Amoeba-Based Fuzzy Computing for Uncertain Knowledge Processing

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

Masashi Aono[‡] **Masahiko Hara**[‡]
‡ Emergent Functions Asian Collaboration Lab., RIKEN (The Institute of Physical and Chemical Research), Wako, Saitama 351-0198, Japan

Abstract

Fuzzy systems have extensively been applied to process uncertain information and knowledge, and have mostly been implemented on traditional silicon based chips or electronic circuits. This paper proposes a new, non-silicon based paradigm called amoeba-based computing. In this scheme, fuzzy systems, such as fuzzy logic (e.g., inference) and control, will be implemented by means of behavior of an amoeboid organism and an associated control system. Recently there has been a growing interest in amoeba-based, massively parallel computing that utilizes chaotic behavior of the amoeba. Amoeba-based fuzzy computing inherits advantages of both amoeba-based computing and fuzzy systems, and provides unique computing capability of uncertain information and knowledge.

Keywords: New computing paradigm, amoeba-based computing, evolvable chaotic fuzzy architecture.

1 Introduction

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. While practical, everyday-use of these proposals as computing devices is yet to be seen, these ideas have stimulated the scientific community due to their fundamental nature, novelty, and potentials for new forms of information processing and applications. These concepts include quantum, atomic (e.g., carbon nanotube transistors), molecular (e.g., organic), DNA, optical, micro/nanofluidic and amoeba-based computing [11]. Some of these new technologies are hoped to complement or replace the current computer architecture based on silicon chips. This paper considers implementing fuzzy systems on the platform of an amoeboid organism. Details of amoeba-based computing and its applications to fuzzy systems are discussed in this paper.

Fuzzy systems

Born in the United States around 1965 [14], fuzzy set theory has grown to a major scientific domain collectively called "fuzzy systems," which include fuzzy sets, logic, algorithms, control, and others. The most fundamental characteristic of a fuzzy system is that it allows a gradual and continuous transition, say, from 0 to 1, rather than a crisp and abrupt change between binary values of 0 and 1. Fuzzy systems are suitable for uncertain or approximate reasoning, especially for the system whose mathematical model is hard to derive. Fuzzy logic allows decision making with estimated values under incomplete or uncertain information. Fuzzy system approaches

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 792–797
Torremolinos (Málaga), June 22–27, 2008

also allow us to represent descriptive or qualitative expressions, and are easily incorporated with symbolic statements. These expressions and representations are more natural than mathematical equations for many human judgmental rules and statements [9,10].

The hardware implementations of fuzzy systems have been extensive, both in digital and analog forms. Because of the prevalence of digital computing in general, digital fuzzy system implementations are dominated by silicon-based chips. Since many fuzzy systems deal with continuous values, analog implementations are often the most reasonable choice and they are typically realized by electronic circuits.

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 $\simeq$ 1 to 2min) of vertical body thickness, and their spatiotemporal oscillation pattern induces horizontal shuttle-streaming of intracellular sol (velocity $\simeq$ 1 mm/sec) to deform the macroscopic shape. Despite its homogeneous and decentralized structure, the amoeba exhibits integrated computational capacities in its shape deformation [12].

We employ the amoeba as it can freely change its planar shape only inside the stellate barrier structure on an agar plate (Fig. 1B), and experimentally implement a model in which its state transition is represented by the amoeba's photoavoidance-based shape deformation under optical feedback control [2]. The i th lane of the stellate structure, called "node i" ($i \in \{1, 2, \cdots, 8\}$), takes the active state $x_i = 1$ whenever the fraction of the area occupied by the amoeba's branch exceeds a threshold value of 1/4, otherwise the inactive

