Autonomy-Oriented Computing (AOC): *The Nature and Implications of a Paradigm for Self-Organized Computing*[*]

Jiming Liu

Department of Computer Science
Hong Kong Baptist University, Kowloon Tong, Hong Kong
E-mail: jiming@comp.hkbu.edu.hk

Abstract

Facing the increasing needs for large-scale, robust, adaptive, and distributed/decentralized computing capabilities [1, 5] from such fields as Web intelligence, scientific and social computing, Internet commerce, and pervasive computing, an unconventional bottom-up paradigm, based on the notions of Autonomy-Oriented Computing (AOC) and self-organization in open complex systems, offers new opportunities for developing promising architectures, methods, and technologies. The goal of this paper is to describe the key concepts in this computing paradigm, and furthermore, discuss some of the fundamental principles and mechanisms for obtaining self-organized computing solutions.

1. Introduction

In response to the real-world needs for robust and computationally scalable means of tackling large-scale, dynamically-evolving, and/or highly-distributed computational problems [1, 5], such as those in Web intelligence [19, 48, 51], as well as other applications in scientific and social computing, Internet commerce, and pervasive technologies, various computing ideas and techniques have been studied that explicitly utilize the models of computational autonomy as inspired by nature, and explore their roles in addressing our practical computing needs. This paper focuses on one of such research initiatives, which concerns the development of an unconventional computing paradigm, called Autonomy-Oriented Computing (AOC) [18, 21, 22].

Generally speaking, AOC tackles a computing problem by defining and deploying a system of local autonomy-oriented entities. The entities spontaneously *interact* with their environments and operate based on their behavioral rules. They self-organize their structural relationships as well as behavioral dynamics, with respect to some specific forms of interactions and control settings. Such a capability is referred to, in this paper, as the **self-organized computability** of autonomous entities.

The goal of this paper is to outline the key concepts in the design and development of an AOC system, and in addition, discuss the distinct roles and characteristics of self-organization in the performance of the AOC system.

The rest of the paper is organized as follows: Section 2 describes some of the basic concepts in AOC. Section 3 further presents a summary of the general steps for performing AOC. Section 4 discusses the general principles for achieving self-organized computability in an AOC system. Section 5 revisits some of the earlier AOC-related models. Finally, Section 6 concludes the paper by highlighting the essence, applications, as well as features of autonomy-oriented self-organized computing, and at the same time, pointing out open research issues.

2. Autonomy-Oriented Modeling

In this section, we will take a look at some of the basic concepts in the AOC formulation.

2.1 Entities

In AOC, computation involves defining and deploying a system $\mathbf{A}$ of **autonomous entities** $\mathbf{e}$ that can directly or indirectly interact among themselves as well as with their environments. Here the notion of entities should be taken in a broad sense, encompassing conceptual and/or computational entities.

The entities are considered as not only the elements, but also the subsystems of $\mathbf{A}$; one type of entities in $\mathbf{A}$ can fur-

[*]A Keynote Talk at the 4th International Conference on Natural Computation (ICNC'08) and the 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'08), October 18-20, 2008, Jinan, China.

ther contain other types of entities as their elements. $\mathbf{A}$ exhibits its structural and behavioral complexity, through the coupling relationships as well as the corresponding interactions of its entities at and across various levels, for instance, in the forms of cells, organisms, communities, and societies.

The number of entities at a certain level in $\mathbf{A}$ can also vary during their interactions (e.g., [8]).

In the foregoing discussions, the entities are not meant to be homogeneous; they can be, generally speaking, heterogeneous entities at various levels. Without loss of generality, we will not explicitly indicate their specific levels within $\mathbf{A}$.

2.2 Local Autonomy

Active entities spontaneously interact with, and exert changes to, not only their local environment $\mathbf{E}_l$ that can contain other local (e.g., neighboring) entities, but also the global environment of $\mathbf{A}$, denoted by $\mathbf{E}_g$. In doing so, an individual entity $\mathbf{e}$ in $\mathbf{A}$ can be viewed as an automaton dynamically governed by its behavioral rules, as stored in a repository $\mathcal{R}$, to either deliberatively or reactively operate in its environments. In other words, $\mathcal{R}$ defines the necessary local autonomy of entities, internally and/or externally triggered/influenced, for inducing an emergent solution.

In order to effectively apply its behavioral rules, an individual entity needs to evaluate its current state and utility, as well as the current states of its local and *sometimes* global environments (including the current states of other local entities), by using evaluation functions $\mathcal{F}$ and u. Thereafter, the entity accordingly selects its behaviors b from $\mathcal{B}$. It produces certain effects to itself, and to its local and global environments, in the forms of: (i) state changes, (ii) utility updates (e.g., marking and scoring a local configuration including its state and $\mathcal{F}$ values), and (iii) external operations or interactions that will further influence other entities as well as the computation of the AOC system.

