A Bio-Inspired Self-learning Coevolutionary Dynamic Multiobjective Optimization Algorithm for Internet of Things Services

Zhen Yang, Yaochu Jin, *Fellow*, *IEEE*, *and* Kuangrong Hao, *Member*, *IEEE*

Abstract—**The ultimate goal of the Internet of Things (IoT) is to provide ubiquitous services. To achieve this goal, many challenges remain to be addressed. Inspired from the cooperative mechanisms between multiple systems in the human being, this paper proposes a bio-inspired self-learning coevolutionary algorithm (BSCA) for dynamic multiobjective optimization of IoT services to reduce energy consumption and service time. BSCA consists of three layers. The first layer is composed of multiple subpopulations evolving cooperatively to obtain diverse Pareto fronts. Based on the solutions obtained by the first layer, the second layer aims to further increase the diversity of solutions. The third layer refines the solutions found in the second layer by adopting an adaptive gradient refinement search strategy and a dynamic optimization method to cope with changing concurrent multiple service requests, thereby effectively improving the accuracy of solutions. Experiments on agricultural IoT services in the presence of dynamic requests under different distributions are performed based on two service-providing strategies, i.e., single service and collaborative service. The simulation results demonstrate that BSCA performs better than four existing algorithms on IoT services, in particular for high-dimensional problems.**

Index Terms—**coevolutionary optimization, dynamic multiobjective optimization, Internet of Things (IoT), self-learning, services provision**

I. Introduction

Internet of Things (IoT) is a very complex heterogeneous network, which bridges the gap between physical and the virtual worlds. The ultimate goal of all IoT applications is to provide seamless services without human intervention. IoT is considered to be the next logical revolution [1], which is able to provide extensive services in smart cities [2], [3], smart agriculture, manufacturing [4], [5], smart healthcare, and smart home. Autonomous IoT systems are of great importance but many challenges remain to be addressed [6].

In general, IoT services can be categorized into four classes [7], [8], i.e., identity-related services, information aggregation services, collaborative-aware services and ubiquitous services. This article focuses on optimization of ubiquitous services, which is very challenging, since there exist many heterogeneous and dynamic links and a large amount of uncertainties. Some efforts have been reported along this line of research. For example, the concept of opportunistic IoT services was proposed [9], while sensing as a service was suggested in [10], [11].

A large-scale IoT environment is composed of thousands of distributed entities. Once service requests are detected, multiple optimal services selected from enormous candidate sets need to be provided immediately. IoT should not only be able to provide services for dynamic concurrent requests, but also to reduce energy consumption [12], [13], reduce service time, and improve information accuracy. In addition, availability of services, bandwidth allocation, and reliability should be taken into account, especially in resource constrained environments [14]. Consequently, service selection is a multiobjective optimization problem (MOP). In [15], services are assigned to interfaces with heterogeneous resources to achieve optimal solutions. However, the development of efficient algorithms for service selection remains challenging and has not been widely investigated. Hence, the purpose of this article is to implement global composition and multiobjective optimization of ubiquitous services in dynamic IoT environments.

Dynamic multiobjective optimization problems (DMOPs) are challenging due to the fact that multiple conflicting objectives that change over time must be optimized simultaneously [16], [17]. Evolutionary computation and swarm intelligence have been shown to be powerful methods to solve optimization problems in dynamic environments [18]. Among many others, coevolutionary approaches are very attractive [16], [19], [24]-[29]. Through competitive-cooperative coevolution, different subpopulations separately optimize a subset of the decision variables, where the decomposition process of the optimization problem is adaptive rather than being manually designed and fixed at the beginning of the evolutionary optimization [19]. Liu *et al.* [16] proposed a decomposition method, where the subcomponents cooperate to evolve for better solutions. A linear regression prediction

This work was supported in part by the International Collaborative Project of the Shanghai Committee of Science and Technology under Grant 16510711100, the National Natural Science Foundation of China under Grants 61473078, 61473077, 61503075 and 61603090, in part by Shanghai Science and Technology Promotion Project from Shanghai Municipal Agriculture Commission under Grant 2016-1-5-12, and in part by the Program for Changjiang Scholars from the Ministry of Education (2015-2019). (*Corresponding authors: Yaochu Jin and Kuangrong Hao*)

All the authors are with the Engineering Research Center of Digitized Textile and Apparel Technology, Ministry of Education, and also with the College of Information Science and Technology, Donghua University, Shanghai 201620, China (E-mail: yaochu.jin@surrey.ac.uk; krhao@dhu.edu.cn).

Yaochu Jin is also with the Department of Computer Science, University of Surrey, Guildford, GU27XH, United Kingdom, and the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. Zhen Yang is also with the College of Information Engineering, Huzhou University, Huzhou 313000, China.

strategy was used to produce rapid responses to new changes in the environment. Similarly, prediction strategies for dynamic multiobjective optimization have also been proposed [20]-[23]. Furthermore, Mestari *et al.* [24] developed a neural network architecture and a new processing method for solving nonlinear equality constrained MOPs. Shang *et al.* [25] adopted the immune clonal selection to solve DMOPs. To enhance the uniformity and the diversity of the solutions, they also employed coevolutionary competitive and cooperative operations. Ding *et al.* [26], [27] proposed an immune co-evolutionary algorithm for MOPs with specific application scenarios. In [28], diversified multiobjective cooperative evolution was adopted for scheduling problems, while an endocrine regulation mechanism was applied in [29]. Jiang and Yang [30] proposed a steady-state and generational evolutionary algorithm (SGEA) for handling MOPs with time-varying characteristics. If a change is detected, SGEA reuses a portion of previous solutions to quickly adapt to changing environments. The work reported in [31] integrated transfer learning approaches into an evolutionary algorithm. Finally, collaborative decomposition [32], neighborhood concept and local search were applied [33]-[35].

