2010 Third International Joint Conference on Computational Science and Optimization

Amoeba-Based Knowledge Discovery System

Toshinori Munakata
Computer and Information Science Department,
Cleveland State University,
Cleveland, OH, 44115, USA
t.munakata@csuohio.edu

Masashi Aono and Masahiko Hara
RIKEN-HYU Collaboration Research Center,
Advanced Science Institute, RIKEN,
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 *k*nowledge-based unit implemented on an ordinary computer and an amoeba-based search unit, with an interface of an optical control unit. 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

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].

New computing paradigms - For the past 40 years computer hardware has been dominated by the traditional CMOS 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 [2]. This article proposes a knowledge discovery scheme employing an amoeba-based system, one of the new computing paradigms.

II. 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 implement a chaotic neurocomputing. 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 the 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.

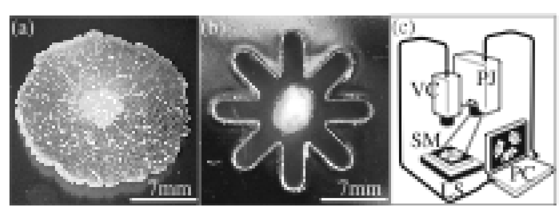

Figure 1. (a) A true slime mold amoeba. (b) An Au-coated barrier resting on an agar plate. The amoeba restricts itself inside the barrier where the agar is exposed because of its aversion to metal surfaces. (c) Experimental setup. For transmitted light imaging using a video camera (VC), a surface light 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).

978-0-7695-4030-6/10 $26.00 © 2010 IEEE
DOI 10.1109/CSO.2010.88

III. 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 suffices 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 spatiotemporal 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.

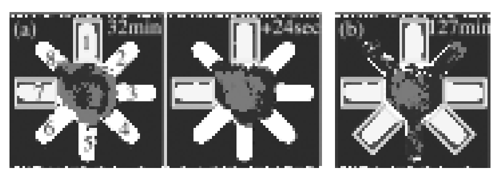

Figure 2. Experimentally observed problem-solving process. (a) Transient configuration <0,0,0,0,0,0,0,1>. Whitelight was projected to white rectangular regions (No 1 and No. 7). By means of digital image processing, the phase of vertical thickness oscillation was binarized into the relaxing (thickness increasing) and contracting (decreasing) states, represented by the black and gray 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>.

IV. 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. We show 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, $x_i = \mathrm{NOR}(x_{i-1}, x_{i+1})$, $i = 1, 8$. 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 $x_i = \mathrm{NOR}(x_{i-1}, x_{i+1})$. The problem then is to maximize the fitness function; a configuration with $f = 8$ is a solution. We see that such capability 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 in a real interval [0.0, 1.0].

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: 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 $x_i = \mathrm{NOR}(x_{i-1}, x_{i+1})$.

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 [5]. 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. Fig. 2 shows some results after this initial configuration. The solution (Fig. 2b) was stably maintained for about 4 h. This result implies that our amoeba-based computer can surely perform the connected NOR functions, and so other arbitrary logic functions [10].

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 if 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 a modified version of the well-known neural network algorithm developed by Hopfield and Tank [11]. We have confirmed that our system has high optimization capability in solving TSP [7] and that the amoeba might be characterized as a set of coupled chaotic oscillators [8]. In theoretical models of coupled chaotic neurons, it has already been shown that chaotic dynamics is highly efficient for solving combinatorial optimization problems [12,13].

C. Knowledge Discovery Based on Neural Network Models

A set of neurons and/or associated weights can be represented by an amoeba configuration. The same principle employed in the above examples then can be applied to the neural network models.

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

This is a totally new concept for a humanoid-like problem solving technique. For example, 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 also modify the original problem and solve the new ones [7].

V. 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 [8].

Additionally, amoeba-based computing can exercise positive and negative feedback together with the amoeba's adaptability to find new ways to survive under unexpected environments [7]. By extracting the essential dynamics of the multilevel optimization processes [14,15], the computing scheme may be implemented by other faster materials capable of multilevel self-organization. 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, vol. 42 (11), 1999, pp. 26-67.

[2] T. Munakata, Ed, Special issue "Beyond silicon: new computing paradigms," Commun. ACM, vol. 50 (9), 2007, pp. 30-72.

[3] T. Nakagaki, H. Yamada, and A. Toth, "Maze-solving by an amoeboid organism," Nature, vol. 407, 2000, pp. 470.

[4] M. Aono and Y. P. Gunji, "Beyond input-output computings: Error-driven emergence with parallel non-distributed slime mold computer," BioSystems, vol. 71, 2003, pp. 257-287.

[5] M. Aono and M. Hara, "Spontaneous deadlock breaking on amoeba-based neurocomputer," BioSystems, vol. 91, 2008, pp. 83-93.

[6] M. Aono, M. Hara and K. Aihara, "Amoeba-based neurocomputing with chaotic dynamics," Commun. ACM, vol. 50 (9), 2007, pp. 69-72.

[7] M. Aono, M. Hara, K. Aihara, and T. Munakata, "Amoeba-based emergent computing: combinatorial optimization and autonomous meta-problem solving," International Journal of Unconventional Computing, in press.

[8] M. Aono, Y. Hirata, M. Hara, and K. Aihara, "Amoeba-based chaotic neurocomputing: combinatorial optimization by coupled biological oscillators," New Generation Computing, vol. 27, 2009, pp. 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 and D. W. Tank, "Computing with neural circuits: a model," Science, vol.233, 1986, pp. 625-633.

[12] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," Phys. Lett. A, vol. 144, 1990, pp. 333-340.

[13] M. Hasegawa, T. Ikeguchi, and K. Aihara, "Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems," Phys. Rev. Lett., vol.79, 1997, pp. 2344-2347.

[14] M. Aono, Y. Hirata, M. Hara, and 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.

[15] M. Aono, Y. Hirata, M. Hara, and K. Aihara, "A model of amoeba-based neurocomputer," Journal of Computer Chemistry, Japan, in press.