International Journal of Computational Science
1992-6669 (Print) 1992-6677 (Online) www.gip.hk/ijcs

2010, Vol. 4, No. 5, 453-462.

Amoeba-Based Knowledge Discovery

Toshinori Munakata[1]*, Masashi Aono[2], Masahiko Hara[2]

[1] Computer and Information Science Department, Cleveland State University,
Cleveland, OH, 44115, USA
t.munakata@csuohio.edu
[2] RIKEN-HYU Collaboration Research Center, RIKEN Advanced Science Institute,
Wako, Saitama 351-0198, Japan
masashi.aono@riken.jp

Abstract. We propose an amoeba-based knowledge discovery or data mining system, that is implemented using an amoeboid organism and an associated control system. The amoeba system can be considered as one of the new non-traditional computing paradigms, and it can perform intriguing, massively parallel computing that utilizes the chaotic behavior of the amoeba. Our system is a hybrid of a traditional knowledge-based unit implemented on an ordinary computer with an amoeba-based search unit and an optical control unit interface. The solutions in our system can have one-to-one mapping to solutions of other well-known areas such as neural networks and genetic algorithms. This mapping feature allows the amoeba to use and apply techniques developed in other areas. Various forms of knowledge discovery processes are introduced. Also, a new type of knowledge discovery technique, called "autonomous meta-problem solving," is discussed.

Keywords: amoeba-based computing, knowledge discovery, data mining, new computing paradigm.

* Corresponding Author. Email: t.munakata@csuohio.edu.

I Introduction

Knowledge discovery - the notion of computers automatically finding useful information is an exciting and promising aspect of any application intended to be of practical use [1]. There are several closely related areas to knowledge discovery called by different names. *Data mining* or *knowledge discovery in databases* (*KDD*) primarily focus on extracting useful information from data, particularly from a large amount of data. *Machine learning* explores techniques to make the machine learn or get smarter by itself.

New computing paradigms - For the past 40 years computer hardware has been dominated by the traditional CMOS (Complementary Metal-Oxide Semiconductor) or silicon-based integrated circuits (so-called "silicon-based architecture"). Recently, computer architecture concepts based on totally new principles other than the silicon-based technology have been given much attention. These concepts include quantum, atomic (e.g., carbon nanotube transistors), molecular (e.g., organic), DNA, optical, micro/nanofluidic and amoeba-based computing [2]. This article proposes a knowledge discovery scheme employing an amoeba-based system, one of the new computing paradigms.

2 Amoeba-Based Computing

A plasmodium of a true slime mold *Physarum Polycephalum* (Fig. 1a), a unicellular amoeboid organism with a single gel layer (cellular membrane) encapsulating intracellular sol, can be regarded as a kind of massively parallel computer whose elements are microscopic actomyosins (fibrous proteins) taking contracting or relaxing states. Collectively interacting actomyosins in the gel layer generate rhythmic contraction-relaxation oscillation (period = 1~2 min) of vertical body thickness, and their spatiotemporal oscillation pattern induces horizontal shuttle-streaming of intracellular sol (velocity=~1 mm/sec) to deform the macroscopic shape. Despite its homogeneous and decentralized structure, the amoeba exhibits integrated computational capacities in its shape deformation [3].

In [4-8], Aono, *et al*. showed that such a system can be implemented as a chaotic neurocomputer. Because of its slow processing speed, it is not proposed as a high-speed alternative to replace traditional silicon-based technology, but it is interesting from a scientific point of view for the following reasons: (1) It is the first actual, non-silicon based implementation of a chaotic neuron model; (2) It exhibits an interesting problem solving capability in which speed may not be an issue; (3) There are many chaotic phenomena in nature such as lasers and certain properties observed in atoms and molecules. The dynamic speed of these phenomena is very fast; some can easily surpass their current silicon-based counterparts. When the problem solving techniques in this scheme are realized in these areas, they could lead to a new fast computing paradigm. Additional unique features of amoeba-based computing are discussed below.

