

Research Article

A Cooperative Optimization Algorithm Inspired by Chaos–Order Transition

Fangzhen Ge,¹ Liurong Hong,¹ Qilin Wu,² and Li Shi¹

¹*School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China*

²*College of Information Engineering, Chaohu University, Chaohu 238000, China*

Correspondence should be addressed to Fangzhen Ge; gfz203377@163.com

Received 17 May 2015; Revised 29 July 2015; Accepted 3 August 2015

Academic Editor: Matteo Gaeta

Copyright © 2015 Fangzhen Ge et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The growing complexity of optimization problems in distributed systems (DSs) has motivated computer scientists to strive for efficient approaches. This paper presents a novel cooperative algorithm inspired by chaos–order transition in a chaotic ant swarm (CAS). This work analyzes the basic dynamic characteristics of a DS in light of a networked multiagent system at microlevel and models a mapping from state set to self-organization mechanism set under the guide of system theory at macrolevel. A collaborative optimization algorithm (COA) in DS based on the chaos–order transition of CAS is then devised. To verify the validity of the proposed model and algorithm, we solve a locality-based task allocation in a networked multiagent system that uses COA. Simulations show that our algorithm is feasible and effective compared with previous task allocation approaches, thereby illustrating that our design ideas are correct.

1. Introduction

Distributed system (DS) is composed of many distributed computational components/units that mutually communicate and cooperate in networks. Owing to developments in computer networks and communication and distributed programming technology, DSs have become extensive. Ad hoc networks [1], peer-to-peer systems [2], and pervasive computing systems [3] are various types of DS. In general, DSs have a few common features as follows.

(1) *Decentralized Control.* DSs have a large scale and dynamical structure, so entire systems do not have a global control node. The actions of nodes are random. An example is the routing protocol for resource-constraint opportunistic networks [4], in which the intermittent connected nodes transfer messages from one node to another based on dynamic topology.

(2) *Sharing of Local Resources.* In this paper, resources refer to the buffer of nodes in opportunistic networks, replicas of popular web objects, computational capacity of nodes in wireless sensor networks, and so forth. All these resources are

distributed among a group of nodes and are shared by these nodes [5].

(3) *Openness, Dynamics, and Unreliability of DS.* Most DSs are open with a dynamic structure, so the link of nodes may be unreliable [6]. Moreover, due to the autonomy of nodes, each node always communicates with its neighboring nodes to maximize the total utilities [7].

(4) *Constraints of DS.* In a DS, each task will be executed on available nodes. However, in a real network, the resources of each node are limited, such as energy, channel, and bandwidth. Sometimes a variety of conditions exist in a DS, numerous relationships are intertwined (including the relationship between nonlinear dynamics and unknown mechanisms), and “relationships” are difficult to decompose into a combination of a number of simple relationships.

Recently, DSs received considerable attention because of their large number of applications, such as task and resource allocation in grids, performance optimization in multi-agent-based e-commerce systems, data fusion in distributed sensor networks, and logistics in enterprise resource planning. However, the application requirements of DSs suffer from

the aforementioned four common features. Traditional optimization methods often fail in terms of rational allocation of resources and optimal performance of DS. For instance, Asynchronous Distributed Constraint Optimization [8] and Optimal Asynchronous Partial Overlay [9] require a global topological structure to ensure optimal performance. However, acquiring global topology in DS is difficult or even impossible.

Cooperative optimization approaches of DSs have attracted the attention of many scholars. Existing methods can be divided into two categories. The first category is designing a distributed algorithm. The objective function of a DS is the sum of the node's functions. The function value of each node is acquired by exchanging with its neighbors. Bertsekas and Tsitsiklis [10, 11] put forward a parallel and distributed computing theory frame for a set of processor scheduling problems. Nedić and Ozdaglar [12] propose an average subgradient consensus algorithm to solve the convex optimization problem of a DS with constraints and further present an extended subgradient algorithm in light of the constraints on a set of nodes. Srivastava and Nedić [13, 14] devise a distributed optimization algorithm with asynchronous communication. Zhu and Martinez [15] use the primal dual subgradient algorithm to solve the optimization problem with constraints, which are described by a set of global equality and inequality. In summary, these studies are closely related with the networked utility maximization optimization [16, 17]. The second category is heuristic algorithms based on a specific behavior. Consensus-based decentralized auction algorithm (CBAA) [18] is a distributed task allocation algorithm based on auction and consistency, where each iteration runs the auction and consensus, and the algorithm uniformly converges to the optimal solution. Market-based algorithm (MBA) [19] is a distributed task allocation algorithm based on the market mechanism, where each agent can buy a task, trade transactions among them, and minimize the purchase price. Swarm-GAP (SGAP) [20] is an approximate algorithm for distributed task allocation based on the principle of division of labor in social insects, where an individual selects a task based on the stimuli of the task and their preference to the task.

However, none of the aforementioned studies consider the autonomy of the individual, especially self-organization, self-adaptation, and other advanced functions in DSs. Consequently, these algorithms cannot embody the advanced features of DSs. The current paper aims to employ the advanced functions and construct a distributed cooperative optimization algorithm (COA).

Importantly, the thinking of chaos–order transition in the foraging behavior of ants [21] gives us a vital spark of imagination for solving cooperative optimization problems. The entire foraging process of chaotic ants is controlled by three successive strategies, namely, hunting, homing, and path building. During this process, the behavior of ants transforms from chaos to order under the influence of the ants' intelligence and experience.

This paper investigates the dynamic characteristics of an agent with autonomy at the microlevel and devises a cooperative optimization model for DSs at the macrolevel.

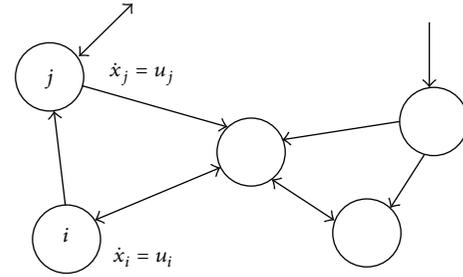


FIGURE 1: Networked multiagent system.

On the basis of our studies on chaotic ant swarm (CAS) [22–24], we propose a COA. This paper focuses on various perspectives as follows.