2.3 Coupling Relationships

The forms of input conditions received, as well as output operations exerted, by entities in their environments will be constrained by structural and functional constraint relationships, both among the entities and with their local and global environments. Such constraint relationships are referred to as **coupling**.

For instance, the local behavior of an entity can be directly influenced by, or computed upon, those in its neighborhood by means of local coupling functions. One example would be the coupling relationship that controls a spin alignment with its nearest neighbors as in the Ising-like spin model [15, 40]. The *control parameter* of temperature in this case will determine the *degree* of coupling. In other forms of coupling, a gain can be used to control the degree of local response. Given different local coupling forms and degrees, different collective effects can be reached and observed over different temporal and spatial scales. A certain dominant (or *critical*) behavior can be observed if the gain is set to a critical value.

In general, the forms and degrees of coupling at various levels in $\mathbf{A}$ can be defined using functions $\mathcal{C}$, which are either pre-defined or dynamically adjusted, signifying the evolution of dominant structural and functional relationships (e.g., coordination) among entities. As a result of self-organization based on positive feedback processes, the varying degrees of relatively stable coupling relationships among entities can further dynamically lead to the emergence of self-organized, structurally and functionally stable yet interrelated 'hyper-modules', e.g., behavioral **motifs** [26, 27] and network motifs [35, 36].

Example 2.1 (Social network community mining) *In the case of distributed community mining for social networks [46], as shown in Figure 1, the coupling relationships among entities, e.g., network nodes as in an ad hoc network, will be initially determined based on their local connectivities, as obtained by means of using the local behavior of* `get_view`. *During the course of self-organized community mining, the entities will locally update and reinforce their coupling relationships, through their local behaviors of* `shrink_view` *and* `enlarge_view`*, according to their local communication frequencies.*

2.4 Direct and Indirect Interactions

The emergent complex behavior of entities originates from their local interactions, as *enabled* by their coupling relationships at various levels. There exist two types of interactions, namely, *direct* interactions among entities and *indirect* interactions through the shared environments of entities. Different AOC systems will have different ways of direct and indirect interactions.

Indirect interactions can be achieved based on historically aggregated effects in the environments [21], or through nonlinear feedback between entities and their interacted/updated adaptive landscape, e.g., in information generation and exchange [37]. In such a case, interactions can occur among different entities at the same temporal and/or spatial scale, e.g., at the same time step or in the same environments, as well as spontaneously across different temporal and spatial scales. In AOC, the shared environments will serve not only as the domains for autonomous entities, but also as the media for their indirect interactions.