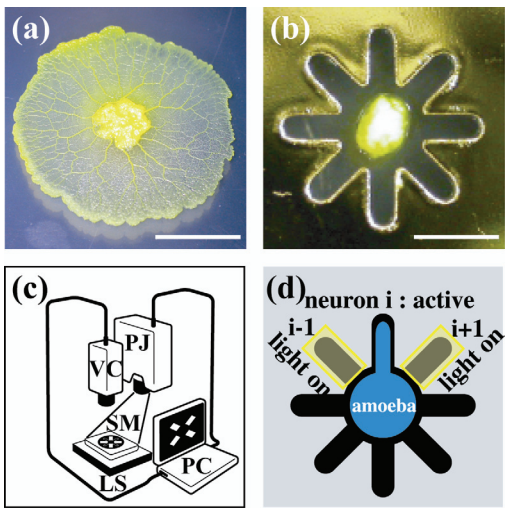

Fig. 1. (a) A true slime mold amoeba. (b) An Au-coated barrier resting on an agar plate. The amoeba restricts itself inside the barrier where the agar is exposed because of its aversion to metal surfaces. (c) Experimental setup. For transmitted light imaging using a video camera (VC), a surface light source (LS) placed beneath the sample amoeba (SM) was used to emit light of a specific wavelength, which did not affect the amoeba's behavior. The recorded image was processed using a personal computer (PC) to visualize the monochrome image by using a projector (PJ). (d) Optical feedback rule: active state $x_i(t) = 1$ triggers light illumination $y_{i-1}(t + 1) = y_{i+1}(t+1) = 1$ (white light projected to yellow rectangular regions).

3 Structure and Implementation

System configuration

An amoeba-based knowledge discovery system consists of three major units, called the Amoeba-Based Search Unit (Fig. 1b and SM in Fig. 1c), Silicon-Based KD (Knowledge Discovery) Unit (VC and PC in Fig. 1c), and Optical Control Interface (PJ in Fig. 1c). The Amoeba-Based Search Unit can represent, for example, a configuration < 1, 0, 0, 1, 0, 0, 1, 0 > at a specific time. The Silicon-Based KD Unit is the command center of the entire system. It knows the target problem and a basic strategy for finding a solution, and gives guidance to the amoeba through the Optical Control Interface. A key element of the system is the Amoeba Unit that searches for a solution in a unique fashion.

Representation of a configuration and a solution

In this article, a state represented by the amoeba at a specific time, such as < 1, 1, 0, 1, 0, 0, 1, 0 >, is called a "configuration." When a configuration sufficiently resolves the problem condition, it is called a "solution."

Knowledge discovery process

Some well known basic types of procedures by which knowledge discovery is performed are: classification, clustering, association, pattern recognition, and control. To perform these types of procedures, specific techniques are employed. Neural networks, genetic algorithms and statistical approaches are some well known techniques. Further, these techniques are generally variations of basic processes, such as optimization [9].

We note that many attribute-based symbolic forms of knowledge representation and data mining employ the above type of configuration [9]. For example, a symbolic form of a rule: "if there is no headache, the temperature is high, and a cough exists, then there is a cold" can be coded as < 0, 1, 1, 1 >, where the first 0 represents "there is no headache."

The amoeba can represent configurations and solutions of neural networks, genetic algorithms, and so on. Amoeba-based systems have also successfully solved many types of problems involving the above-mentioned processes such as optimization and constraint satisfaction. They include the traveling salesman problem (TSP) and arranging 1s and 0s to satisfy the logical NOR function. The core of this paper is to put all the above together. Summarizing, typical steps of an amoeba-based knowledge discovery system are as follows:

- Given a specific application problem, select a basic type of procedure by which knowledge discovery is performed, e.g., classification.
- Select a technique to be employed, e.g., neural networks, and a representation of a solution, e.g., a string of bits.
- Identify the type of process to be performed, e.g., optimization.
- Consider representing each solution by a geometric configuration of the amoeba. They should have one-to-one correspondence. The coding of the geometric configuration of the amoeba, e.g., what it means in terms of the original problem, is understood by the silicon-based KD unit, not by the amoeba.
- Implement a knowledge discovery algorithm, either previously developed for the technique (e.g., backpropagation) or for a new one, in the silicon-based KD unit. Devise an appropriate optical control scheme to drive the amoeba to search for a solution.

Given a target application problem, an amoeba is placed, and its geometric configuration is determined. Then, the current configuration is fed back to the KD unit. Next, the unit determines a desirable direction the amoeba should take and sends this information to the amoeba through the optical control interface. The amoeba evolves to a new configuration, partially on its own spatio-temporal dynamics and partially under the guidance of the optical stimulation. This leads to an intriguing and unique computing paradigm. In the following, we show some examples to illustrate amoeba-based knowledge discovery systems.