(1) The autonomy of agents in DSs plays a key role in representing the functions of DSs, where autonomy means that agents have spontaneous, initiative behaviors.

(2) A mechanism is shown to bridge the microlevel and macrolevel.

Specifically, our main contributions are summarized as follows.

(1) Our collaborative optimization model can be applied to describe the dynamic evolution of self-organization in DS.

(2) Our algorithm is parallel because the individuals in the proposed algorithm evolve dynamically and are parallel with little scale relations.

(3) Our algorithm fully embodies the autonomy of individual and interactive systems.

The remainder of the paper is organized as follows. Section 2 analyzes the basic dynamic characteristics of an autonomous agent in a multiagent network from the point of view of dynamics and explores the collaborative optimization model of DS. Section 3 describes how a CAS can be used to derive an effective COA. Section 4 depicts the application to a locality-based task allocation. Finally, Section 5 concludes our paper and provides further research content.

2. Cooperative Optimization Model

To construct our cooperative optimization model, we first analyze the dynamic characteristics at the micro scope and then devise the mapping method at the macro scope.

2.1. Dynamic Characteristics of an Autonomous Agent. Many researchers use the networked multiagent system based on local information structures to study DSs and have achieved good results [25, 26]. We also apply the networked multiagent system to analyze the dynamic characteristics of DSs and to set up the general dynamic characteristics of an agent.

The networked multiagent system can be expressed by a graph, where the vertex of the graph denotes an agent, while the edge represents the relationship between two agents. A networked multiagent system is shown in Figure 1, where agent j is under the influence of agent i .

From a multiagent system perspective, DSs can be regarded as a large-scale multiagent system. A single agent

of this system lacks a global unit, which adjusts its state according to its own information. A large number of agents can communicate with one another, so the dynamic evolution of agents tends to emerge globally with consistent behaviors [27].

We now consider the dynamic characteristics of an agent with autonomy in DS. In a networked multiagent system, a dynamic characteristic of autonomous agents will be a set of partial differential equations or differential equations related to the various nonlinear constraint conditions on time and space. From the perspective of relationship characteristics, the dynamic expression of a single autonomous agent is determined jointly by its current states, environmental factors, and control information. Therefore, the dynamic characteristics of an agent are written as

$$(Q_o, H_o) = F_t^k(S_t, Q_i, C_i), \quad (1)$$

where t is the current time; S_t is the current state of autonomous agent k ; Q_o is the output of autonomous agent k ; H_o is the output entropy flow of autonomous agent k ; Q_i is the input information of autonomous agent k , including normal control information Q_N and regulation information Q_A on autonomous agent k ; C_i is the input environmental information that comes from the surrounding of autonomous agent k , including the normal input C_N where autonomous agent k is imposed on by the environment and input C_A , which autonomous agent k is aware of; and F is a nonlinear function.

Equation (1) shows that the dynamic characteristics of an autonomous agent are under influence of the interaction among individuals and the macro functions of the DS. Furthermore, the structure of a single autonomous agent is described in light of the characteristics of a multiliving agent [28]. Thus, the dynamic characteristics of autonomous agent k can be expressed as

$$\text{Agent}_{d,t}^k(S_t, Q_i, C_i, \delta, \sigma, Q_o, H_o), \quad (2)$$

where d denotes the number of dimensions, S_t is a state set of agent k at time t , $Q_i = Q_i \cdot (Q_N \times Q_A)$ is the input set of agent k , Q_N is the normal control set, Q_A is the regulation of agent k by its neighbors, Q_o is the output set of agent k , $C_i = C_i \cdot (C_N \times C_A)$ is the environmental input set of agent k , C_N is the normal environmental input set of agent k , C_A is the environmental information that agent k is aware of, and H_o is the output entropy of agent k from the perspective of information theory. δ is an operator, which can generate an input sequence and is expressed as

$$\delta : S_t \times Q_i \times C_i \longrightarrow S_{t+1}. \quad (3)$$

σ is an operator, which can generate an output sequence and can be written as

$$\sigma : S_t \times Q_i \times C_i \longrightarrow Q_o. \quad (4)$$

Now we can describe an autonomous agent in Figure 2.

The seven-tuple $\text{Agent}_{d,t}^k(S_t, Q_i, C_i, \delta, \sigma, Q_o, H_o)$ is a general structure, which depicts an autonomous agent. For example, the output flow $Q_o = \sigma(S_t \times Q_i \times C_i)$, where

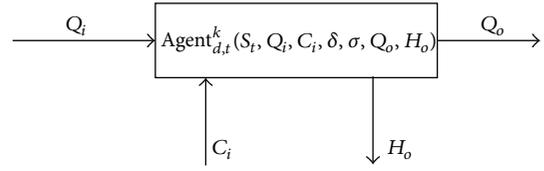


FIGURE 2: Dynamic characteristics of an autonomous agent.

$S_t, Q_{i,t}, C_{i,t}$ are the state information, input information, and environmental information of autonomous agent k at time t , respectively. When $Q_{i,t} = Q_{i,t,N} + Q_{i,t,A}$, $Q_{i,t,A}$ is the regulation imposed by agent k 's neighbors at time t and $Q_{i,t,N}$ is the normal input flow. If $Q_{i,t,A}$ is bigger than $Q_{i,t,N}$, then agent k is dominated by the regulation; otherwise, it is governed by the normal input. Accordingly, from the individual point of view, the autonomy of an agent can spontaneously adjust the state and output information with respect to the input and control information. With their continuous interaction, the agents gradually cooperate and ultimately achieve consistency.

The seven-tuple highly summarizes the idea that each autonomous agent in a DS has owned complex relationships. The tuple also reflects the microscopic characteristics of autonomous agents.

To depict the various relations in DS, constructing certain mappings and operations according to the specific problems is necessary. We can extract the main relationship or conclude similar relationships. As a result, establishing these reasonable and correct relationships is key to solving cooperative optimization problems in a DS.