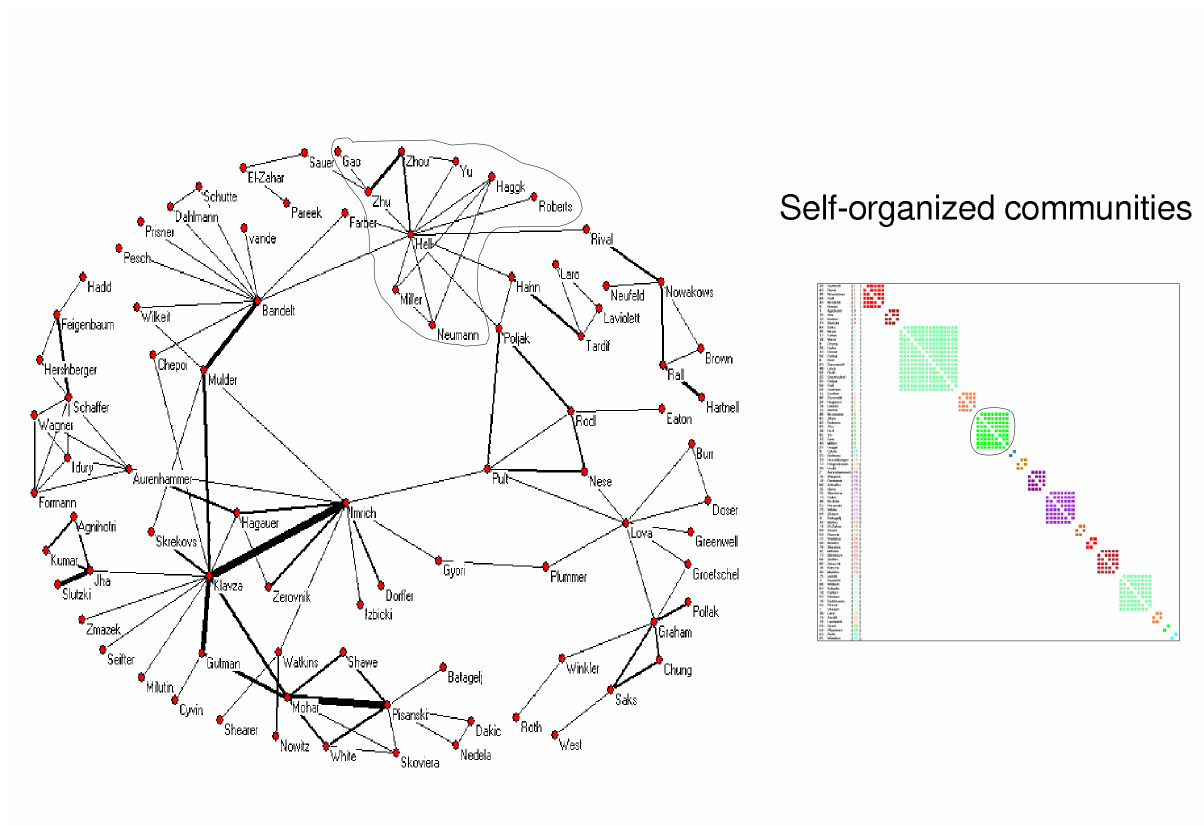

Figure 1. Self-organized community mining in a distributed environment [46], where each person (e.g., a pervasive device user) in this synthetic network is represented as a node being operated by an autonomous entity. The edges of the network show the communication among entities; the thicker an edge is, the more frequent the communication will be. The desired global solution here is for the **locally-interacting** entities to collectively discover, or self-organize, their distributed communities, without involving a centralized control mechanism.

3 General Steps for Performing AOC

Having described several basic concepts in the bottom-up self-organized computing paradigm, in what follows we will present the general steps for formulating and solving a computational problem using an AOC system:

Step 1. Local-to-global thinking: Identify the global objective of *emergent computation*, and delimit the corresponding inputs and outputs.

Inputs are concerned with *local variables* and their quantitative or qualitative configurations $\mathcal{X}$. They are identified by analyzing the local properties of the independent variables in the problem space. Outputs refer to the desired macroscopic structural and/or behavioral patterns/characteristics that are dominant at the systems level, i.e., *global solution or performance* $\mathbf{\Phi}$.

Step 2. Modeling and mapping: Define *local autonomy* for individual entities $\mathbf{e}$ of $\mathbf{A}$.

A *model* of local autonomy is composed of internally and/or externally triggered/influenced behaviors b in $\mathcal{B}$, which are controlled according to the behavioral rules in a repository $\mathcal{R}$. Entities first make evaluations of their states and utilities using $\mathcal{F}$ and u. Having performed their selected behaviors b, entities update their states and utilities, and interact with their local environments (e.g., interacting with other entities) and global environment (e.g., performing certain external operations $\mathcal{O}$). The evaluations and interactions will be constrained by the coupling relationships of entities, $\mathcal{C}$.

Therefore, in modeling local autonomy, the following will be specified:

1. state space of an **entity** $\mathbf{e}$ in $\mathbf{A}$, $S_{\mathbf{e}} = \{s_{\mathbf{e}}\}$;
2. state space of its **local environment** $S_l = \{s_l\}$ and that of the **global environment** $S_g = \{s_g\}$ of $\mathbf{A}$;
3. a set of **behavioral rules**, $\mathcal{R}$, and a set of behaviors, $\mathcal{B}$;
4. a set of **coupling** relationships among entities and environments, $\mathcal{C}$ on $S_{\mathbf{e}} \times S_l \times S_g$;

5. a set of **evaluation** functions, $\mathcal{F}$ on $S_{\mathbf{e}} \times S_l \times S_g$;
6. local **utility** function u;
7. a set of **external operations**, $\mathcal{O}$.

Note that this step also implies to establish a **mapping** between input configurations $\mathcal{X}$ and autonomous entities in $\mathbf{A}$, i.e., $\mathcal{X} \rightarrow S_{\mathbf{e}}^{|\mathbf{A}|}$, where $|\mathbf{A}|$ denotes the number of entities as involved in computation.

Step 3. Autonomy-oriented computing: Deploy autonomy-oriented entities for manipulating the identified input variables and generating desired macroscopic patterns or a desired global solution.

Entities sense their states and environments, select their behaviors, and perform interactions, until $\mathbf{A}$ collectively achieves the desired emergent solution.

The *basic cycle* can be stated as follows:

1. initialize entities of $\mathbf{A}$, with respect to $\mathcal{X}$;
2. WHILE stopping criteria (e.g, the maximum number of cycles, or desired macroscopic patterns or a global solution) are not met,
   DO *self-organized computation* by individual entities operating, spontaneously, based on their local autonomy:
   (a) **assess** $\mathcal{F}$ of current states, $s_{\mathbf{e}}$, s_l, and s_g, based on $\mathcal{C}$, and utility u;
   (b) **apply** behavioral rules in $\mathcal{R}$;
   (c) **activate** selected behaviors b;
   (d) **aggregate** effects of $\mathbf{O} = \{\mathcal{O}\}$;
   (e) **adjust** states and utilities, $s_{\mathbf{e}}$, s_l, s_g, and u;
   (f) **adapt**, if necessary, behavioral rules and control parameters, based on the feedback of $\mathcal{F}$ and u.