Ideally, algorithms for solving dynamic optimization problems (DOPs) should be able to self-learn like human beings to better deal with the changing environment. In [36], an artificial neural network was employed and the online learning capability of the neural network controller has been shown to be able to help the control system respond quickly to changes in system. In [37], a novel iterative adaptive dynamic programming based infinite horizon self-learning optimal control algorithm was developed for nonaffine discrete-time nonlinear systems, which uses a neural network. A new avenue was opened up in [38] by integrating adaptive neural dynamic programming into self-learning control for continuous-time uncertain nonlinear systems. A self-learning mechanism of immune systems was employed in [39] to design reconfigurable controllers. A hybrid learning clonal selection algorithm was proposed by incorporating Baldwinian learning and orthogonal learning to guide search [40]. In [41], [42], active learning was shown to be able to reduce the computation time in solving MOPs, while in [43], active learning and information theory were combined for dual control. Liu *et al*. [44] proposed a new swarm intelligence algorithm inspired from social learning to solve QoS-aware cloud service composition problems. Barkoczi and Galesic [45] investigated how different social learning strategies affect the balance of exploration and exploitation, and the resulting performance was very encouraging.

All aforementioned research focuses on DMOPs with a fixed dimension of the decision space. This work, by contrast, aims to solve DMOPs with a changing number of decision variables, where the number of requests sent at any given moment is constantly changing. These situations are often seen in both single and collaborative IoT services. Note that in single services, one requested task is completed by one service provider. In collaborative services, on the contrary, one requested task can be collaboratively completed by a composition of service providers.

To solve DMOPs of IoT services, we develop new cooperative mechanisms inspired from the nervous, endocrine, and immune (NEI) systems of human beings [46]-[50]. The main contributions of this paper are as follows. (1) Inspired by the NEI systems in human beings, a three-layer progressive architecture for DMOPs is suggested. In the proposed algorithm, layer 1 aims to quickly approximate the location of the Pareto optimal solutions by co-evolving a number of subpopulations. Then, layer 2 focuses on increasing the diversity of solutions. Finally, layer 3 refines the solutions to further enhance their accuracy. (2) A social learning mechanism inspired from the human brain [48], [49] is introduced into the second layer, which enables the algorithm to self-learn, thereby increasing the solution diversity. Meanwhile, individual hormones are used to adjust the step sizes in the learning. (3) By making full use of the information about the locations of the service requests and optimal solutions from previous environments, knowledge-based adaptive local search is designed to cope with dynamic requests and improve tracking accuracy.

The rest of this paper is organized as follows. Section II briefly presents the preliminaries related to this work. Section III formulates the optimization of IoT services as a dynamic multiobjective optimization problem. Section IV presents the proposed bio-inspired self-learning coevolutionary algorithm (BSCA) and provides a detailed description of its search mechanisms and strategies. Section V compares the effectiveness of the proposed algorithm with four existent ones in a dynamic agricultural IoT environment. Finally, conclusions are drawn in Section VI.

II. PRELIMINARIES

A. Dynamic Multiobjective Optimization

Existing DMOPs can be classified into four types, as shown in Table I [51], where *n* is the number of decision variables, $Ð_{X}$ is the decision space, *m* is the number of objectives, and R^{m} denotes the objective space. Existing research has been limited to addressing Type I DMOPs, where the numbers of decision variables and objectives remain constant and only the objective or constraint functions change over time.

TABLE I. TYPES OF DMOPS

Decision space	Objective space (*m*, R^{m})	
(*n*, $Ð_{X}$)	No change	Change
No change	Type I	Type III
Change	Type II	Type IV

B. Dynamic IoT Service

Since the number of requests sent at any given moment is constantly changing, the number of optimal service providers also changes. In optimization of IoT services, the number of service providers determines the dimension of the decision space. Consequently, the optimization problem belongs to Type II. However, it can also happen that the positions of requests and services to be provided change even if the number of requests remains the same. Thus, IoT optimization problems may also belong to Type I. Several typical distributions of the locations of the requests are considered to rigorously validate the proposed algorithm. The test instances are grouped into five categories, in which requests come from one congestion area, two adjacent areas, two opposite areas, three adjacent areas or the entire area, respectively. These five distribution categories are illustrated in Appendix A of the Supplementary material.

III. Dynamic Multiobjective Optimization Model of IoT Services

A. IoT Service Model

IoT service systems typically consist of three layers, a perception layer, a network layer and an application layer. The application layer is dedicated to providing services for people or things. An IoT system contains a large number of heterogeneous devices and networks, and the data streams from the perception layer are regarded as service requests. The ultimate goal of IoT systems is to autonomously provide services by these devices, resulting integrated cyber-physical systems [52].

A service request can be represented as a five-tuple (*request_id*, *type*, *workload*, *priority*, *location*), where *request_id* indicates the unique identification code of the service request. *type* indicates the type of the request, which can be very different due to the heterogeneity of the devices in the IoT system. *workload* and *priority* depend on the collected data from all kinds of sensors. *location* represents the geographical coordinates of the request.

A service provider accepts the response of service requests, which is expressed as a six-tuple (*service_id*, *type*, *u*, *e*, *δ*, *location*), where *service_id* indicates the unique identification code of the service provider. *type* denotes the type of service. *u* indicates the usage status of the service. *e* represents the unit energy consumption. *δ* denotes the ability value, which is the amount of work done per unit time. *location* denotes the geographical coordinates of the service.