2.2. Description of a Cooperative Optimization Model.

Explaining the characterization of autonomous agents at the macro scope requires an analysis on the self-organization mechanism of DS and the provision of an abstract cooperative optimization model. A mapping from the state set of autonomous agents to self-organization mechanism of DS is shown in Figure 3. The state set is composed of a number of interacting relationships among autonomous agents, an interaction that can be represented by operator \oplus . For instance, a self-organization mechanism of an open dissipative system is a coherent effect of important interacted characteristics. A self-organization mechanism set consists of various interacted elements, and the interaction can be represented by operator \odot . The mapping model from the state set to the self-organization mechanism set can be described as follows.

First, the elements of the state set run operator \oplus . Second, the results of the last step will be mapped to the self-organization mechanism set with operator θ . Finally, the self-organization mechanism set carries out operator \odot , and the various characteristics of the self-organization mechanism are formed. Operator θ is a core of this model because the self-organization mechanism is established under the control of constraints. Figure 3 shows the mapping model from the state set to the self-organization mechanism set to more clearly demonstrate the mechanism.

Let the state set and operator \oplus at time t be $(s_1, s_2, \dots, s_n, \oplus)$, where s_1, s_2, \dots, s_n are the elements of the

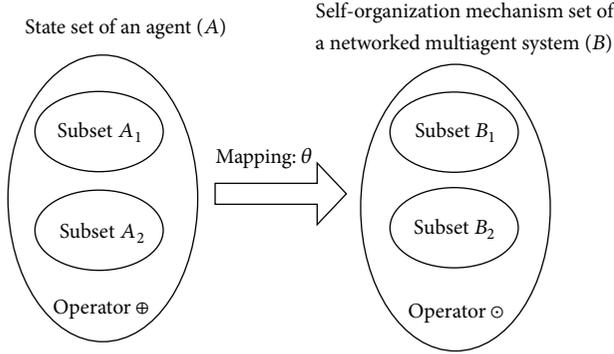


FIGURE 3: Mapping model from the state sets to the self-organization mechanism sets.

state set of an autonomous agent. Let the self-organization mechanism set and operator \odot at time t be $(s'_1, s'_2, \dots, s'_n, \odot)$, where s'_1, s'_2, \dots, s'_n are the elements of the self-organization mechanism set. Consequently, the internal or external operators of these two sets can be expressed as follows.

Internal operator of the state set \oplus :

$$s_m \oplus s_n = s_{mn}. \quad (5)$$

External operator of the self-organization mechanism set θ :

$$s_{mn} \xrightarrow{\theta_{mn}} s'_{xy}, \quad (6)$$

where s' is the mapping of s in the self-organization mechanism set.

Internal operator of the self-organization mechanism set \odot :

$$s'_x \odot s'_y = s'_{xy}. \quad (7)$$

As a consequence of (5), (6), and (7), the mapping model reflects the macroscopic process of the self-organization mechanism of DS. In other words, cooperative optimization is the mapping from state set to self-organization mechanism set under the influence of operators \oplus , \odot , and θ . Through this mapping, each part of a DS interdepends and interacts. Accordingly, the whole system evolves spontaneously, forming ordered structures and generating overall synergy.

In the succeeding sections, we devise a COA combined with the novel CAS according to the above model.

3. Description of COA

Based on the above dynamic characteristics of autonomous agents and the cooperative optimization model, a COA combined with CAS is proposed.

3.1. Chaotic Ant Swarm. Chaotic ant swarm (CAS) [29] is a global optimization algorithm developed in light of two ant behaviors, namely, chaos of a single ant and self-organization of the whole colony. CAS is constructed according to chaotic

dynamics and swarm intelligence. It has been successfully applied to solve real world problems [23, 30, 31], such as parameter identification [30], clustering [31], traveling salesman problem [23], and power optimization problem [32]. Li et al. [21] further research the foraging behavior of ants and point out that efficient foraging is accomplished under the guidance of their nest, physical abilities, and experience. They demonstrate that their viewpoint of the transition from chaos to order results from CAS. In other words, the core idea of CAS is the transition from chaos to order, that is, the self-organization of ant colony sets from the pheromones, home, physical abilities, and experience of ants. Simply put, CAS ascribes the four factors to pheromones to minimize its complexity. A brief description of CAS is provided in the succeeding section.

During foraging, an individual ant shows low-dimensional determinate chaos, whereas the colony displays a periodic macroscopic behavior. Assuming that the search space of CAS is in accordance with the domain of a specific problem, the search space S is defined as R^D , which is D dimensional continuous space, and each point s in search space S is a validated solution. Let the position of ant i be $s_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, $i = 1, 2, \dots, N$. Foraging is the process of search optimum; that is, minimize $f : S \rightarrow R$. Thus, the mathematical model of CAS algorithm can be formulated as

$$y_i(t) = y_i(t-1)^{(1+r_i)}, \quad (8)$$

$$x_{id}(t) = \left[x_{id}(t-1) + \frac{7.5}{\psi_d} v_{id} \right] \cdot e^{[1-e^{-ay_i(t)}]\{3-\psi_d[x_{id}(t-1)+(7.5/\psi_d)v_{id}]\}} - \frac{7.5}{\psi_d} v_{id} \quad (9)$$

$$+ e^{-2ay_i(t)+b} [p_{id}(t-1) - x_{id}(t-1)],$$

where t is the current step; $t-1$ is the last step; $y_i(t)$ is the organizational variables of ant i at the current step, which shows the effect of pheromone, $y_i(0) = 0.999$; $x_{id}(t)$ is the d th dimensional position of ant i , $d = 1, 2, \dots, D$, $x_{id}(0) = 7.5/\psi_d \times (1 - v_{id}) \times \text{rand}(1)$; $p_{id}(t-1)$ is the best position of ant i and its neighbors at $t-1$ step; a is a sufficiently large positive number (here $a = 200$); b is a constant, $0 \leq b \leq 2/3$; r_i is an organizational factor of ant i , which can affect the convergence of CAS; and v_{id} ($0 < v_{id} < 1$) is the d th control parameter of ant i , which can adjust the search space in accordance with the concrete problem. For a more detailed description of parameters, see [29].