Once again, as noted in Section 2.1, the entities that are referred to here are general notions; they can be *groups* or *populations* composed of entity elements.

4 Self-Organized Computability

As has been alluded to in the above descriptions, there are some *fundamental principles* as well as mechanisms underlying the local autonomy of entities in AOC.

Postulate 4.1 (Diversification and aggregation) *Short and long range exploratory actions, e.g., biased random walks, are necessary for achieving computable diversity, whereas positive feedback-based accelerated aggregations, e.g., through (i) entity coupling or influences and (ii) reproduction and inactivation, are necessary for emerging macroscopically-dominant patterns.*

Postulate 4.2 (Collective regulation) *In order to achieve desired macroscopic patterns or a desired global solution, the local autonomy of entities needs to be collectively regulated in their deliberative or reactive interactions. The collective regulation can be implicitly realized by incorporating into the evaluations of $\mathcal{F}$ and u, as well as the behavioral rules of $\mathcal{R}$, certain traits that are consistent with, and favored by, the desired global solution, over a greater temporal and/or spatial scale (e.g., in a long run).*

In this section, we will discuss the behavioral issues and implications as related to local autonomy, which can affect the outcomes of self-organized computing.

4.1 Diversification and Aggregation

In AOC, the **necessary condition** for self-organized computability lies in the diversity and emergence of computable configurations. This can be achieved by means of: (i) short and long-range stochastic or exploratory actions, and (ii) positive feedback-based accelerated aggregations.

While short and long-range *stochasticity* creates sufficiently diverse configurations for an AOC system, positive feedback mechanisms are essential in order to enable the system to quickly reach a *critical mass* or threshold level, e.g., aggregated population or effects, moving toward certain stable states or dominant patterns.

A right balance between stochasticity and aggregation is crucial in order to bring the AOC system to a *critical* state, operating at the boundaries of stability, or at the **edge of chaos** [2, 17]. At this critical state, the AOC system exhibits certain emergent complex behavior, in which the diversity, information flow, computational efficiency, and/or resource savings of the system, can be maximized.

Example 4.1 (Scale-free interactions) *Suppose that we have some autonomous entities interacting with others, in either a random rewiring or a chaotic connection mode as set by some initial conditions. The patterns of entity interactions can become* **scale-free** *[3], due to either the presence of* self-organized criticality *[2] in interaction dynamics or the mode of Levy flight-like [33] connections. In such a case, it will be easy for entities to connect to (or 'travel to') others (or 'locations'), or even the whole system, if the computational time is long enough. The exact connections in which the entities interact cannot be pre-determined or predicted, even though the chaotic traversal is deterministic.*

4.2 Interaction vs. Diversity

The relationship between local interactions and emergent patterns in AOC follows a certain inverse rule. As shown in the case of constraint satisfaction, too many interactions among *variable agents* will slow down or even *deteriorate* the performance in finding a solution [10, 11]. The *right* extent of interactions needs to be set, so that sufficient diversity and robustness in AOC can pervade.

We **conjecture** that such phenomena exist, and can be empirically observed, in the emergence and evolution of structural and functional motifs in complex systems and networks [16, 26, 27, 35, 36], including structured knowledge ontology, scientific topics and areas, knowledge ecology, Web communities, and social networks. And, the relationship between interactions and stability is similar to that in an ecosystem, where the probability of making a connection C between two species in the system is inversely related to the number of stable species communities N_s, i.e., $N_s \propto C^{-1+\epsilon}$ [34][40, pp.197].

4.3 Collective Regulation

As mentioned in Section 2, apart from reflecting local goal-directed self-autonomy, e.g., in some pre-programmed rules or utility function, the behavior of an entity will be triggered or affected by its environmental conditions. The environmental conditions, as evaluated with $\mathcal{F}$, consist of:

1. **Local influences:** Local influences come from entities or structures of entities, as well as other local conditions, such as resources. A simple example of influences from locally-coupled interacting entities would be the spin-spin interactions from the neighborhood in the Ising-like model [15, 40].

   Example 4.2 (Local repulsion) *Suppose that entities perform gradient descent movements in a potential field as created by other entities and by the goal of the whole system. The potential field can be dynamically acquired or updated, based on the interactions of other entities, as in the case of multi-agent robotic systems [25, 31]. In this case, the* local influences *of the environment including neighboring entities can be viewed as short-range influences or repulsion for entities to move away from unfavorable configurations.*

2. **Global influences:** While exploratory actions and aggregated/generative effects create structural and/or behavioral emergence, the **sufficient condition** for self-organized computability is that, as suggested in Postulate 4.2, entities implicitly incorporate certain *global influences or biases* ('influencing-but-not-dictating') into their local autonomy that are in a long run consistent with, and enhance, the ability of the AOC system in generating a desired global solution $\mathbf{\Phi}$.