In this work, priorities are divided into two levels, urgency and general. Ideally, if the amount of data collected by the sensors is larger than a predefined threshold, a request is considered to be urgent, otherwise it is general. We also assume that the data has been processed and prioritized, and all requests are within the scope of the service facilities.

B. Dynamic Multiobjective Optimization Model

There are many indicators and objectives that can be considered in IoT services [14]. This work focuses on the minimization of the energy consumption and service time. These objectives can be achieved by optimizing the configuration of the limited resources and the selection of the service providers. Consequently, optimization of IoT services is formulated as a bi-objective optimization problem.

Consider a set of service providers denoted by $S(t) = \{1, \ldots, N\}$. The service requests are derived from a set $R(t) = \{1, \ldots, M\}$. Then the two objectives are expressed as follows.

$$\min f_1 = \sum_{i=1}^{N(t)} \sum_{j=1}^{M(t)} x_{ij}(dist(s_i^t, r_j^t) + ec(s_i^t, r_j^t))) \quad (1)$$

$$\min f_2 = \max_{i=1}^{N(t)} \{ST_i\} \quad (2)$$

s.t.

$$dist(s_i^t, r_j^t) = \sqrt{(X_i^t - X_j^t)^2 + (Y_i^t - Y_j^t)^2} \quad (3)$$

$$ec(s_i^t, r_j^t) = \begin{cases} workload_j^t \cdot p_i \cdot e_i & ; x_{ij} = 1 \\ 0 & ; x_{ij} = 0 \end{cases} \quad (4)$$

$$ST_i = workload_i / \delta_i \quad (5)$$

$$CoSerNum_j^t = \lceil \eta \cdot workload_j^t \cdot priority_j^t \rceil \quad (6)$$

$$x_{ij} \in \{0, 1\}, \forall i \in S, j \in R \quad (7)$$

$$\sum_{j=1}^{M_{t_0}} x_{ij} = 1 \quad ; i \in S \quad (8)$$

$$\sum_{j=1}^{M(t)} CoSerNum_j^t \le N_{t_0} \quad (9)$$

Assume that the layout of the initial IoT has N_{t_0} service providers and a maximum of M_{t_0} concurrent requests. *N*(*t*) and *M*(*t*) are two time-varying functions, which will be explained later on. In (1), objective f_1 is to minimize the total service energy consumption. Specifically, *dist*(s_i^t,r_j^t) indicates the transmission cost between service s_i^t and request r_j^t. As the dimension and locations of the requests change, the dimension of the decision space changes as well, where each dimension s_i of the decision vector changes over *t*. Therefore, the coordinates (*X*, *Y*) of service s_i and request r_j in (3) vary with *t*. In (4), *ec*(s_i^t,r_j^t) represents the energy consumption between service s_i and request r_j. Similarly, for the same s_i, *ec*(s_i^t,r_j^t) may be different for different requests. $workload_j$ is the workload of request *j*. p_i represents the share of the workload assigned to be completed by s_i. e_i is the unit energy consumption of s_i.

In (2), objective f_2 is to minimize the service time, which is the maximum completion time of all tasks performed by the service providers. The completion time ST_i is given in (5), where $workload_i$ indicates the amount of work done by the service *i*, and δ_i is the ability value of service *i*.

Considering the priority and the amount of the task, several service providers are required to serve a request collaboratively. The exact number of service providers depends on the product of three values, i.e., availability of services, *workload* and *priority* of r_j, as shown in (6), where *η* represents the availability ratio of services at time *t*. We assume that the requested task at time (*t*-1) has been completed before a request at time (*t*) arrives in the system. No breakdown of service providers is considered in this work.

Constraints are given in (7), (8) and (9). In (7), x_{ij} denotes whether service *i* is assigned to request *j*. If yes, the value is 1; otherwise, the value is 0. Constraint (8) describes that a service provider is only assigned to one request in a moment. Constraint (9) restricts the number of all the required cooperative services for multiple requests at time *t*, which should be less than the total number N_{t_0} of available services. Besides, *M*(*t*) and *N*(*t*) are defined in (10) and (11).

$$M(t) = \lceil rand \cdot M_{t_0} \rceil \quad (10)$$

where $M(t)$ represents the number of concurrent requests at time t.

$$N(t) = jfs_i jx_{ij} = 1; i 2 S; j 2 R_t g. \qquad (11)$$

where $N(t)$ represents the current number of service providers serving the concurrent requests at time t.

IV. Bio-Inspired Self-Learning Coevolutionary Algorithm

In this section, we introduce the proposed BSCA for dynamic multiobjective optimization of IoT services. To autonomously provide services in a changing environment, we treat IoT service system as a living system such as human beings. In a living organism, three major systems, namely, nervous, endocrine and immune (NEI) systems, cooperate with each other to achieve the overall goal of balancing and stabilizing the whole system. NEI systems can be seen as complex network control structures, which regulate the functions of various organs by means of cytokines, hormone, and chemical transmitter [53]. Inspired from the working mechanisms of human NEI systems, this work develops an integrated model of BSCA consisting of three layers, as shown in Fig. 1. The corresponding roles of cytokines and neurotransmitters in BSCA are also denoted in parentheses. In the following, we describe in detail the functions of each layer.