4 Specific Forms of Amoeba-Based Knowledge Discovery Systems

A. *Knowledge Discovery by Means of Constraint Satisfaction Problem Solving*

A small piece of the amoeba (0.75 ± 0.05 mg) cut from an individual acts only inside the star-shaped barrier structure on an agar plate (Fig. 1b) by expanding or shrinking its multiple branches. During our experiment, the amoeba's total volume is kept almost constant. The amoeba's branch expands or shrinks at a velocity of at most 1 cm/h, as shuttlewise efflux-influx of the sol (velocity = 1 mm/s) for the branch is iterated for several periods of the contraction-relaxation oscillation (one period = 1–2 min).

We call the *i*th path of the star-shaped structure "neuron *i*." Digital image processing is used to evaluate the state of each neuron, with the amoeba's shape being captured at specific time intervals (t = 6 s) using a video camera (Fig. 1c). For each neuron i at time t, if more than a quarter of the area of the ith neuron is occupied by the amoeba's branch, then state 1 (active) is assigned as x_i (t) = 1, otherwise x_i (t) = 0 (inactive). To start the computing, it is possible to input the arbitrary initial configuration $x_1(0)$, x_2 (0), ⋯ , $x_N(0)$, because the amoeba's shape is freely deformable.

We develop a simple example of solving a constraint satisfaction problem to illustrate the basic idea of our systems. The problem can be described as follows: We work on a cyclic 8-bit string, $<x_i, i = 1, 8>$; the problem is to find a string that satisfies the logical NOR function; more specifically, x_i = NOR(x_{i-1}, x_{i+1}), i = 1, 8. For example, < 1, 1, 0, 1, 0, 0, 1, 0 > is a configuration but not a solution, and < 1, 0, 0, 1, 0, 0, 1, 0 > is a solution. We can consider this as a kind of an optimization problem. As in genetic algorithms, we can define a fitness function f as the number of bits that satisfy the condition x_i = NOR(x_{i-1}, x_{i+1}). The problem then is to maximize the fitness function; a configuration with f = 8 is a solution. We see that the capabilities of our amoeba-based systems can translate to knowledge discovery. For example, most knowledge discovery processes in genetic algorithms employ this form. The size of the string can be scaled up, or each x_i can assume a fractional value on [0, 1].

In the following, we describe more about our experiment for the problem [4]. We write y_i = 1 when the illumination for node i is turned On, whereas y_i = 0 represents that the illumination is turned Off. The optical control unit automatically updates the illumination according to a certain rule. Here, we introduce the following rule for updating the illumination at 6 sec intervals (Fig. 1d): The node i is illuminated ($y_i(t+1)$ = 1) to be inactive ($x_i(t+1)$ = 0), if at least one of its adjacent nodes is active ($x_{i-1}(t)$ = 1 or $x_{i+1}(t)$ = 1); otherwise ($x_{i-1}(t) = x_{i+1}(t)$ = 0) nonilluminated ($y_i(t+1)$ = 0) to be active ($x_i(t+1)$ = 1). This rule establishes the above-mentioned constraint satisfaction problem: Find the system configuration $< x_1, x_2, \ldots, x_8 >$ such that all nodes satisfy x_i = NOR(x_{i-1}, x_{i+1}).

With respect to solutions to this problem, there are 10 configurations consisting of rotation symmetries of < 1, 0, 1, 0, 1, 0, 1, 0 > and < 1, 0, 0, 1, 0, 0, 1, 0 >, that are expected to be stably maintained. The reason is that amoeba taking one of these configurations is no longer forced to reshape by illumination and can terminate the expansion of its branches inside all nonilluminated nodes. Any configuration can be clearly judged as a solution, distinguished from a transient state, if and only if all nodes satisfy the following condition: $y_i(t+1) = 1 - x_i(t+1)$.

It should be noticed that concurrent processing of the circularly connected NOR-operators, analogous to Dijkstra's "dining philosophers problem", entails deadlock-like unsolvability of the problem when all operations are executed in a synchronous manner [3]. Suppose that all branches expand or shrink with a uniform velocity. From the initial configuration <0, 0, 0, 0, 0, 0, 0, 0> evoking no illumination, the synchronous growth movements of all branches will lead to <1, 1, 1, 1, 1, 1, 1, 1> in which all neurons are illuminated. Then, all branches shall shrink uniformly to evacuate from the illuminations, until they reach the initial configuration allowing them to expand again. In this manner, the system can never reach a solution, as the synchronous movements result in perpetual oscillation between <0, 0, 0, 0, 0, 0, 0, 0> and <1, 1, 1, 1, 1, 1, 1, 1>. The synchronous movements would be inevitable, if the amoeba's oscillatory behavior could only produce periodic spatiotemporal patterns with circular symmetry. However, our system can actually solve the problem, because the amoeba produces chaotic oscillatory behavior involving spontaneous symmetry breaking.