   Example 4.3 (Global attraction) *In order to make the emergent patterns of self-organized computing converge toward a desired global solution, besides following the repulsive force in their potential field, the behavior of entities should also be affected by an attractive force that corresponds to the* global attraction *of* $\mathbf{\Phi}$ *for the AOC system. This global bias should be incorporated in the performance of local entities, through their behavioral rules, in order to implicitly regulate the self-organization of the AOC system, i.e., for entities to move toward the desired configurations and achieve the global solution.*

As described in [21], in a macroscopic sense, the above process can be viewed as collectively solving the problem of system-level performance *optimization*, in which the goal of the emergent computation in AOC, $\mathbf{\Phi}$, becomes the global objective function to be collectively satisfied.

4.4 Positive Feedback

In order to make the collective regulation of local autonomy more *effective and efficient* in converging toward a globally optimal solution, positive-feedback control mechanisms can play crucial roles in triggering and speeding up the global emergence. In doing so, the mechanisms will favor certain globally desired traits, and/or furthermore, amplify them. At the same time, less favorable configurations will be quickly eliminated.

Example 4.4 (Self-organized imaging) *Self-organized imaging [18, 28, 29] utilizes distributed autonomous entities to collectively detect certain desired image features. The entities that reside in an image environment are capable of performing a set of local behaviors, such as directional diffusion and directional reproduction, as shown in Figure 2 (illustrated in the middle box). In this case, positive feedback mechanisms are incorporated into the local autonomy of entities in the following forms: (i) directional adjustment as control-parameter* aggregation *in individuals, and (ii) selection and* reproduction *in successful entities. As a result, the global performance of autonomy-oriented feature extraction, especially adaptive to the static and/or dynamic locality of an image (i.e., without wasting computational resources), can readily be achieved.*

5 Some AOC-Related Models

As outlined in [30], in the bottom-up self-organized computing paradigm, one of the key tasks is to further develop and apply **nature-inspired** [30, 38] autonomy-oriented models for designing the local autonomy of entities. Models inspired by physical, biological, cognitive, and

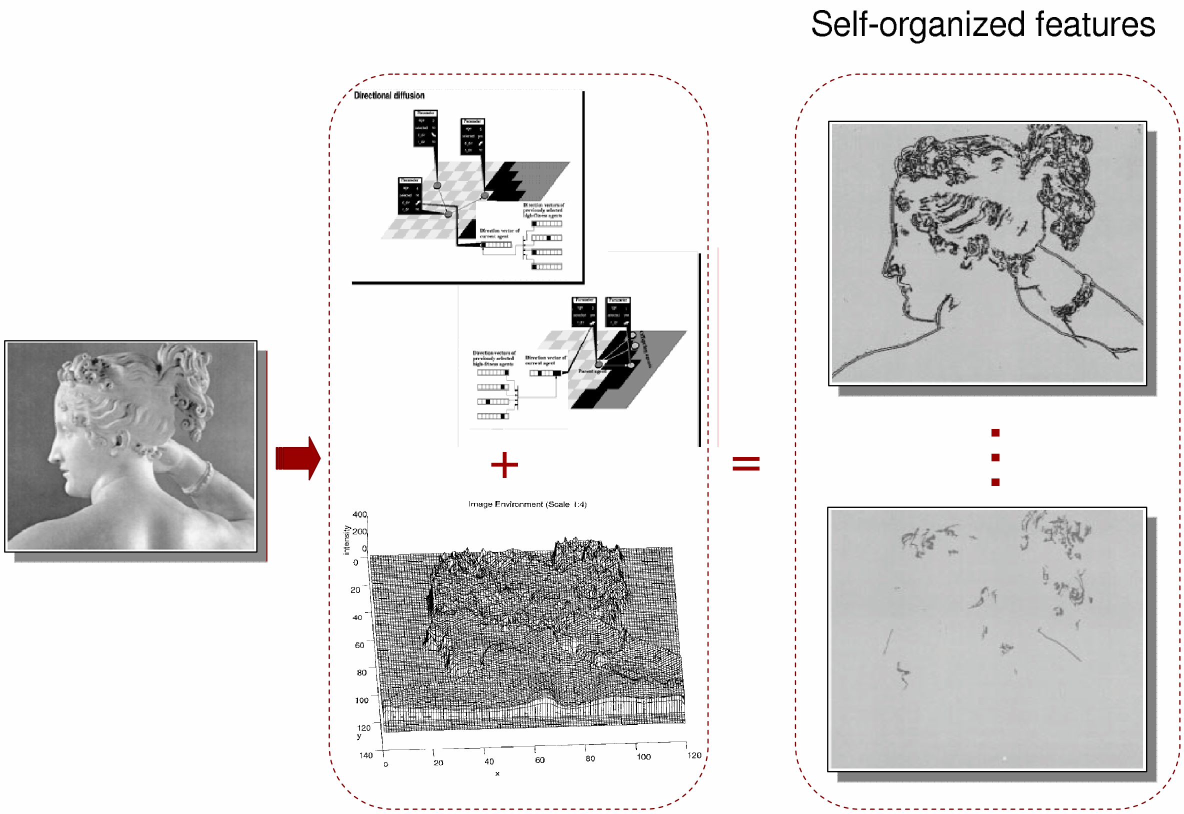

Figure 2. Self-organized imaging, where autonomous entities with directional **diffusion** and directional **reproduction** behaviors (as schematically illustrated inside the middle box) react to the conditions of their local image environments [18, 28, 29]. The desired global performance of autonomous entities in the emergent computation is to collectively extract dominant patterns or features.

social systems can be best utilized in this regard, e.g., Turing's reaction-diffusion model of morphogenesis [42], and the diffusion-reproduction model in Example 4.4 of Figure 2.