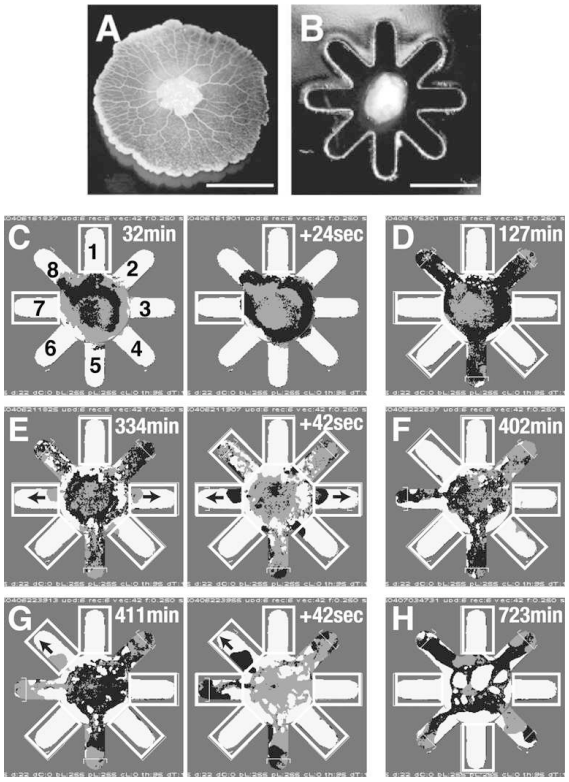

Figure 1: (A) An individual amoeba of a true slime mold (scale bar = 10 mm). (B) Barrier structure on agar plate without nutrients (scale ba r= 2 mm). A small piece of the amoeba cut from a huge one can survive as a complete individual even without food supply for up to about a week, because it can store nutrients fed before the experiment as internal energy source. Initial configuration $< 0, 0, 0, 0, 0, 0, 0, 0 >$ is input by placing the individual amoeba at the center. (C) Transient configuration $< 0, 0, 0, 0, 0, 0, 0, 1 >$. White light was projected to white rectangular regions. The amoeba's oscillation phases are binarized as black and gray for relaxing (thickness increasing) and contracting (decreasing) states, respectively. Phase wave propagates from the center to periphery with symmetry breaking. (D) First-reached solution $< 0, 1, 0, 0, 1, 0, 0, 1 >$ (duration $\simeq$ 4 h). (E) Spontaneous destabilization of solution D. Arrows indicate the growth directions of newly-emerged branches growing under illumination contrary to photoavoidance. (F) Second-reached solution $< 0, 1, 0, 0, 1, 0, 1, 0 >$ (duration $\simeq$ 1 h). (G) Spontaneous destabilization of solution F. (H) Third-reached solution $< 0, 1, 0, 1, 0, 1, 0, 1 >$ (duration $\simeq$ 7 h).

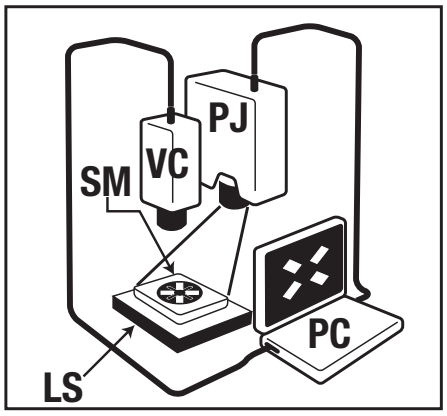

Figure 2: Optical feedback system. For transmitted light imaging using a video camera (VC), the sample circuit (SM) was illuminated from beneath with a surface light source (LS). The recorded image was processed using a PC to update the monochrome image for illumination with a projector (PJ).

state $x_i = 0$. Using image processing technique, the state of each node is measured from a digital image taken by a camera at every 6 sec (Fig. 2). We can inactivate a node by illuminating the corresponding region, because the amoeba's branch in the node shrinks (degenerates) due to its photoavoidance. Conversely, any node is activated naturally if it is not illuminated, as the amoeba inherently tries to expand (grow) all branches to occupy the entire agar region with its total volume kept constant. A branch grows or degenerates (velocity $\simeq$ 1 cm/h) as shuttlewise sol influx-efflux for the branch is iterated.

In [3-5], Aono, et al. showed that such system can implement a chaotic neurocomputing. In this system, each node is interpreted to represent a neuron of an artificial neural network when the optical feedback is applied according to a recurrent neural network algorithm for combinatorial optimization [8]. The state transition of such neuron, the oscillatory behavior of the amoeba's branch in the corresponding lane, is chaotic (i.e., deterministic, nonlinear, and sensitive to its initial condition) due to the amoeba's inherent nature. The usefulness of chaotic dynamics for optimization has already been clarified with chaotic neural network models [1, 7]. Because of its slow processing speed, amoeba-based computing is not proposed as a high-speed alternative to replace traditional silicon-based technology, but it is interesting from a scientific viewpoint 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. Such capability includes deadlock breaking, escaping from local minima for optimization problems, learning and memory [3-5, 13]. (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, it could lead to a new fast computing paradigm.