The experiment was started from the initial configuration <0, 0, 0, 0, 0, 0, 0, 0> by placing the amoeba's spherical piece at the center of the star-shaped structure as shown in Fig. 1b. In the early stages, the spherical amoeba flattened into a disc-like shape and expanded its thin periphery horizontally with circular symmetry.

The symmetry, however, was spontaneously broken as a defective site was created in the periphery. The defective site was exclusively expanded at a velocity relatively larger than that of the rest of the periphery and developed into a distinct branch having enough occupying area inside neuron 8 to evoke illuminations, as shown in Fig. 2a. Owing to the asynchronously fluctuated movements, the amoeba reached the solution shown in Fig. 2b. The solution was stably maintained for about 4 h. This result implies that our amoeba-based computer can surely perform the circularly-connected NOR functions, as well as other arbitrary logic functions [7]. We can also see that our amoeba-based system can solve problems involving logic. Logic is well known to represent the basic process of the brain functions; Prolog – PROgramming in LOGic is a major AI language.

Interestingly, the long-maintained stabilizing mode of the solution was spontaneously switched to the destabilizing mode even though no explicit external perturbation was applied. As shown in Fig. 2c, two branches suddenly emerged and started to invade illuminated regions, contrary to their photoavoidance response. While aggressive expansion of branch 7 was sustained under illumination, branch 8 was shrunk by illumination, and the once-reached solution was destabilized.

After the branches performed their asynchronously fluctuated movements, the amoeba stabilized its shape again at another solution, as shown in Fig. 2d. The second solution was maintained for about 1 h. Fig. 2e shows that the spontaneous destabilization occurred once more. This consequently led the amoeba to search for the third solution, which was maintained for about 7 h until we quit the observation, as shown in Fig. 2f. During this 16 h experimental trial, the amoeba successively searched three solutions and broke through the deadlock by repeatedly switching between the stabilizing and destabilizing modes.

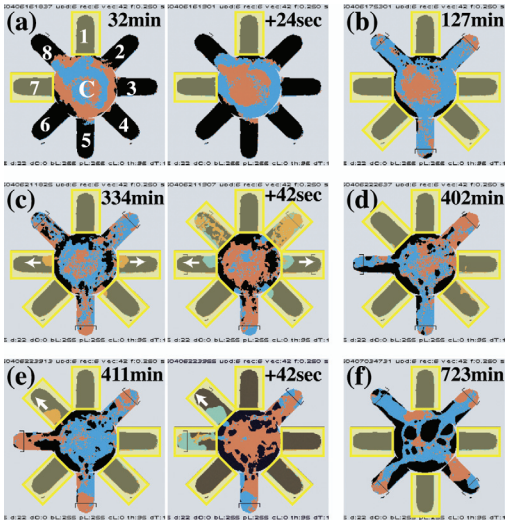

Fig. 2. Experimentally observed problem-solving process. (a) Transient configuration <0,0,0,0,0,0,0,1>. White light was projected to yellow rectangular regions (No 1 and No. 7). By means of digital image processing, the phase of vertical thickness oscillation was binarized into the relaxing (thickness increasing) and contracting (decreasing) states, represented by the blue and red pixels, respectively. Phase wave propagated from the center to periphery with symmetry breaking. (b) First-reached solution <0,1,0, 0,1,0,0,1>. (c) Spontaneous destabilization of solution (b). Arrows indicate the growth directions of the newly emerged branches expanding under illumination contrary to photoavoidance. (d) Secondreached solution <0, 1, 0, 0, 1, 0, 1, 0>. (e) Spontaneous destabilization. (f) Third-reached solution < 0, 1, 0, 1, 0, 1, 0, 1 >.

B. *Knowledge Discovery by Means of Optimization*

As mentioned earlier, many knowledge discovery systems are based on optimization processes. The Traveling Salesman Problem (TSP) is a well known, hard optimization problem. We have examined whether our amoeba-based system is capable of solving the four-city TSP, and found that the system reached an optimal solution with a high probability. In this system, we apply the well-known neural network algorithm developed by Hopfield and Tank [9]. We have confirmed that our amoeba-based neurocomputer has high optimization capability in solving TSP [5] and that the amoeba might be characterized as a set of coupled chaotic oscillators [6]. In theoretical models of coupled chaotic neurons, it has already been shown that chaotic dynamics is highly efficient for solving combinatorial optimization problems [1, 8].