In what follows, we will revisit some of the earlier AOC-related models, and highlight the relationships of the previous efforts with our current AOC paradigm.

5.1 Swarms

Models of social swarm behavior dynamics have been proposed using individual-based [37] or population-based [7] motion/kinematic equations expressed in terms of repulsion and attraction. These models are useful in understanding the collective behavior dynamics, spatial or non-spatial, and their properties in social swarms. In doing so, details on individual interactions and environmental inputs are simplified and several assumptions in formulation are adopted (e.g., non-evolutionary, position information, and fixed strategies), and uniformly treated for all entities, for instance, as affected by repulsive and attractive forces among the entities and between the entities and their environment [7].

Besides understanding the collective behavior dynamics of social swarms, other models have been explicitly dealing with the task of optimization. In this regard, some specific implementations of AOC systems may deal with applications and/or computational issues similar to those of ant colony optimization [4, 12] and particle swarm optimization [14]. However, in AOC, we are further interested in the **general methodology**, e.g., the three approaches to AOC [21, 22], for designing and discovering robust and computationally scalable means for tackling large-scale, dynamically-evolving, and/or highly-distributed/decentralized computational problems, e.g., **in distributed problem solving and complex systems modeling as well as their interplay.** In addition, we explicitly observe **the principles and crucial roles of self-organization in computation.**

5.2 Rational Agents

AOC entities can have bounded rationality [39]. They may perform competition due to the *law of diminishing returns* or in view of internal utility/energy costs, as well as cooperation following the *law of increasing returns* as determined by the critical-mass effects, by means of various interacting behavioral policies or strategies. In a competitive AOC system, the objective is still to make sure that the whole system remains efficient in spite of the 'selfishness' of the entities. As it has been tested earlier [43], the efficiency of the system, e.g., the best utilization of given resources, can be achieved if we carefully design or automatically determine the appropriate number of strategies that an individual entity can have, and the suitable size of its memory, i.e., the amount of past performance that the entity should keep.

Example 5.1 (Cooperation vs. competition) *Entities will cooperate, e.g., form structural/functional relationships, among themselves, and therefore, amplify the performance throughput and utility gains. Due to their specific interests, goals, and/or utilities to achieve as well as their cooperative efficiency, different structures of entity organization can emerge. Since the computational environment in which the entities inhabit has limited resources, e.g., resources are not produced fast enough as compared to the rate of consumption. Different entities, or groups of entities, will start to compete in order to prevail in such a scarcity-perceived environment.*

As competition becomes more and more fierce, the gains of some teams will be diminishing. This will in turn encourage them to compromise their positions and to establish their common interests and cooperative efficiency. Once this is achieved, a new form of cooperation at the individual and/or organizational levels is established. The newly established partners will function as a new group. Forming such a new group makes the partners perform better than working competitively. On the contrary, the group may also split.

An AOC system is a multi-agent system (MAS) [44]. However, the key issues and applications of AOC differ from those of MAS. **AOC explicitly addresses the issues of self-organization, self-organized computability, interactivity, and computational scalability** in solving large-scale computationally-hard problems or modeling complex systems, whereas the latter is aimed primarily at providing a distributed AI or systems development methodology that is based on the models of rational agents. On the other hand, the local entities of an AOC system can be 'light-weight'; they may not necessarily be cognitive or rational decision-making entities. AOC also differs from conventional agent-based simulation, in that the goals of AOC are much more explicit and broader.

6 Concluding Remarks

In this paper, we have discussed some of the key concepts, principles, and implications underlying an unconventional self-organized paradigm, namely, Autonomy-Oriented Computing (AOC).

An AOC system is, generally speaking, an **open, non-equilibrium** system, in which autonomous entities react to their internal/external stimuli, or behave in a goal-directed deliberative manner, and perform interactions, e.g., information exchanges and utility updates. As a result, certain behaviors of the entities and/or their effects will be nonlinearly aggregated and amplified, as opposed to others. This process is known as *self-organization*, and serves as the *core* of AOC. For instance, in complex problem solving, AOC utilizes self-organization to efficiently converge to a desired solution state. In complex systems modeling, AOC enables the process of self-organization to effectively characterize certain empirically-observed emergent behavior and hence provide a model for its working mechanisms.

