried out as follows. The size of the cell space is 20×20, the initial configuration is the beehive pattern, the initial temperature is 0.5, and the

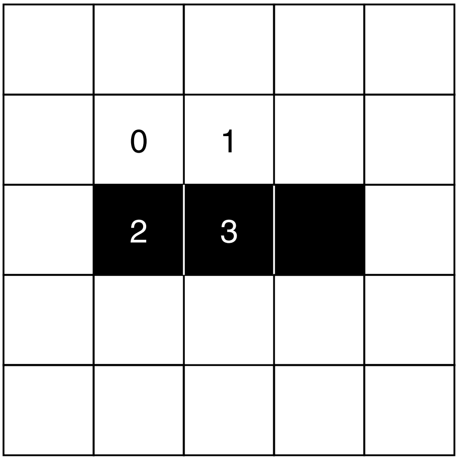


Fig. 4. A blinker pattern. The cell C_3 takes the maximum value of the gradient.

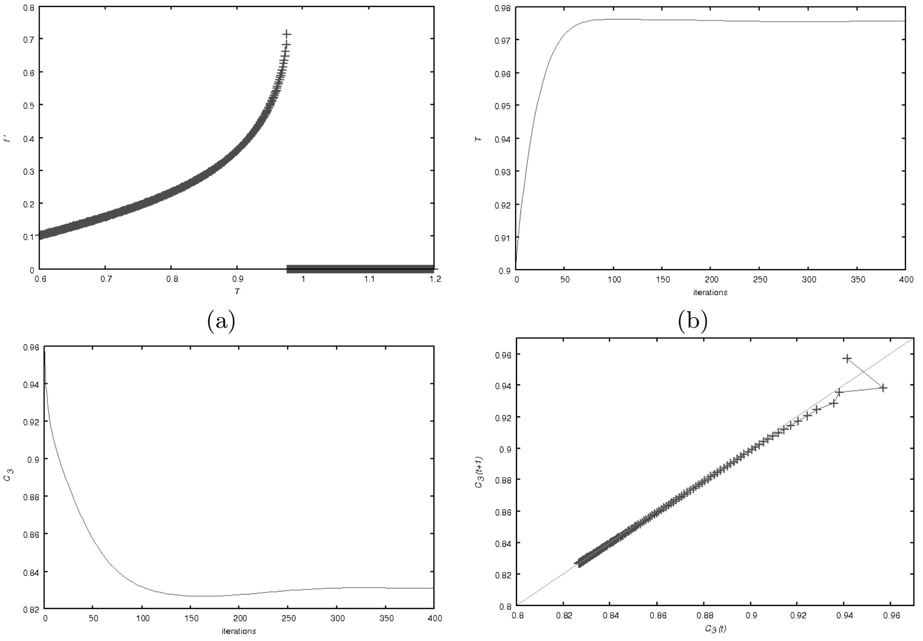
constant is $\mu = 0.1$. Some preliminary iterations are performed to make the cell space stable. The result of the experiment is that the temperature oscillates and converges towards the critical point, as shown in Fig. 3 (b). The time evolution of the cell state C_2 and its map representation are illustrated in Fig. 3 (c) and (d).

The next example concerns the blinker pattern shown in Fig. 4. It is known that the blinker is a period 2 oscillator. The center cell of the blinker C_3 is the weakest cell, which means its gradient has the maximum value. The gradient plotted as a function of T is shown in Fig. 5 (a). In the figure, the maximum value is 0.715 at $T = 0.976$, so in this case it is possible to set $\theta = 0.7$.

The experiment is carried out as follows. The size of cell space is 20×20, the initial configuration is the blinker pattern, the initial temperature is 0.9, and the constant is $\mu = 0.01$. Some preliminary iterations are performed to make the cell space stable at every even steps. As a result, the temperature oscillates and converges to the critical point, as shown in Fig. 5 (b). The time evolution of the cell state C_3 and its map representation are illustrated in Fig. 5 (c) and (d).