In our amoeba-based TSP solving, each node of a configuration represents a degree of visiting; it is either 1, 0 or between. Again, such a configuration can be represented by directly utilizing the continuous values of amoeba. Such an approach may be effective for an extended TSP, possibly involving various conditions and constraints. For example, cities should not necessarily be geographical locations, but instead represent some set of tasks in broad sense. Each task may be car-

ried out 100%, 0%, or partially somewhere between. We will drive the configuration to our desirable direction through the optical control unit.

C. *Knowledge Discovery Based on Neural Network Models*

Many neural network models, which aim at simulating the human brain, are applied as various forms of data mining techniques. Here we discuss the backpropagation model, the one that has been employed most extensively for industrial and commercial applications. As a simple special case, we consider a perceptron (i.e., no hidden layers) with three input layer neurons and one output layer neuron. This case can be extended to more general neural networks by employing the same principle. Our problem can be described as follows: Let three input layer neurons be $< x_1, x_2, x_3 >$. For each combination of $< x_1, x_2, x_3 >$, we are given a target output value of t. We can also determine a computed output value y for each combination of $< x_1, x_2, x_3 >$ as follows: y = 0 if s = $w_1x_1 + w_2x_2 + w_3x_3 < 0.5$; y = 1 otherwise. The problem is to determine $< w_1, w_2, w_3 >$ to maximize the number of matches between t and y for all given combinations of $< x_1, x_2, x_3 >$. So-called guided random search techniques, such as neural networks and genetic algorithms, have been applied to solve this type of machine learning problem [9, 12]. Our amoeba-based system can represent and determine solutions of the problem. The basic idea is to represent $< w_1, w_2, w_3 >$ as an amoeba configuration, and search for a solution through the optical control interface under the guidance of the knowledge discovery unit.

D. *Knowledge Discovery by Means of Autonomous Meta-Problem Solving*

This is a totally new concept of a humanoid-like problem solving technique. It can be illustrated by the TSP discussed earlier. Ordinary TSP solving is applied to a fixed problem, i.e., the number of cities and the distances among the cities never change. Our amoeba-based system can not only solve the ordinary TSP but can also modify the original problem and solve the new ones [5]. In the new problem, the number of cities and the distances may be changed. What is the significance of such technique? In TSP, we can think of an early planning stage where no highways or airplane routes are established. We set up a tentative initial layout and determine an optimal solution. Our system extends this initial layout, bringing forth new problems (maps) and solutions, as well. Adding new cities is similar to the Steiner-tree graph problem. We can select the best layout with the associated best solution.

The most important point here is not a specific application to the TSP described above. Rather, the primary concept is that, given a problem (e.g., a TSP), creating new problems (e.g., new TSPs) and solving them. That is, we are solving a higher-level problem above the original lower-level problem; this is why we use the term "meta-problem solving." Further, this is done automatically by amoeba without human intervention; this is why the term "autonomous."

It is conceivable that this technique can be applied to many domains. For attribute or parameter based systems, our system can suggest adding or deleting some attributes or parameters leading to new problems. As another example, we can consider designing a walking robot. The objective is to design a robot having good performance in terms of control stability, speed and cost. Our origi-

nal robot may have two legs, following our common way of thinking about humans. Our system may suggest a three or four leg robot that would give better overall performance than the original design.

5 Conclusions

A biological organism, such as the amoeba, is a hierarchically structured system in which a number of self-organization processes run simultaneously on their characteristic spatiotemporal scales at multiple levels. The multiple levels are: the molecules, genes, proteins, cells, tissues, organs, and body parts, as well as the whole body. Because the self-organization process at each level involves a certain kind of benefit optimization, such as energy minimization and stability maximization, it would be reasonable to consider the organism as a particular kind of concurrent computing system in which a number of computing processes to solve different benefit optimization problems are executed concurrently by sharing common computational resources such as energies and structured substances. If the multilevel optimization processes are capable of making a self-disciplined decision, for example, a decision to accept a loss in short-term benefits of body parts for the sake of long-term gains of the organism's whole body, the decision capability may be exploited for performing some un-programmed but reasonable operations when incorporated in a bio-computer.