To more clearly understand CAS, we explain how to construct CAS. Equation (8) shows how to generate the organizational variables y_i , which are used to achieve gradual change from chaos to order. Chaotic system $x(t+1) = x(t)e^{[3-\psi x(t)]}$ is the prototype of (9), while this chaotic system is transformed by $x(t+1) = x(t)e^{\mu[1-x(t)]}$ when $\mu = 3$ [29, 30]. The term $[p_{id}(t-1) - x_{id}(t-1)]$ of (9) is designed according to optimal adaptive control, where $p_{id}(t-1)$ is the d th dimensional best position of ant i , which is calculated by virtue of a given strategy.

Consequently, the CAS algorithm describes the search process of an ant colony. The organizational variables $y(t)$ are

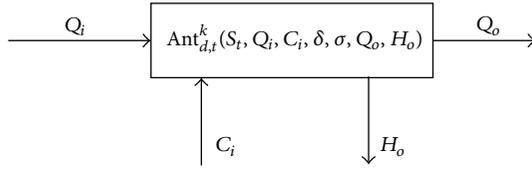


FIGURE 4: Dynamic characteristics of a single ant.

in control of the chaotic motion of ants. The control ability of $y(t)$ is weaker at the beginning of the search process but becomes stronger with time evolution. Finally, the chaos is lost and formulates an order state under the influence of $y(t)$ and $p(t-1)$. At this point, a clear understanding of how CAS depicts the transition from chaos to order can be obtained.

Chiang et al. [16] do not give a definition of an ant's neighbor. In general, the neighbors of ant i are the ants that are closer to ant i with limited distance. The authors use the Euclidean distance as a criterion. Assuming that two ants i, j exist, their respective positions are $(x_{i1}, x_{i2}, \dots, x_{iD})$ and $(x_{j1}, x_{j2}, \dots, x_{jD})$ and their distance is

$$\sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{iD} - x_{jD})^2}, \quad (10)$$

where $i \neq j, i, j = 1, 2, \dots, N$.

3.2. Cooperative Mechanism of CAS. In this subsection, we analyze the self-organization mechanism of ants and the dynamic characteristics of the individual ant in CAS algorithm to illustrate that CAS algorithm can solve cooperative optimization problems of DS.

First, the dynamic characteristics of ants are analyzed in combination with the dynamic characteristics of an autonomous agent. If an agent is treated as an autonomous agent, then the d th dynamic characteristic of ant k can be represented as in Figure 4.

We then analyze the corresponding relationships between the symbols of CAS and those of the dynamic characteristics of ants. S_t corresponds to the feasible region of ant i at time t ; Q_i concludes Q_N and Q_A ; Q_N corresponds to the position $x_{k,d}(t)$ of ant k at time t ; and Q_A corresponds to the best position $p_{k,d}(t-1)$ of ant k at time $t-1$, which can readjust $x_{k,d}(t)$. C_i corresponds to the organizational variable $y_k(t)$ of ant k at time t . CAS does not consider environmental information C_A that ants perceive. Q_o corresponds to the position $x_{k,d}(t+1)$ of ant k at time $t+1$, while H_o corresponds to the entropy change within time Δt of ant k . δ corresponds to chaos under the influence of pheromone. σ corresponds to the selection strategy for the best neighbor. Accordingly, the dynamic characteristics of ant k are expressed as

$$\begin{aligned} \delta : (S_t \times x_{k,d}(t) \times p_{k,d}(t) \times y_k(t)) \\ \longrightarrow (S_{t+1} \times x_{k,d}(t+1) \times H_{o,d}), \end{aligned} \quad (11)$$

$$\sigma : (x_{k,d}(t) \times p_{k,d}(t)) \longrightarrow p_{k,d}(t+1). \quad (12)$$

Compared to the above seven-tuple with the CAS algorithm, we can see that (11) corresponds to (8) and (12)

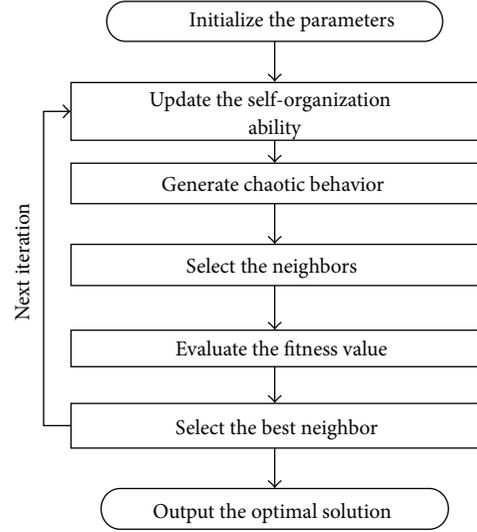


FIGURE 5: Flow chart of CAS.

corresponds to (9) and the selection strategy of an ant for the best neighbor.

Second, the self-organization mechanism of CAS is analyzed based on the cooperative optimization model. Figure 5 shows that CAS is an iterative process of optimization with three operators, namely, updating the self-organization ability, generating chaotic behavior of a single ant, and selecting the best neighbor.

We can now derive the correspondences between the cooperative optimization model and CAS. The state set corresponds to the feasible region of CAS. The internal operator \oplus can be seen as the chaotic behavior of ants, that is, the chaotic system $x(t+1) = x(t)e^{[3-\psi x(t)]}$. The self-organization mechanism set corresponds to the position set of ants after the chaotic behavior. The internal operator \odot is deemed to be the selection strategy for the best neighbor. Operator θ , which maps from the state set to the self-organization mechanism set, can be considered as continuously transforming a chaotic position to the appropriate location under the influence of a pheromone. Note that the calculations in (9) include the two operators \oplus and θ .

The above discussion shows that a chaotic ant colony is an instance of a DS. As an effective algorithm, CAS can solve the cooperative optimization problems in DSs.

However, an issue will arise if the cooperative optimization problem is tackled directly; that is, (8) of CAS is established in a hypothetical case wherein the density of a pheromone gradually increases. Although this hypothesis is reasonable in CAS and can reflect the chaos-order law, it can neither embody the mutual influence of the individual ant nor show the autonomy of an ant through the perception of local environment information. The next subsection provides an approach to calculate the self-organization ability of ants based on data field theory of cognitive physics; that is, CAS is converted into COA.