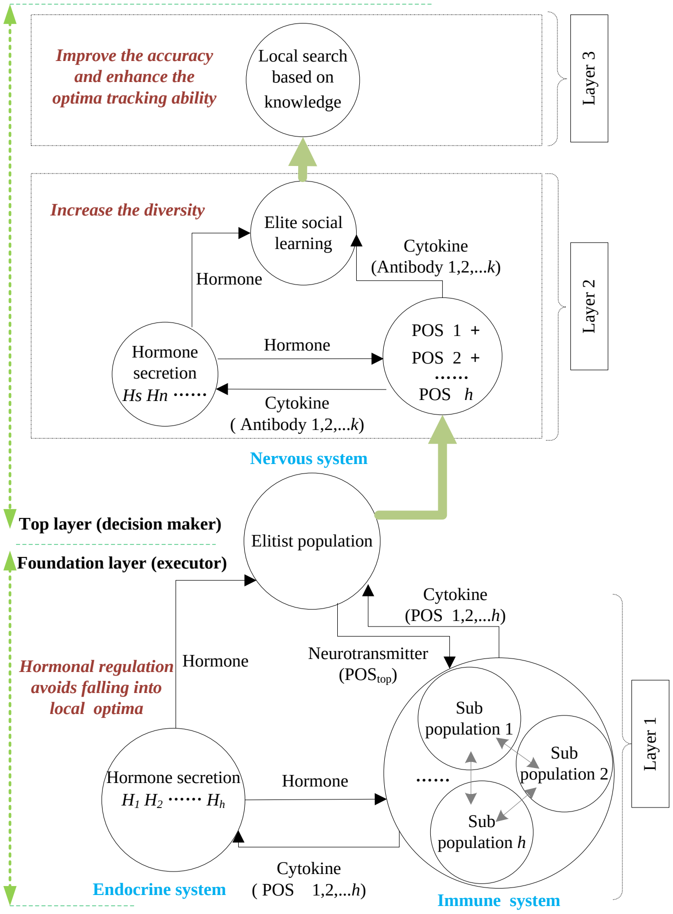

Improve the accuracy and enhance the optima tracking ability
Local search based on knowledge
Layer 3
Increase the diversity
Elite social learning
Cytokine (Antibody 1,2,...k)
Hormone
Layer 2
Hormone secretion Hs Hn ……
Hormone
POS 1 + POS 2 + …… POS h
Cytokine (Antibody 1,2,...k)
Nervous system
Top layer (decision maker)
Elitist population
Foundation layer (executor)
Cytokine (POS 1,2,...h)
Hormonal regulation avoids falling into local optima
Hormone
Neurotransmitter (POS_top)
Sub population 1
Sub population 2
……
Sub population h
Layer 1
Hormone secretion H_1 H_2 …… H_h
Hormone
Cytokine (POS 1,2,...h)
Endocrine system
Immune system

Fig. 1. The architecture of BSCA inspired by human NEI systems.
NS: nervous system; IS: immune system; ES: endocrine system.

In BSCA, layer 1 is meant to find as many candidate solutions as possible. Layer 1 acts as the executor, called foundation layer, which mainly responds to requests and finds optimal services. This is similar to the function of the immune system, which fights against foreign substances (equivalent to requests) and produces antibodies (equivalent to service providers) that can match and destroy the foreign substances. The Pareto optimal set (POS) 1, 2,…, h obtained by each subpopulation by a non-dominated immune algorithm (NNIA) [54] are transmitted as cytokines to NS and ES. At this time, ES secretes different hormones H_1, H_2,…, H_h according to the characteristics of each set of solutions, thereby affecting the interaction among subgroups and avoiding getting stuck in local optima. Then the elitist population in layer 2 performs a sequence of behaviors and makes decisions. After that, NS feeds back the POS of the elitist population ($\mathrm{POS}_{\mathrm{top}}$) as neurotransmitters to IS. In this way, the stimulus of $\mathrm{POS}_{\mathrm{top}}$ on the immune cells can enhance the function of IS. In other words, $\mathrm{POS}_{\mathrm{top}}$ is helpful for subpopulations in exploring a larger solution space and accelerating the convergence.

The function of layer 2 aims to increase the diversity of solutions. Layer 2 and layer 3 together constitute the top layer, which is equivalent to human NS and acts as the decision maker. These two layers perform the search starting from the solutions found by layer 1. Layer 2 chooses only good individuals (POS 1, 2,…, h) as the initial solutions; layer 3 performs refining operations based on the results of layer 2. Thus, the three layers form a progressive hierarchy.

Once POS 1, 2, …, h are passed to layer 2, they are cloned according to hormone concentration and individual differences. Then they become independent elite individuals, which are transmitted as cytokines to NS. Inspired from social learning mechanisms observed in the human brain, BSCA enhances population diversity through promoting information exchange and avoiding collisions among individuals. Next, when antibodies 1, 2,…, k are transmitted as cytokines to ES, ES secretes different hormones H_s and H_n according to the characteristics of each antibody, which are used to dynamically regulate the learning step size. The details will be presented in Section IV (C).

Finally, layer 3 intends to improve the accuracy of solutions. Layer 3 attempts to capture the structure information of the problem to refine the solutions of layer 2. Given time-varying concurrent multiple requests, the ability to track the dynamic optimum and improve the accuracy must rely on a robust and effective local search strategy. Based on knowledge, gradient zones of service providers are established to guide and enhance local search, which ensures the evolution towards better direction and assists with future searches for the global optimum.

The implementation details of BSCA are described below.

A. *Synergistic Mechanisms of Endocrine Regulation*

Layer 1 and layer 2 use synergistic mechanisms of endocrine regulation to assist the evolution of the solutions. Here, the hormone of layer 1 aims to regulate the migration of non-dominated solutions between subpopulations, while the hormone of layer 2 is used to adjust the learning step size.

1) *Hormonal regulation of layer 1*

The subpopulations 1, 2,…, *h* in layer 1 emulate the multiple antibody groups in the human immune system, which evolve separately. In NEI systems, the immune function is activated or inhibited by the hormone receptors on immune cells when the secreted hormone reaches a certain concentration level. In the model, we use the migration frequency to control the frequency of interaction among the three systems, which aims to prevent the subpopulations from getting stuck in local optima. For example, for the two subpopulations Ab_j and Ab_k with the largest and smallest hormone concentration, POS_j migrates to Ab_k and POS_k migrates to Ab_j. Migrating the non-dominated solutions helps the subpopulation escape from local optima and expand their search space.