The basic concept of an amoeba-based computing system can be described as follows. Given a target problem, such as implementation of logical operations or an optimization, we set an appropriate optical feedback scheme. An amoeba is placed, and its geometric configuration that may reach a solution to the problem is measured by the optical feedback system (Fig. 2). The system then determines a desirable direction the amoeba should take and sends this information to the amoeba as light stimulation. The amoeba evolves to a new configuration, partially by its own response and partially under the guidance of the light stimulation. This leads to an intriguing and unique computing paradigm. Uncertain knowledge can be incorporated in the feedback system, interacting with the amoeba.

3 Amoeba-Based Fuzzy Computing

Amoeba-based computing has primarily been applied to non-fuzzy problems by quantizing or discretizing the continuous characteristics of the amoeba [3]. Amoeba-based fuzzy computing implements fuzzy quantities by directly utilizing the amoeba's continuous val-

ues (i.e., the state of each node is represented by a fraction of the area occupied by the amoeba's branch in the corresponding lane). Amoeba-based fuzzy computing is a new computing paradigm that implements fuzzy system functions, such as inference and control, on the platform of the amoeba. When it is realized, it will have advantages inherited from both systems. These include robustness and the ability to deal with hard problems inherited from both systems. In the following we discuss simple examples to illustrate the basic concepts.

4 Application Example - Amoeba-Based Constraint Satisfaction Problem Solving

We show here a simple example of amoeba-based computing — solving a constraint satisfaction problem established by concurrently operated logical NOR functions.

4.1 Problem Solving for Non-fuzzy NOR Functions

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*. Optical feedback system 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 following constraint satisfaction problem: Find the system configuration $< x_1, x_2, \cdots, x_8 >$ such that all nodes satisfy $x_i = NOR(x_{i-1}, x_{i+1})$.

As the solutions of 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 because the 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 gdining philosophers problemh, entails deadlock-like unsolvability of the problem when all operations are executed in a synchronous manner [2]. 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 (Fig.1B), 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, as shown in Figures 1C-H, our system can actually solve the problem, because the amoeba produces chaotic oscillatory behavior involving spontaneous symmetry breaking.

The symmetry-broken oscillation pattern (Fig.1C) yields mutual time lags among movements of branches and decides which branches should exclusively expand. Owing to this asynchronously fluctuating movements, the system first reached and stably maintained a solution (Fig.1D). This result implies that our amoeba-based computer can surely perform the connected NOR functions, and so other arbitrary logic functions [6]. Intriguingly, the stabilizing mode of the first solution was maintained for a long time, but was spontaneously switched to the destabilizing mode without any explicit external perturbation, as two branches newly emerged with localized high oscillation amplitude and

started to invade illuminated regions contrary to photoavoidance (Fig.1E). While aggressive expansion of the branch 7 was sustained under illumination, the branch 8 was shrunk by illumination, and the first solution eventually evolved into another solution (Fig. reffigure1F). Then the spontaneous destabilization occurred again (Fig.1G), and consequently the system achieved the transition among three solutions (Figs. 1D, F and H) within 16 h.

4.2 Problem Solving for Fuzzy NOR Functions

We can consider a fuzzy version of the above NOR function. Let A and B be propositional variables. In ordinary (non-fuzzy) logic, each of A and B can assume 0 (*False*) or 1 (*True*). In fuzzy logic, each of A and B can assume 0 (*False*), 1 (*True*) or any fractional value between 0 and 1. Although there are several versions to define fuzzy logic functions, the most widely employed one are as follows:

$$\begin{aligned} A\ AND\ B &= min(A, B), \\ A\ OR\ B &= max(A, B), \\ NOT\ A &= 1 - A. \end{aligned}$$

Based on these definitions, we have:

$$\begin{aligned} NOR(A, B) &\equiv NOT(OR(A, B)) \\ &= 1 - max(A, B). \end{aligned}$$

We can implement the fuzzy version of the NOR function in our amoeba-based computing system. A fuzzy propositional variable can be represented by a fractional value of the dimension (area) of the amoeba's branch occupying the corresponding node without considering the threshold. Circularly connecting the above NOR functions, we can establish the fuzzy version of the constraint satisfaction problem requiring all nodes to satisfy $x_i = NOR(x_{i-1}, x_{i+1}) = 1 - max(x_{i-1}, x_{i+1})$. This problem can be implemented through the optical feedback system as follows:

$$\begin{aligned} y_i(t+1) &= 1 - x_i(t+1) \\ &= 1 - NOR(x_{i-1}(t), x_{i+1}(t)) \\ &= 1 - (1 - max(x_{i-1}(t), x_{i+1}(t))) \\ &= max(x_{i-1}(t), x_{i+1}(t)). \end{aligned}$$