3.3. *Framework of COA.* Assume that two close ants exist in the domain, ants i and k located in (x_i, y_i) and (x, y) , respectively. The potential energy of ant i with respect to that of ant k is

$$\varphi_{xy}(x_i, y_i) = \Omega_{x_i, y_i} e^{-[(x-x_i)^2 + (y-y_i)^2]/\omega^2}, \quad (13)$$

where Ω_{x_i, y_i} is the perceived effect of ant i , which comes from ant k , corresponding to factor C_A of the seven-tuple and showing the strength of agent k that the external environment information imposes on. ω is a regulator, which is proportional to the influential radius r of ant i ; that is, $\omega = \sqrt{2}r/3$. The neighboring ants of ant i can then be defined as

$$\Gamma_{xy} = \left\{ \text{Ant}_{x_i, y_i} \mid 0 < \sqrt{(x-x_i)^2 + (y-y_i)^2} \leq r \right\}. \quad (14)$$

Accordingly, the imposed potential energy of ant i by other ants within its neighborhood can be expressed as

$$\begin{aligned} \varphi_{xy} &= \sum_{\text{Ant}_i \in \Gamma_{xy}} \varphi_{xy}(x_i, y_i) \\ &= \sum_{\text{Ant}_i \in \Gamma_{xy}} \Omega_{x_i, y_i} e^{-[(x-x_i)^2 + (y-y_i)^2]/\omega^2}. \end{aligned} \quad (15)$$

Consequently, the probability distribution of ant i 's potential is

$$P_i = \frac{\varphi_{xy}}{\sum_{\forall \text{Ant} \in \Gamma_{xy}} \varphi_{xy}}, \quad (16)$$

where $\forall \text{Ant}$ indicates any ant in the neighborhood of ant i .

According to the definition of Shannon entropy, the order degree of ant i at time t is defined as

$$H(t) = -\sum_{i=1}^N P_i(t) \ln P_i(t). \quad (17)$$

The smaller the entropy $H(t)$, the greater the order degree of the ant colony and the stronger the self-organization ability of the ant colony. Dynamic information theory [33] points out that only the open system generates the self-organization function when the negative entropy flow is greater than the positive entropy flow. Therefore, we identify if an ant has the self-organization function according to ant i 's entropy change with respect to its neighborhood. Assuming that m ants exist within the neighborhood of ant i , the entropy of ant i is

$$H_i(t) = -\sum_{k=1}^m P_k(t) \ln P_k(t). \quad (18)$$

As a result, the self-organization ability of ant i at time t can be replaced with (18), denoted by $y_i(t)$. Thus, our COA can be expressed as

$$y_i(t) = H_i(t), \quad (19)$$

$$\begin{aligned} x_{id}(t) &= \left[x_{id}(t-1) + \frac{7.5}{\psi_d} v_{id} \right] \\ &\cdot e^{[1-e^{-ay_i(t)}]\{3-\psi_d[x_{id}(t-1)+(7.5/\psi_d)v_{id}]\}} - \frac{7.5}{\psi_d} v_{id} \\ &+ e^{-2ay_i(t)+b} [p_{id}(t-1) - x_{id}(t-1)]. \end{aligned} \quad (20)$$

Note that twofold differences exist between COA and CAS. First, (19) not only reflects the mutual relationships between an ant and its neighbor, but also quantizes the self-organization ability of ants. Importantly, the sensing information of an autonomous agent is added to (13) as a feedback term, which embodies the autonomy of an agent. Second, (20) shows how to generate the state set by chaotic behavior and how to map to the self-organization mechanism under the control of the self-organization ability.

3.4. *Process of COA and Its Complexity.* Based on the above analysis, a cooperative optimization is given, whose process is listed as follows.

Step 1. Initialize the population size N , parameters ψ_d, a, b, r, i , iteration t , and ant i 's initial position x_i randomly in domain, and then calculate their fitness value $f(x)$.

Step 2. Construct the neighborhood Γ_i of ant i , and then calculate the self-organization variant $H_i(t)$ and the position $x_i(t)$ of ant i .

Step 3. If $H_i(t) - H_i(t-1) > 0$, then jump to Step 2.

Step 4. Calculate the current fitness value $f(x_i)$ of each ant, and then get the best position $p_i(t)$, which guides the moving direction of ant i for the next iteration.

Step 5. Consider $t = t + 1$; when t is satisfied, then exit the process; otherwise jump to Step 2.

From the COA process, the complexity of this algorithm is mainly calculated from the organizational variables $y_i(t)$ of ant i . The complexity of self-organizational variables is $O(n_{\text{local}}^2)$, where n_{local} is the number of ant i 's neighbors. Therefore, the complexity of COA is $O(t \cdot N \cdot n_{\text{local}}^2)$, where t is the number of iterations and N is the population size. Clearly, the ants will gradually reach the optimal or suboptimal position under the influence of self-organization. When $n_{\text{local}} \rightarrow N$, the maximum complexity of the proposed algorithm is $O(t \cdot N^3)$.

4. Simulation Experiments

To verify the effectiveness of COA, a locality-based task allocation in networked multiagent systems is solved using COA.

We first explain the locality-based task allocation in detail. When a task is executed on a few agents, the location of these agents is important because of the communication delay and energy consumption. These two factors are known to be proportional to the distance of two agents. Consequently, selecting an appropriate agent is crucial for task allocation according to the location of agents. For example, data transmission between the two nodes in wireless sensor networks needs to consider the position of the related nodes, because the greater the distance between two nodes, the larger the energy consumed. The effectiveness of task allocation is also an important evaluation of an allocation scheme, such as the distribution of network load balance. Therefore, the effectiveness of task allocation can be treated as effective feedback for an allocation scheme to reflect agent autonomy. Task allocation in networked multiagent systems can be thus divided into three steps: location awareness, task allocation strategy, and effectiveness of task allocation.

To fairly evaluate the performance of COA, it is compared with other distributed algorithms of task allocation, CBAA [18], MBA [19], and SGAP [20]. Without loss of generality, the simulation environment is set as follows. A networked multiagent system consists of 50 agents, with each agent deployed in a 40×40 plane area. Each agent can perceive the changes in its neighbors' positions. Here, the task is transferring large amounts of data from the source agent to the destination agent. A five-task set is T_1, T_2, \dots, T_5 , and these tasks increase gradually from 1 MB to 5 MB. Each task may be divided into a few parallel subtasks. COA runs 50 times. The number of iterations is 200, the population size of COA is 20, and the neighborhood radius is 10. The other parameters are set according to CAS. The parameters of comparative algorithms are set in accordance with the related references. Those comparative algorithms also run 50 times. All these algorithms terminate with the same runtimes.