The hormone can be calculated as follows.

$$H(Ab_j) = \alpha \frac{l_j}{\frac{1}{h}\sum_{j=1}^{h} l_j} + \beta \sum_{b=1}^{m} \frac{\frac{1}{h}\sum_{j=1}^{h} f_b^{min}(Ab_j)}{f_b^{min}(Ab_j)} + \gamma \frac{\overline{NS}(Ab_j)}{\frac{1}{L}\sum_{i=1}^{L} NS(a_i)};(j=1,2,...,h) \tag{12}$$

where $H(Ab_j)$ is a composite index that reflects the ratio of the number of non-dominated solutions (*NDS*), the values of objective functions and the diversity of *NDS*. α, β, γ are the corresponding weight coefficients. In the following, we elaborate the three terms in (12).

l_j in the first term denotes the size of *NDS* in subpopulation Ab_j, as notated in (13). *h* is the total number of subpopulations. This item shows the ratio of l_j in the collection of entire *NDSs*.

$$l_j = |NDS(Ab_j)| \tag{13}$$

The second item of (12) $f_b^{min}(Ab_j)$ denotes the minimum of the *b*-th objective function in subpopulation Ab_j. *m* is the total number of the objectives. The numerator of the second item is the average of $f_b^{min}(Ab_j)$ of all the subpopulations.

In the third item of (12), diversity of *NDS* is evaluated by non-similarity. The definition in (14) is used to calculate the non-similarity rate between the two antibodies, i.e., a_i and a_k. a_i^c and a_k^c represent *c*-th dimension of a_i and a_k, respectively. *r* is the dimension of the decision space. *L* is the size of the group consisting of *NDSs* from all the subpopulations.

$$NSRate(a_i;a_k) = \frac{\sum_{c=1}^{r} Bool(a_i^c \neq a_k^c)}{r};(i \neq k;i,k=1,2,...,L) \tag{14}$$

Then, let $NS(a_i)$ be the non-similarity of a_i, which is the average of non-similarity rates to all other antibodies. $\overline{NS}(Ab_j)$ denotes the average non-similarity of *NDS* in Ab_j. This term measures the level of diversity of Ab_j in the whole group.

2) *Hormonal regulation of layer 2*

All POSs in layer 1 become independent elitist individuals in layer 2 after cloning. Like in NEI systems, they are transmitted as cytokines to ES. At this time ES secretes different hormones for each individual. The secreted hormones H_f are synthesized by two kinds of endocrine cells, i.e., H_s and H_n. The former describes the performance of individual itself and the latter reflects the level in the small groups formed by the neighborhood relationship.

For individual *i*, the corresponding hormone H_f is expressed as follows, which is also called the hormone factor.

$$H_s^i = \arctan \frac{\max\{d_j | j \in B\} - d_i}{\max\{d_j | j \in B\} - \overline{d}} \tag{15}$$

$$B = \{1,2,...,L\} \tag{16}$$

$$H_n^i = \arctan \left| \frac{d_{i-1} + d_{i+1}}{2} - d_i \right| \tag{17}$$

$$H_f^i = H_s^i \cdot H_n^i \tag{18}$$

where d_i represents the crowding distance of individual *i* in [54]. A set B is defined as the elitist population. $\overline{d}$ stands for the average crowding distance of all these elite individuals. In (17), the individuals in front of and behind d_i are denoted as d_{i-1} and d_{i+1}, respectively. $d_i > \overline{d}$ means that the solution density in the neighborhood of individual *i* is lower than the average density. The adjustment of the original position of *i* should be minor, i.e., the value of H_s^i should be slightly adjusted. On the contrary, $d_i < \overline{d}$, means the solution density of individual *i* is higher than the average, and therefore, H_s^i should be more dramatically adjusted. When d_i is close to the center of the crowding distances between the two individuals (*i*-1, *i*+1), the distribution is fairly uniform. In other words, the local characteristic of individual *i* is good, thus H_n^i should only be slightly adjusted. Otherwise, H_n^i should be more significantly adjusted and H_f^i should increase. The full use of the crowding distance, which is incorporated into the hormone factor, can effectively help improve the optimization performance.

B. *Clone Operator*

The mothers of the clones are POSs of all the subpopulations. For subpopulation Ab_j, the number of solutions in its POS_j is denoted by l_j. The formula of the clone operator is as follows.

$$q_i = \left\lceil \frac{H(Ab_j)}{index(a_i)} \right\rceil ;i=1,2,...,l_j \tag{19}$$

where $H(Ab_j)$ is hormone concentration of subpopulation Ab_j, $index(a_i)$ indicates the index number of antibody a_i in POS_j. When $index(a_i)$ is equal to 1, the cloning multiple is the largest, whose value decreases as the index increases. This kind of method greatly increases the opportunity of antibodies at or near the extreme points to achieve affinity maturation. In the odd generations, $index(a_i)$ is obtained by sorting f_1; but in the even generation, $index(a_i)$ is obtained by sorting f_2. By doing so, we ensure that both objectives will be evolved with an equal probability.

C. *Human Brain-based Social Learning Mechanism*

There are two motivations for using the brain-inspired social learning mechanism. Firstly, offspring of the individuals in the elitist population are generated by cloning parents, which is the process of generational inheritance of parents' information. Thus, an offspring individual cannot receive information from other parents. Social learning provides a means for individuals to learn from each other, thereby increasing the population diversity. Secondly, there are structures in a society, such as elitists and neighbors, on which social learning is typically based. In the following, we provide the details of social learning implemented in the model.