We note that the fuzzy version of the illumination status yi takes a real value as $0 \leq y_i \leq 1$, representing a fraction of the illuminated region in the entire rectangular region in the corresponding lane (i.e., *the area of the bright illumination pattern / the area of the entire lane*) as completely *Off* (0), *On* (1), or somewhere between. Solutions to this problem, for example, $< 0.9, 0.1, 0.9, 0.1, 0.9, 0.1, 0.9, 0.1 >$ and $< 0.9, 0.1, 0.1, 0.9, 0.1, 0.1, 0.9, 0.1 >$, are given in an analogous fashion to that of the non-fuzzy version.

5 Discussion

In conventional paradigm, deadlock is a common problem for arbitrary concurrent processes, and can be avoided in its "software level" if programmers can know about computational resources requested by all processes in advance of coding the resource allocation. In other words, deadlock avoidance is impossible for many cases, because no one can know beforehand all potential requests of the processes. In contrast, our system can flexibly avoid deadlock-like inoperative conditions and can search for reasonable solutions without any resource allocation program, because in its "hardware level" the amoeba can autonomously materialize and alter the resource allocation (i.e., allocation of sol influx-efflux for selecting which branches should expand) in a nonperiodic but nonrandom manner. This unique capability may be advantageous in development of autonomous systems operated in actual environments, such as robot control systems, that should flexibly respond to concurrent occurrences of unexpected events by avoiding inoperative conditions even if the programs prescribed only for expected events are defective or useless.

6 Conclusion

The proposed amoeba-based computing may open up completely new ways of processing uncertain information. In particular, it can lead to solve problems for inference, control and optimization under uncertainty.

References

[1] K. Aihara, T. Takabe, M. Toyoda (1990). Chaotic Neural Networks. *Physics Letter A*, volume 144, pages 333-340.

[2] M. Aono, Y. P. Gunji (2003). Beyond input-output computings: Error-driven emergence with parallel non-distributed slime mold computer. *BioSystems*, volume 71, pages 257-287.

[3] M. Aono, M. Hara, K. Aihara (2007). Amoeba-based Neurocomputing with Chaotic Dynamics. *Communications of the ACM*, volume 50(9), pages 69-72.

[4] M. Aono, M. Hara, K. Aihara, T. Munakata. Amoeba-Based Emergent Computing: Combinatorial Optimization and Autonomous Meta-Problem Solving. *International Journal of Unconventional Computing* (in press).

[5] M. Aono, Y. Hirata, M. Hara, K. Aihara (2007). Combinatorial Optimization by Amoeba-based Neurocomputer with Chaotic Dynamics. In: *Proceedings of the conference IWNC'07, 2nd International Workshop on Natural Computing*, pages 1-11, Nagoya Univ., Japan, December 2007.

[6] M. A. Arbib, Ed (2003). The Handbook of Brain Theory and Neural Networks (2nd Edition). Cambridge, Massachusetts: The MIT Press.

[7] M. Hasegawa, T. Ikeguchi, K. Aihara (1997). Combination of Chaotic Neurodynamics with the 2-opt Algorithm to Solve Traveling Salesman Problems. *Physical Review Letters*, volume 79, pages 2344-2347.

[8] J. J. Hopfield, D. W. Tank (1986). Computing with neural circuits: A model. *Science*, volume 233, pages 625-633.

[9] T. Munakata, Y. Jani (1994). Fuzzy Systems: An Overview. *Communications of the ACM*, volume 37(3), pages 69-76.

[10] T. Munakata (1998, 2008, 2nd Ed.). Fundamentals of the New Artificial Intelligence: Beyond Traditional Paradigms, New York, Springer Verlag.

[11] T. Munakata (2007). Guest Editor's Introduction in the Special Issue gBeyond Silicon: New Computing Paradigms. *Communications of the ACM*, volume 50(9), pages 30-34.

[12] T. Nakagaki, H. Yamada, A. Toth (2000). Maze-solving by an amoeboid organism, *Nature*, volume 407, pages 470.

[13] Tetsu Saigusa, Atsushi Tero, Toshiyuki Nakagaki, Yoshiki Kuramoto. Amoebae Anticipate Periodic Events. *Physical Review Letters*, volume 100, pages 018101-1-4.

[14] L.A. Zadeh (1965). Fuzzy Set. *Information and Control*, volume 8, pages 338-353.