6.1 Applications of AOC

So far, the autonomy-oriented self-organized computing paradigm has been developed and tested in the following types of applications:

Type 1. Complex systems modeling: It provides the means of modeling/characterizing and hence understanding/unveiling the working mechanisms that lead to emergent behavior in complex systems. Examples of complex systems behaviors that have been studied include:

- user information-foraging behavior on the Web [9, 32],
- dynamics of social networks [50],
- emergent behavior in HIV-immune interactions [49], and
- self-organization in multi-agent systems [20].

Type 2. Complex problem solving: It offers a promising paradigm for designing and developing computationally scalable solutions (e.g., architectures, methods, and technologies) to large-scale, distributed computational problems. Examples of complex computing problems that have been handled include:

- distributed constraint satisfaction [12, 24],
- distributed optimization [41, 45],

- self-organized Web proxies [13],
- sensor network data routing [6],
- social network mining [46, 47],
- robot world modeling [31], and
- dynamic grid resource allocation [23].

It should be mentioned that results from Type 1 applications in complex systems modeling can readily provide useful models for performing Type 2 applications in complex problem solving.

We have noted several features and characteristics of the AOC paradigm:

1. It lends itself well for *natural* formulation, since many complex systems or problems at hand are locally-interacting, autonomous, and distributed/decentralized in nature;
2. It can be *light-weight* and easy-to-implement, since autonomous entities can readily be developed and deployed;
3. It is computationally *scalable* in performing emergent systems modeling and computational problem solving, since the spirit of self-organization lies in the fact that the larger the computational scale, the more effective and efficient the process should become.

6.2 Open Problems in Open Computing

The autonomy-oriented self-organized computing paradigm has presented new opportunities for future theoretical computer science research as well as practical development/deployment. Some important *local-to-global* questions can be further studied, e.g.,

1. When will an AOC system be globally critical?
2. How to define and dynamically evolve the *basin of attraction*, e.g., conditions or settings for entities, corresponding to the fixed points of AOC solutions? How sensitive are they?
3. How to effectively accelerate the emergence, especially for complex problems?

In addition, with respect to emergent computation, we want to better understand:

1. **Convergence:** What will be the right balance between global influences and diversity that can effectively generate an optimal solution?
2. **Robustness:** Will the solution be globally stable and independent of initial settings/conditions?
3. **Efficiency:** How to characterize and achieve cost-sensitive efficiency with respect to computational tasks of different complexities?

Acknowledgment

The author wishes to sincerely express his gratitude to Prof. Yiyu Yao for his time and helpful comments on a previous version of this paper. He would like to thank Dr. Yu-wang Chen for proofreading and useful feedback on the paper. This work has been supported in part by an HKBU-FRG grant (FRG/06-07/II-66).

References

[1] G. Agha. Computing in pervasive cyberspace. *Communications of the ACM*, 51(1):68–70, 2008.
[2] P. Bak. *How Nature Works*. Copernicus, 1996.
[3] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.
[4] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NY, USA, 1999.
[5] R. A. Brooks. The next 50 years. *Communications of the ACM*, 51(1):63–64, 2008.
[6] L. Gan, J. Liu, and X. Jin. Agent-based, energy efficient routing in sensor networks. In *Proceedings of AAMAS'04*, pages 472–479. ACM, 2004.
[7] V. Gazi and K. M. Passino. Stability analysis of social foraging swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(1):539–557, February 2004.
[8] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.
[9] B. A. Huberman, P. Pirolli, J. Pitkow, and R. M. Lukose. Strong regularities in World Wide Web surfing. *Science*, 280:95–97, 1998.
[10] X. Jin and J. Liu. Agent network topology and complexity. In *Proceedings of AAMAS'03*, pages 1020–1021. ACM, 2003.
[11] X. Jin and J. Liu. Efficiency of emergent constraint satisfaction in small-world and random agent networks. In *Proceedings of IAT'03*, pages 304–310. IEEE Computer Society Press, 2003.
[12] X. Jin and J. Liu. From individual based modeling to Autonomy Oriented Computation. In *Agents and Computational Autonomy*, pages 151–169. Springer, 2003.
[13] M. J. Kaiser, K. C. Tsui, and J. Liu. Self-organized autonomous Web proxies. In *Proceedings of AAMAS'02*, pages 1397–1404. ACM, 2002.
[14] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, CA, USA, 2001.
[15] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[16] J. M. Kleinberg and S. Lawrence. The structure of the Web. *Science*, 294:1849–1850, 2001.

[17] C. G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37, 1990.

[18] J. Liu. *Autonomous Agents and Multi-Agent Systems: Explorations in Learning, Self-Organization and Adaptive Computation*. World Scientific Publishing, River Edge, NJ, USA, 2001.

[19] J. Liu. Web Intelligence (WI): What makes Wisdom Web? (Invited Talk). In *Proceedings of IJCAI'03*, pages 1596–1601. Morgan Kaufmann, 2003.