4.1. Location-Aware Method of an Agent. Location is important for two agents that need to communicate with each other because of communication cost. Therefore, the centrality of two related agents is regarded as a measurement method for location awareness. Assuming that a group of agents, A_{group} , exists, d_{ij} is the distance of two agents, a_i, a_j , in the group A_{group} , and then the centrality of agent a_i is defined as follows:

$$C(a_i) = \left[\sum_{j=1}^{|A_{\text{group}}|} d(a_i, a_j) \right]^{-1}. \quad (21)$$

4.2. Quantitative Effectiveness of Task Allocation. In task allocation, if the amount of allocated tasks on an agent is only determined by the computing ability, then it leads to a long waiting time, which affects the efficiency of a system. Accordingly, the waiting time of a task is applied to measure the effectiveness of task allocation.

Assuming that the waiting time depends on the queue length of waiting tasks, let the task set be T , let the required resource be R , and let the allocated agent group be A . The task queue of agent a_i leads to resource r_k and waiting time L_{ik} . Given that the organizing and scheduling tasks in a DS are parallel, the total waiting time is

$$w_T = \max_{\forall r_k \in R, \forall a_i \in A} (|L_{ik}|). \quad (22)$$

For task allocation, we replace $1/w_T$ with the term Ω_{x_i, y_i} of (15) to show the effectiveness of feedback in task allocation, thereby embodying the autonomy of agents.

Based on the aforementioned, the centrality of task allocation is taken as the objective function:

$$f = \arg \max_{a_i \in A} (C_{a_i}). \quad (23)$$

4.3. Task Representation. Considering that CAS is a continuous optimization algorithm, the position of an ant is in the real place $[-\psi_d/2, \psi_d/2]$ and does not directly represent the task allocation. To represent a task by the position of the ant, the absolute value of the position of the ant is taken and a positive real number is transformed into an integer in the range $1 - n$ using a rounding up function. In other words, the integral position of ant i is $z_i \in [1, n]$, where n is the number of agents, which means that subtask t_i of task T is allocated to agent a_{z_i} .

4.4. Simulation Analysis. To evaluate the performance of COA, we design four measures of simulation results to make a comparative analysis of COA, CBAA, MBA, and SGAP. The measures include the occupancy rate of an agent, satisfaction rate of task allocation, fairness rate of task allocation, and task execution time. Furthermore, two kinds of methods are considered: COA with feedback (COAf) and COA without feedback (COAnf). "Without feedback" means that the influence strength between ants is fixed in the process of evolution, which cannot dynamically reflect the autonomy of ants. Thus, the influence strength Ω_{x_i, y_i} of ant i is set as a fixed value; here, it is set as 1.

Occupancy rate of an agent (OR) is the ratio of a number of allocated agents to that of agents in the networked multiagent system:

$$\text{OR} = \frac{|A_T|}{|A|} \times 100\%, \quad (24)$$

where $|A_T|$ is the number of allocated agents and $|A|$ is the number of agents in a networked multiagent system.

The satisfaction rate of task allocation (SR) is the average distance of any two neighboring allocated agents:

$$\text{SR} = \frac{2}{|A_T|(|A| + 1)} \sum_{(a_i, a_j \in A_T) \wedge (i \leq j)} d_{ij}, \quad (25)$$

where d_{ij} is the distance of agents a_i and a_j .

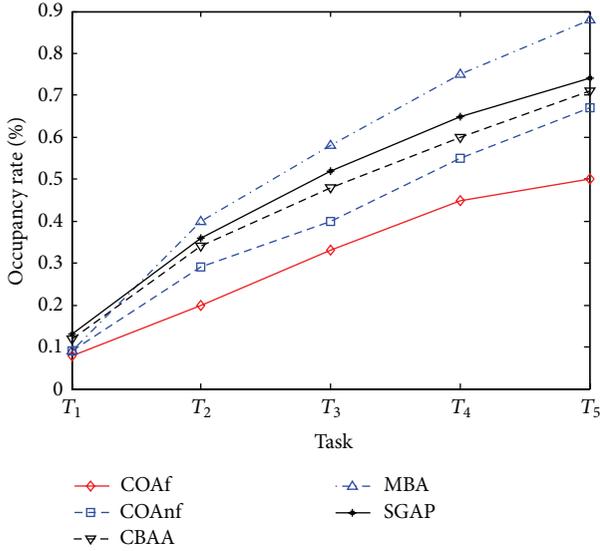


FIGURE 6: Occupancy rate for each task.

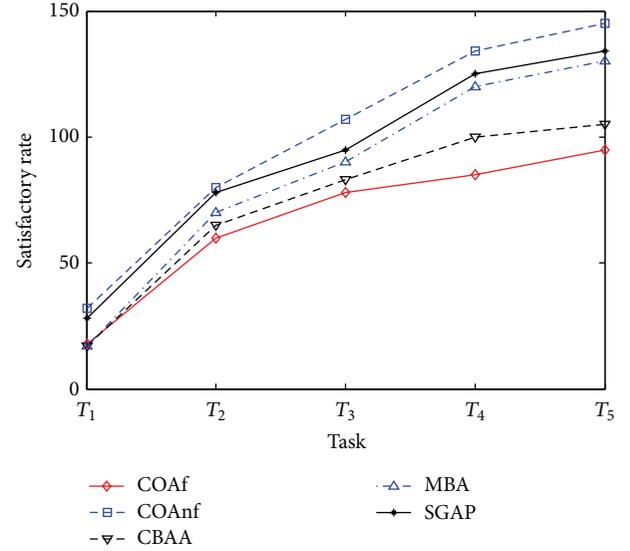


FIGURE 7: Satisfactory rate of each task.