1) *Elitist and neighborhood based learning*

We regard the entire elitist population as a certain social group. Here we employ two self-learning models in the elite evolutionary learning strategy, namely, one based on the neighborhood relationship and the other on fitness.

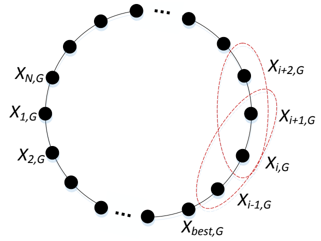


Fig. 2. The ring topology of neighborhood.

The ring topology is one structure widely used to describe the neighborhood relationship of individuals in a population [33], as shown in Fig. 2, where N is the size of the population, $X_{i,G}$ represents one individual. Here, we divide the population into groups consisting of three individuals, and one individual can be in different groups. In addition, an individual may learn from the best individual in a group. These two learning patterns can be described in (20), which are conducted in an alternate manner with a probability.

$$SL_{i,G} = \begin{cases} X_{i,G} + N(0.7,\sigma^2)\cdot(X_{i-1,G} - X_{i+1,G}) + H_f^{i,G} & ; \text{if } rand_i(0,1) < 0.3 \\ X_{best,G} + N(0.7,\sigma^2)\cdot(X_{r1,G} - X_{r2,G}) + H_f^{i,G} & ; \text{otherwise} \end{cases} \quad (20)$$

where $X_{i-1,G}$ and $X_{i+1,G}$ are neighbors of $X_{i,G}$ in the neighborhood-based learning mode. $N(0.7, \sigma^2)$ denotes a Gaussian distribution with a mean of 0.7 and standard deviation σ, which is adaptively adjusted according to the distribution of the changing requests. In the elitist learning mode, the individual with the minimum f_1 value at the G-th generation in the entire elite group is denoted as $X_{best,G}$, since we consider f_1 as the main objective in this work. X_{r1} and X_{r2} are chosen randomly from the population. In (20), $H_f^{i,G}$ is the hormone factor for individual i at the G-th generation, as defined in (18). $H_f^{i,G}$ adaptively adjusts the learning step size. The elitist learning mode helps accelerate convergence while the neighborhood based learning aims to increase diversity. Details of the learning mechanism are also described in Algorithm 1.

2) *Critical learning*

Kumaran *et al.* [50] provided evidence that the medial prefrontal cortex selectively mediates the updating of knowledge. In other words, one critically and selectively learns based on one's current knowledge level. This critical learning process can be described as follows in a high abstraction level.

$$SA_{ij,G} = \begin{cases} SL_{ij,G} & ; \text{if } rand_j(0,1) < AF_G \,||\, j == rand_j(1,n) \\ X_{ij,G} & ; \text{otherwise} \end{cases} \quad (21)$$

$$AF_G = 0.55 + \frac{1}{\pi}\cdot\arctan\left(2 - 10\frac{G}{G_{max}}\right) \quad (22)$$

where G and G_{max} are the current and maximum generations, $X_{ij,G}$ and $SA_{ij,G}$ denote the j-th decision variable of individual X_i before and after critical learning at the G-th generation, respectively. AF_G is a learning factor at the G-th generation, which gradually decreases over the generations from 0.8958 to 0.0896. In the earlier search stage, the differences between individuals are relatively large, thus AF_G is larger to encourage more explorative search. At the later search stage, the individuals become more similar and thus AF_G will be smaller, which promotes more exploitative search. In short, AF_G determines the amount of information from other individuals being passed to an individual's offspring, which has considerable impact on the convergence speed and population diversity. Critical learning is implemented from lines 11 to 13 in Algorithm 1. N is the current size of the clones, while n is the current dimension of decision variables. For the repair operation (line 18), illegal and repetitive decision variables in one individual $X_{i,G}$ are replaced by the randomly generated elements in the complementary set of all available service providers and $X_{i,G}$.

Algorithm 1: Self-learning based on NEI

```
1   Input: Clones of POSs of all subpopulations
2   Output: Offspring population
3   for $i$ =1 to $N$
4       if $rand_i < 0.3$
5           Implement the neighborhood-based learning
            mode according to (20)
6       else
7           Implement the elitist-based learning mode
            according to (20)
8       end if
9   end for
10  for $i$ =1 to $N$
11      for $j$ =1 to $n$
12          if $rand_j < AF_G || j == rand_j(1,n)$
13              $SA_{ij,G} = SL_{ij,G}$
14          else
15              $SA_{ij,G} = X_{ij,G}$
16          end if
17      end for
18      Repair solutions to ensure that each decision
        variable is within the range
19  end for
```

D. *Knowledge-based Local Search*

To track moving optima efficiently in a time-varying environment, layer 3 aims to search promising areas not found in layers 1 and 2. By making full use of the problem structure as well as optimal solutions found in the previous environments, BSCA is able to speed up convergence and improve the accuracy of the final solutions in changing environments.

When new requests arrive, a portion of the optimal solutions found for previous requests will be adopted in layer 3 to increase the speed of response to environmental changes. Recall that the dimension of the decision space in the new environment may be different from that in the previous environments. To address this issue, only portion of the solutions in the old environment corresponding to those requests that are the same as in the new environment will be adopted. In the following, we provide an illustrative example to show how to reuse a portion of the solutions in the previous environment (t-1).