[20] J. Liu and B. Hu. On emergent complex behaviour, self-organised criticality and phase transitions in multi-agent systems: Autonomy Oriented Computing (AOC) perspectives. *International Journal of Modelling, Identification and Control*, 3(1):3–16, 2008.

[21] J. Liu, X. Jin, and K. C. Tsui. *Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling (Multiagent Systems, Artificial Societies, and Simulated Organizations)*. Springer, Secaucus, NJ, USA, 2004.

[22] J. Liu, X. Jin, and K. C. Tsui. Autonomy-Oriented Computing (AOC): Formulating computational systems with autonomous components. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(6):879–902, 2005.

[23] J. Liu, X. Jin, and Y. Wang. Agent-based load balancing on homogeneous minigrids: Macroscopic modeling and characterization. *IEEE Transactions on Parallel and Distributed Systems*, 16(7):586–598, 2005.

[24] J. Liu, H. Jing, and Y. Y. Tang. Multi-agent oriented constraint satisfaction. *Artificial Intelligence*, 136(1):101–144, 2002.

[25] J. Liu and O. Khatib. Potential fields and neural networks. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA, USA, 2nd edition, 2002.

[26] J. Liu and H. Qin. Organizing synthetic agent behaviors based on a motif architecture. In *Proceedings of AGENTS'99*, pages 340–341. ACM, 1999.

[27] J. Liu and H. Qin. Behavioral self-organization in lifelike synthetic agents. *Autonomous Agents and Multi-Agent Systems*, 5(4):397–428, 2002.

[28] J. Liu and Y. Y. Tang. Adaptive image segmentation with distributed behavior-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):544–551, 1999.

[29] J. Liu, Y. Y. Tang, and Y. C. Cao. An evolutionary autonomous agents approach to image feature extraction. *IEEE Transactions on Evolutionary Computation*, 1(2):141–158, 1997.

[30] J. Liu and K. Tsui. Toward Nature-Inspired Computing. *Communications of the ACM*, 49(10):59–64, 2006.

[31] J. Liu and J. Wu. *Multiagent Robotic Systems*. CRC Press, Boca Raton, FL, USA, 2001.

[32] J. Liu, S. Zhang, and J. Yang. Characterizing Web usage regularities with information foraging agents. *IEEE Transactions on Knowledge and Data Engineering*, 16(5):566–584, 2004.

[33] B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman, New York, NY, USA, 1983.

[34] R. May. Relation between diversity and stability, in the real world. *Science*, 290(5492):714–715, October 2000.

[35] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, March 2004.

[36] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, October 2002.

[37] F. Schweitzer. *Brownian Agents and Active Particles: Collective Dynamics in the Natural and Social Sciences*. Springer, 2003.

[38] N. Shadbolt. From the Editor in Chief: Nature-inspired computing. *IEEE Intelligent Systems*, 19(1):2–3, 2004.

[39] H. A. Simon. *The Science of the Artificial*. MIT Press, Cambridge, MA, USA, 1969.

[40] R. Sole and B. Goodwin. *Signs of Life: How Complexity Pervades Biology*. Basic books (Perseus group), 2000.

[41] K. C. Tsui and J. Liu. Multiagent diffusion and distributed optimization. In *Proceedings of AAMAS'03*, pages 169–176. ACM, 2003.

[42] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London, Series B, Biological Sciences (1934-1990)*, 237(641):37–72, 1952.

[43] T. Wang, J. Liu, and X. Jin. Minority game strategies in dynamic multi-agent role assignment. *Proceedings of IAT'04*, pages 316–322, 2004.

[44] G. Weiß, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1999.

[45] X.-F. Xie and J. Liu. How Autonomy Oriented Computing (AOC) tackles a computationally hard optimization problem. In *Proceedings of AAMAS'06*, pages 646–653. ACM, 2006.

[46] B. Yang and J. Liu. An Autonomy Oriented Computing (AOC) approach to distributed network community mining. In *Proceedings of SASO'07*, pages 151–160. IEEE Computer Society Press, 2007.

[47] B. Yang and J. Liu. Discovering global network communities based on local centralities. *ACM Transactions on the Web*, 2(1):1–32, 2008.

[48] Y. Yao, N. Zhong, J. Liu, and S. Ohsuga. Web Intelligence (WI). In *Proceedings of WI'01*, pages 1–17. IEEE Computer Society Press, 2001.

[49] S. Zhang and J. Liu. A massively multi-agent system for discovering HIV-immune interaction dynamics. In *Proceedings of MMAS'04*, pages 161–173. Springer, 2004.

[50] S. Zhang and J. Liu. From local behaviors to the dynamics in an agent network. In *Proceedings of WI'06*, pages 572–580. IEEE Computer Society Press, 2006.

[51] N. Zhong, J. Liu, and Y. Yao. In search of the Wisdom Web. *IEEE Computer*, 35(11):27–31, 2002.