In the center of a networked multiagent system, for a given networked multiagent system $G = \langle A, E \rangle$, the total communication cost of agent $a_i \in A$ is

$$S_i = \sum_{a_j \in A} d_{ij}. \quad (26)$$

The center of a networked multiagent system is then defined as

$$a_c = \arg \min_{a_i \in A} (S_i). \quad (27)$$

The eccentricity rate of allocated agent (FR) is

$$FR = \frac{(\sum_{a_i \in A_T} d_{ic})}{|A_T|}. \quad (28)$$

Mean execution time (AET) is the average time to obtain the optimal allocation scheme. Total execution time includes task processing time and communication time, while task processing time is specific for a fixed task.

The performance of all algorithms is then compared using the above four criteria.

Figure 6 shows that (1) the occupancy rate of each algorithm has a little difference for a small task; (2) the occupancy rate of all algorithms will increase with the increment of task but that of MBA rises fastest; (3) COAf performs better than other algorithms because it can reasonably allocate resources; and (4) the autonomy of ants leads to a better allocation scheme of resources, whereas the market behavior of MBA has great randomness.

The satisfaction rates of all algorithms are shown in Figure 7. The satisfaction rate of COAf is the best, whereas that of COAnf is the worst. These results indicate that the feedback of COAf can effectively enhance the autonomy of individual ants and the search purpose of an ant colony. As a result, the satisfaction rate is better than other algorithms.

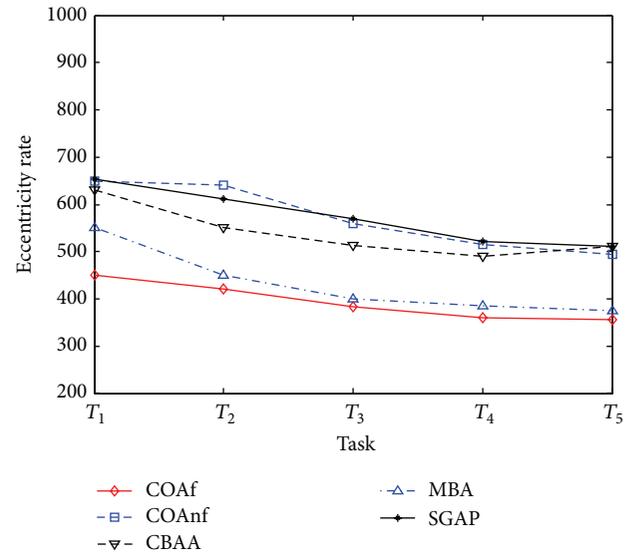


FIGURE 8: Eccentricity rate of each task.

These results also confirm that feedback is necessary for COAf to effectively adjust the input information flow.

The eccentricity rate of each task is shown in Figure 8. According to the definition of eccentricity rate, the closer the total distance between allocated agents and the central agent, the smaller the eccentricity rate and the lower the communication cost our algorithm requires (as usual, the wireless communication cost is proportional to the distance between two nodes). Figure 8 shows that the larger the task, the smaller the eccentricity rate. COAf has the lowest eccentricity compared with other algorithms when they deal with the same task. All these results show that the larger the task, the more the agents becoming involved, the even the locations of allocated agents, and the greater the decrease in communication cost.

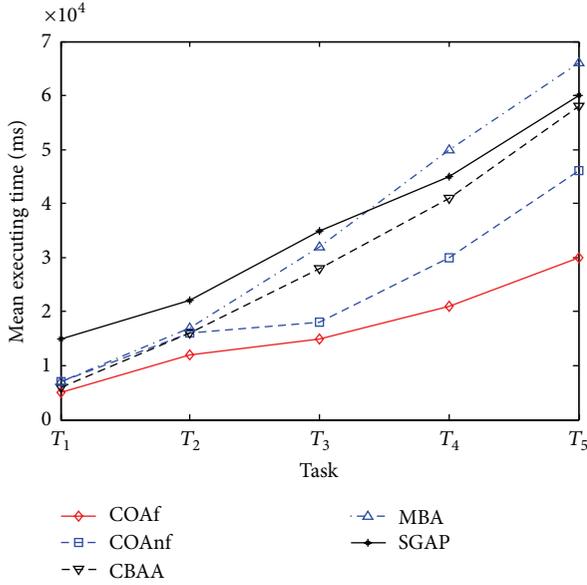


FIGURE 9: Mean executing time (ms) of each task.

The mean execution time of the five tasks on each algorithm is shown in Figure 9. The figure indicates that the mean execution time of COAf is the shortest among the algorithms and that the larger the task, the longer the mean execution time. These results prove that large tasks can be allocated evenly and that COAf performs best.

Figures 6 to 9 show that our algorithm outperforms other algorithms on the four aspects. Furthermore, the autonomy of agents in DS can improve the purpose of swarm search and can be aware of the efficient solution, thereby enhancing the efficiency of our algorithm. The communication cost is also dramatically decreased.

To indicate the evolution of COAf for task allocation, the convergence of our algorithm on task T_1, \dots, T_5 is shown in Figure 10.

Figure 10 shows that COAf can converge to a steady state for task allocation, and the larger the task, the greater the centrality of agents. As for task allocation, our algorithm can effectively allocate those subtasks to the autonomous agents because the interaction of the related agents causes them to converge to the best solution.

In summary, the above experimental results verify the effectiveness and correctness of our cooperative optimization model.

5. Conclusion

In this paper, the application requirements of DSs are used as a starting point, and a distributed collaborative optimization model is put forward in light of the transition from chaos to order. A COA is devised based on chaotic behavior and self-organization mechanism. Finally, the proposed algorithm is employed to task allocation in networked multiagent systems. This paper highlights the idea that the autonomy of agents can play a role in constructing a collaborative optimization model

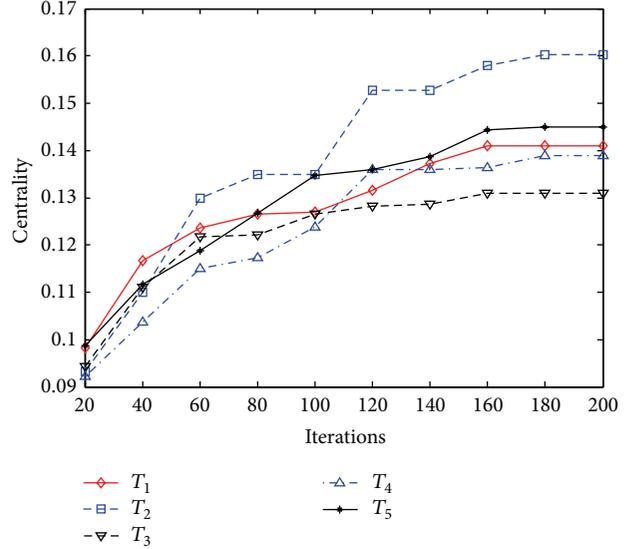


FIGURE 10: Collaborative optimization process of task allocation.

and COA. Specifically, we contribute the orderly evolution to autonomy.