Recently, inspired from the amoeba's photoavoidance response, a powerful parallel search algorithm was developed [14, 15]. The algorithm, called the "Tug-Of-War (TOW) model", is applied to the Multi-armed Bandit Problem (MBP), a problem of finding the most rewarding one from a number of slot machines. To maximize profit, a player drops coins into a bank of machines, evaluating as quickly and correctly as possible which machine wins jackpots with the highest probability. The player, therefore, should make this decision in a trade-off created with incompatible demands: one either exploits the rewards obtained using already collected knowledge or explores new alternatives for acquiring higher payoffs involving risks. The MBP represents the so-called "exploration-exploitation dilemma" that many algorithms for *tree search* face in practical situations. Thus, efficient algorithms for the MBP are useful for a wide range of applications requiring powerful tree search capability.

Among well-known strong algorithms for the MBP, the TOW model exhibited the best performance as the amoeba-like branches perform nonlocally-correlated parallel searches. By extracting the essential dynamics of the amoeba's spatiotemporal oscillatory behavior [16-18], the powerful computing scheme may be implemented by other faster, man-made materials in a massively parallel manner. Placing all the unique features together, the proposed amoeba-based system may open up completely new methods of knowledge discovery and data mining.

References

1. T. Munakata, Ed.: Special Section "Knowledge Discovery," *Commun. ACM* 42 (1999) (11) 26-67.
2. T. Munakata, Ed.: Special Issue "Beyond Silicon: New Computing Paradigms," *Commun. ACM* 50 (2007) (9) 30-72.
3. T. Nakagaki, H. Yamada, A. Toth: Maze-solving by an Amoeboid Organism, *Nature* 407 (2000), pp. 470.
4. M. Aono, Y. P. Gunji: Beyond Input-output Computings: Error-driven Emergence with Parallel Non-distributed Slime Mold Computer, *BioSystems* 71 (2003) 257-287.
5. M. Aono, M. Hara: Spontaneous Deadlock Breaking on Amoeba-based Neurocomputer, *BioSystems* 91 (2008) 83-93.
6. M. Aono, M. Hara, K. Aihara: Amoeba-based Neurocomputing with Chaotic Dynamics, *Commun. ACM* 50 (2007) (9) 69-72.
7. M. Aono, M. Hara, K. Aihara, T. Munakata: Amoeba-based Emergent Computing: Combinatorial Optimization and Autonomous Meta-problem Solving, *International Journal of Unconventional Computing* 6 (2010) (2) 89-108.
8. M. Aono, Y. Hirata, M. Hara, K. Aihara: Amoeba-based Chaotic Neurocomputing: Combinatorial Optimization by Coupled Biological Oscillators, *New Generation Computing* 27 (2009) 129-157.
9. T. Munakata: *Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More*, 2nd ed., London: Springer. (2008)
10. M. A. Arbib, Ed.: *The Handbook of Brain Theory and Neural Networks*, 2nd ed., Cambridge, Massachusetts: The MIT Press. (2003)
11. J. J. Hopfield, D. W. Tank: Computing with Neural Circuits: A Model, *Science* 233 (1986) 625-633.
12. K. Aihara, T. Takabe, M. Toyoda: Chaotic Neural Networks, *Phys. Lett. A* 144 (1990) 333-340.
13. M. Hasegawa, T. Ikeguchi, K. Aihara: Combination of Chaotic Neurodynamics with the 2-opt Algorithm to Solve Traveling Salesman Problems, *Phys. Rev. Lett.* 79 (1997) 2344-2347.
14. S.-J. Kim, M. Aono, M. Hara: Tug-of-war Model for the Two-bandit Problem: Nonlocally-correlated Parallel Exploration Via Resource Conservation, *BioSystems* 101 (2010) 29-36.
15. S.-J. Kim, M. Aono, M. Hara: Tug-of-war Model for Multi-armed Bandit Problem, Unconventional Computation 2010, *Lecture Notes in Computer Science,* vol. 6079, Springer, 2010, pp.69-80..
16. M. Aono, Y. Hirata, M. Hara, K. Aihara: Resource-competing Oscillator Network as a Model of Amoeba-based Neurocomputer, Unconventional Computation 2009, *Lecture Notes in Computer Science,* vol. 5715, Springer, 2009, pp.56-69.
17. M. Aono, Y. Hirata, M. Hara, K. Aihara: A Model of Amoeba-based Neurocomputer, *Journal of Computer Chemistry*, Japan vol. 9 (3), 2010, pp.143-156.
18. Y. Hirata, M. Aono, M. Hara, K. Aihara: Spontaneous Mode Switching in Coupled Oscillators Competing for Constant Amounts of Resources, *Chaos* 20 (2010), pp.013117.