Assume a fragment of requests at time (t-1) are:

…19, 6, 37, 57, 50, 17, 58, 10…

and a fragment of requests at time t are as follows:

…10, 19, 8, 60, 46, 9, 26, 57…

At first, we identify the same requests between the previous environment (t-1) and the present environment t. These are requests 19, 57, and 10. Suppose that at time (t-1), providers 21, and 25 are assigned to request 19, providers 68, 41, 50, and 62 are assigned to request 57, and providers 11, 15, 8 are assigned to request 10. These providers are included in the current solution if they are not already in the solution.

Finally, local search is performed for requests that are different from those in the previous environment. In the following, we describe in detail the refinement search conducted in layer 3.

Different service requests r_j have different sensing ranges, meaning that the intensity of requests on service providers s_i is also different. Thus, according to the distance between service requests and service providers, and unit energy consumption of providers, a connection weight sequence of (s_i, r_j) pair is designed. Assume there are N service providers and M requests, so the following connection weight matrix W_m can be built.

$$W_m = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1i} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2i} & \cdots & w_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{j1} & w_{j2} & \cdots & w_{ji} & \cdots & w_{jN} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{Mi} & \cdots & w_{MN} \end{bmatrix}$$

where each row corresponds to a request and each column corresponds to a service provider. The connection weight w_{ji} between request r_j and service s_i is defined as follows.

$$w_{ji} = \| s_i, r_j \| + ec(s_i) \cdot uw \qquad (23)$$

where || s_i, r_j || is the transmission cost determined by the distance measure between the two elements. $ec(s_i)$ denotes the energy consumption of s_i and uw is the unit workload defined to be 1.

An affinity matrix A_m is constructed according to W_m. The service providers for r_j on the j-th row in matrix W_m are sorted in an ascending order. The sorted elements for r_j will become the j-th row of A_m. The corresponding indexes for the service providers are thus obtained. The service provider in the first place of j-th row indicates the highest affinity with request r_j.

However, selecting an appropriate subset of providers to perform local search is challenging, since local search should be adequate yet computationally efficient. Therefore, the service providers are divided into three zones according to A_m before local search is carried out. The providers in the first zone are used to determine whether local search should be performed. The second is to extend the local search, hoping to explore a better feasible solution. Only when the number of service providers available for substitution in the first and second zones is less than required, providers in the third zone will be considered in local search. Since the numbers of substitutions required by single service and collaborative service are different, the zone size for the two service policies should vary. Based on our pilot studies, the zone size for single service is set to 3 and that for collaborative service is set to 5.

Due to simultaneous processing of multiple concurrent requests, an individual solution contains multidimensional service providers. A schematic diagram is shown in Fig. 3, where $CurSol_i$ denotes the current solution, $reqNum$ is the number of concurrent requests and each bit corresponds to a service provider. For example in Fig. 3, service providers 69 and 46 are for request r_1. For collaborative service, the solution to one request has more than one bit.

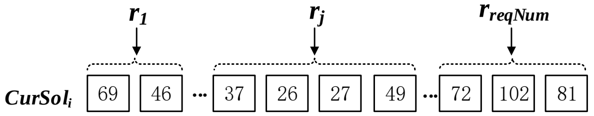


Fig. 3. Corresponding relation between concurrent requests and bits of $CurSol_i$.

Algorithm 2: Knowledge-based Local Search

```
Input: Offspring population, requests at time instances $t$ and ($t$-1), POS_top at time ($t$-1), affinity matrix $A_m$
Output: Updated offspring population
for $i$=1 to $N$
    Reuse a portion of outdated solutions for the same requests at time ($t$-1).
    Calculate $SamePro$ for the remainder of requests according to (24).
    Then calculate $AvePro_i$.
    for $j$=1 to $restReqNum$
        if $SamePro_j$<= $AvePro_i$
            $DSet$= []; $Flag$=zeros(1,$CoSerNum_j$);
            In turn, determine whether each service $s$ in the first and second zones exists in $CurSol_i$.
            If exists, determine whether $s$ is in the services composition responding to $r_j$. If yes, write down the position $pos$ and assign $Flag(pos)$ to 1.
            If not, $s$→$DSet$.
            if |$DSet$| <$CoSerNum_j$
                Add the services in the third zone but not in $CurSol_i$ to $DSet$.
            end if
            if ~$isempty(DSet)$
                for $k$=1 to $CoSerNum_j$
                    if $Flag(k)$==0&&~$isempty(DSet)$
                        $DSet(1)$→$r_j^k$ of $CurSol_i$
                        $DSet(1)$ = [];
                        if $isempty(DSet)$    break;
                        end if
                    end if
                end for
            end if
        end if
    end for
end for
```

The local search is carried out as follows. Firstly, we determine for the request r_j whether a combination of services exists in its first zone. If not, $SamePro_j$ defined in (24) is equal to 0. If yes, the number of intersection of the combination set C_j and the zone Z_j is calculated. In (24), $CoSerNum_j$ is defined in (6). The greater the $SamePro_j$, the better the local solution of r_j.

$$SamePro_j = \frac{|\{C_j \cap Z_j\}|}{CoSerNum_j}, j \in R \qquad (24)$$

Secondly, $SamePro_j$ of the j-th request is calculated according to (24). Then the probabilities are averaged over the number of requests and the result is denoted as $AvePro$. $AvePro$ conveys the overall level of matching between the combination set for each request and its first zone. The search intensity for each request depends on itself and *AvePro* of the current individual.

Details of the local search are described in Algorithm 2, where *restReqNum* represents the number of remaining requests except the same requests. *DSet* denotes the set of providers in three zones that are not used by the current solution. |*DSet*| is the size of *DSet*.

E. Overall Algorithm of BSCA

The pseudo code of overall algorithm is given in Algorithm 3. The algorithm is executed from Layer 1 to Layer 3, as shown in Fig. 1. Hormonal regulation (lines 4 and 6) is performed as described in (12) and (18), respectively. Refer to Section IV(A).