In both cases of the beehive and the blinker, the algorithm can work well, because the constant θ plays a role of an early stopping to increase the temperatures. There are also some cases it fails. If μ is large, the cell's states can not follow the updated temperature, then the states can not converge and in the end the temperatures continue to oscillate. On the other hand, if μ is small, the temperature converges below the critical point or oscillates periodically tracking a hysteresis loop.

Finally, the results of the critical temperatures for some famous patterns including the above patterns can be shown in Table.1.



(a)

(b)

(c)

(d)

Fig. 5. Gradient of a living cell as a function of T (a), dynamics of temperature (b), dynamics of the cell state (c), and its map representation (d). The size of the cell space is 20 × 20, the initial configuration is a blinker, the initial temperature is 0.9, and the constants are $\mu = 0.01$ and $\theta = 0.7$.

Table 1. Threshold values of the gradient and critical temperatures for some patterns are listed

pattern	θ	T
beehive	0.2	0.539694
paperclip	0.3	0.577207
loaf	0.3	0.597193
boat	0.35	0.612362
tub	0.195	0.639911
eater	0.454	0.665002
pond	0.28	0.680342
barge	0.3	0.696146
blinker	0.7	0.975632

4 Concluding Remarks

We have proposed a simple algorithm to approximate the critical temperature of the continuous-state Game of Life. The algorithm alternatingly updates cell states on one hand and the temperature parameter on the other hand, such that it avoids overshoot of the temperature. The algorithm uses knowledge about the maximum value of the gradient of the weakest cell; this works correctly because in this case the cell state is the most sensitive to the temperature. In other words, the maximum value of the gradient is needed by the algorithm to avoid decay of patterns under all circumstances.

In our experiments, the critical temperatures for the beehive and the blinker were obtained without these patterns loosing their characteristic Game of Life shapes.

References

1. Flocchini, P., Geurts, F., Mingarelli, A., Santoro, N.: Convergence and aperiodicity in fuzzy cellularautomata: revisiting rule 90. Physica D 142, 20–28 (2000)
2. Ermentrout, G.B., Keshet, L.E.: Cellular Automata Approaches to Biological Modeling. J. Theor. Biol. 160, 97–133 (1993)
3. Adachi, S., Peper, F., Lee, J.: The Game of Life at Finite Temperature. Physica D 198, 182–196 (2004)
4. Kaneko, K.: Lyapunov analysis and information flow in coupled map lattices. Physica D 23, 436–447 (1986)
5. Ding, M., Yang, W.: Stability of synchronous chaos and on-off intermittency in coupled map lattices. Phys. Rev. E 56, 4009–4016 (1997)
6. Berlekamp, E.R., Conway, J.H., Guy, R.K.: Wining Ways For Your Mathematical Plays. Academic Press, New York (1982)
7. Poundstone, W.: The Recursive Universe. Morrow, New York (1985)
8. Schiff, J.L.: Cellular Automata: A Discrete View of the World. Wiley & Sons (2007)
9. Ilachinski, A.: Cellular Automata. World Scientific Publishing, Singapore (2001)
10. Wolfram, S.: Cellular Automata and Complexity. Addison-Wesley, Reading (1994)
11. Langton, C.G.: Computation at the edge of chaos: phase transitions and emergent computation. Physica D 42, 12–37 (1990)
12. Ninagawa, S., Yoneda, M., Hirose, S.: $1/f$ fluctuation in the Game of Life. Physica D 118, 49–52 (1998)
13. Alstrøm, P., Leão, J.: Self-organized criticality in the “game of Life”. Phys. Rev. E 49, 2507–2508 (1994)
14. Blok, H.J., Bergersen, B.: Effect of boundary conditions on scaling in the “game of Life”. Phys. Rev. E 55, 6249–6252 (1997)