Further research must be conducted to deepen the autonomy and self-organizational evolution mechanisms of dynamic DSs, to set up a general autonomous multiagent coordination mechanism, and to promote their applications in distributed systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments and feedback that helped to improve this paper. This work is supported by the National Natural Science Foundation of China (Grant no. 61070220), the Anhui Provincial Natural Science Foundation (Grants nos. 1408085MF130 and 1308085MF101), and the Education Department of Anhui Province Natural Science Foundation (Grants nos. KJ2013A229 and KJ2013Z281).

References

- [1] Y. Xue, B. Li, and K. Nahrstedt, "Optimal resource allocation in wireless ad hoc networks: a price-based approach," *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 347–364, 2006.
- [2] W. Rao, L. Chen, A. W.-C. Fu, and G. Wang, "Optimal resource placement in structured peer-to-peer networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 1011–1026, 2010.
- [3] P. Lukowicz, A. Pentland, and A. Ferscha, "From context awareness to socially aware computing," *IEEE Pervasive Computing*, vol. 11, no. 1, pp. 32–41, 2012.

- [4] H. Chen and W. Lou, "GAR: group aware cooperative routing protocol for resource-constraint opportunistic networks," *Computer Communications*, vol. 48, pp. 20–29, 2014.
- [5] S.-J. Kim and G. B. Giannakis, "Optimal resource allocation for MIMO ad hoc cognitive radio networks," *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 3117–3131, 2011.
- [6] Y. C. Jiang, Y. F. Zhou, and W. Y. Wang, "Task allocation for undependable multiagent systems in social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 8, pp. 1671–1681, 2013.
- [7] G. Attiya and Y. Hamam, "Task allocation for maximizing reliability of distributed systems: a simulated annealing approach," *Journal of Parallel and Distributed Computing*, vol. 66, no. 10, pp. 1259–1266, 2006.
- [8] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo, "ADOPT: asynchronous distributed constraint optimization with quality guarantees," *Artificial Intelligence*, vol. 161, no. 1-2, pp. 149–180, 2005.
- [9] R. Mailler and V. Lesser, "Solving distributed constraint optimization problems using cooperative mediation," in *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)*, pp. 438–445, New York, NY, USA, 2004.
- [10] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Cambridge, Mass, USA, 1997.
- [11] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [12] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [13] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.
- [14] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, 2011.
- [15] M. Zhu and S. Martinez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.
- [16] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: a mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, 2007.
- [17] F. P. Kelly, A. K. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.
- [18] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.
- [19] E. Kivelevitch, K. Cohen, and M. Kumar, "Market-based solution to the allocation of tasks to agents," *Procedia Computer Science*, vol. 6, pp. 28–33, 2011.
- [20] P. R. Ferreira Jr., F. S. Boffo, and A. L. C. Bazzan, "Using swarm-GAP for distributed task allocation in complex scenarios," in *Massively Multi-Agent Technology*, vol. 5043 of *Lecture Notes in Computer Science*, pp. 107–121, Springer, Berlin, Germany, 2008.
- [21] L. Li, H. Peng, J. Kurths, Y. Yang, and H. J. Schellnhuber, "Chaos-order transition in foraging behavior of ants," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, no. 23, pp. 8392–8397, 2014.
- [22] F. Ge, Z. Wei, Y. Lu, L. Li, and Y. Yang, "Disturbance chaotic ant swarm," *International Journal of Bifurcation and Chaos*, vol. 21, no. 9, pp. 2597–2622, 2011.
- [23] Z. Wei, F. Ge, Y. Lu, L. Li, and Y. Yang, "Chaotic ant swarm for the traveling salesman problem," *Nonlinear Dynamics*, vol. 65, no. 3, pp. 271–281, 2011.
- [24] F. Ge, Z. Wei, Y. Lu, Y. Tian, and L. Li, "Decentralized coordination of autonomous swarms inspired by chaotic behavior of ants," *Nonlinear Dynamics*, vol. 70, no. 1, pp. 571–584, 2012.
- [25] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [26] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint," *IEEE Transactions on Circuits and Systems. I. Regular Papers*, vol. 57, no. 1, pp. 213–224, 2010.
- [27] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, pp. 1226–1229, 1995.
- [28] Y. Wang, R. Tao, and B. Li, "Complex information systems based on multi-living agent," *Science in China Series E: Information Sciences*, vol. 38, no. 12, pp. 2020–2037, 2008 (Chinese).
- [29] L. Li, Y. Yang, H. Peng, and X. Wang, "An optimization method inspired by 'chaotic' ant behavior," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2351–2364, 2006.
- [30] H. Peng, L. Li, Y. Yang, and F. Liu, "Parameter estimation of dynamical systems via a chaotic ant swarm," *Physical Review E*, vol. 81, no. 1, Article ID 016207, 2010.
- [31] M. Wan, L. Li, J. Xiao, Y. Yang, C. Wang, and X. Guo, "CAS based clustering algorithm for Web users," *Nonlinear Dynamics*, vol. 61, no. 3, pp. 347–361, 2010.
- [32] J. Cai, X. Ma, L. Li, Y. Yang, H. Peng, and X. Wang, "Chaotic ant swarm optimization to economic dispatch," *Electric Power Systems Research*, vol. 77, no. 10, pp. 1373–1380, 2007.
- [33] X. Xing, "Description of the dynamic information theory and dynamical systems," *Scientia Sinica: Physica, Mechanica & Astronomica*, vol. 40, no. 2, pp. 158–176, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