Algorithm 3: The Overall Algorithm of BSCA

```
1  Input: Initial parent subpopulations
2  Output: $POS_{top}$
3  for $G$ =1 to $G_{max}$
4      Layer 1: Evolve subpopulations to obtain the POS of each subpopulation; migrate POS according hormonal regulation
5      Clone the POS of each subpopulation
6      Layer 2: Adjust the learning step size with hormonal regulation. Self-learning (Algorithm 1)
7      Layer 3: Knowledge-based local search (Algorithm 2)
8      Evaluate solutions
9  end for
```

V. Performance Evaluation with Application to Agricultural IoT Services

In order to validate the effectiveness of BSCA, we apply it to the field of dynamic optimization of agricultural IoT services.

For an agriculture greenhouse control system based on IoT, sensors are set for monitoring the growth of various vegetables and fruits. Sensors include humidity sensors, soil moisture sensors, temperature sensors, nutrient elements sensors, and carbon dioxide sensors. The data collected by these sensors are sent to the service platform and processed there. When the processed data meet certain predefined conditions, they are regarded as service requests. In the agricultural IoT scenarios considered in this work, devices, such as irrigation machine, fertilizer applicator and pesticide spraying machine are regarded as service providers. Selecting the best matching services to deal with dynamic concurrent multiple requests to minimize the total cost and service time is the main goal to achieve. Note that the proposed algorithm will be triggered once requests are received.

A. Experimental Settings

An agricultural IoT region (80×80) is designed for experimental studies. 120 sensors (service requests) are evenly distributed in the region. In addition, the agricultural IoT devices (service providers) distribute evenly in the considered region in the form of 11 by 11, resulting 121 devices in total. In this work, the number of random requests is fixed to 30, which can be divided into five typical distributions under two service providing strategies, as shown in Fig. S1 in the Supplementary material. In case of single service, the number of decision variables equals to the number of requests, while in case of collaborative service, the dimension of decision space varies according to (6). The solution dimensions in the five cases for two service strategies are described in TABLE II. These experiments are based on the assumption that the previous batch of requests has been completed before the next batch of requests arrives. The dynamic multiobjective algorithm SGEA [30], two popular immune algorithms HEIA [55], NNIA [54], and a popular multiobjective optimization algorithm, NSGA-II [56] are compared with the proposed BSCA. SGEA, like other dynamic multiobjective algorithms, assumes that the dimension of the search space does not change during the optimization. Thus, a minor modification must be made to SGEA. Specifically, the dimension of the old solutions at the previous time instance (t-1) is adjusted to the current dimension at time t when there is a change in the search dimension. Similar to BSCA, a portion of old solutions for the same requests are utilized as a portion of the current solutions. Then the remaining decision variables of the current solutions are randomly generated. Other settings of SGEA are exactly the same as in [30].

In the experiments, it is assumed that the five request categories arrive sequentially in every 100 generations, meaning that the environment changes in 100 generations and 500 generations in total will be run for each compared algorithm. The setting of the parameter is given in TABLE III, which are specified through empirical studies. Note that BSCA uses three subpopulations for different exploration regions and the size of the elite population varies. Based on our empirical results, the average number of fitness evaluations used by BSCA in 100 generations is 9868, which is less than the compared algorithms.

TABLE II. The dimensions of Decision space

Strategy	Case 1	Case 2	Case 3	Case 4	Case 5
Single service	30	30	30	30	30
Collaborative service	63	72	57	67	61

TABLE III. Parameters setting

Parameter	NSGA-II	NNIA	HEIA	SGEA	BSCA Sub pop	 Elitist pop
Population size	100	100	100	100	25	adap-tive
Crossover rate	0.9	1	1	1	1	/
Mutation rate	0.8	1/N(t)	1/$N(t)$	1/$N(t)$	1/N(t), 1.5/N(t), 1/N(t)/1.5	/

B. Distribution Density of Service Requests

In the following, we calculate the distribution density of requests for tuning the variance of the Gaussian distribution N(0.7, ¾^2) in (20). Assume a collection of requests (30 in this work) sent out at time t, denoted as R$_t$, which can be categorized into five situations as shown in Fig. S1.

1) *Distribution density of requests*

First, we determine whether the nearest neighbors of request r_j are in R_t. Then, the distribution density of r_j is calculated as follows.

$$½_j = \frac{jfr\ 2\ N\ eiF\ ield\ ;9r\ 2\ R_t g\ j+1}{5} \qquad (25)$$

where *r* indicates one of neighboring fields. Note that the maximum number of neighbors is 4 and consequently, $½_j$ ranges between 0.2 and 1.

2) *Overall distribution density of* R_t

Next, the overall distribution density is calculated as follows.

$$½_d = \frac{1}{jR_t j}\, ¢_{j=1}^{jR_t j} ½_j \qquad (26)$$

where $½_j$ is calculated according to (25).

For the five categories of request distributions, the overall distribution density $½_d$ averaged over multiple randomly generated scenarios listed in Table IV. Finally, we can tune the standard deviation in $N(0.7, ¾^2)$ based on the overall request density. In principle, the variance should be tuned according to $½_d$, and the larger $½_d$ is, the larger $¾^2$ should be. This is because the fewer idle service resources there are in the neighborhood, the more explorative search must be performed. The best values of $¾^2$ are then obtained according to the above discussions.

TABLE IV. THE RELATIONSHIP BETWEEN $½_d$ AND $¾^2$

Strategy	Parameter	Case 1	Case 2	Case 3	Case 4	Case 5
Single service	$½_d$	0.8533	0.5333	0.4933	0.4133	0.3467
	$¾^2$	0.3	0.2	0.15	0.1	0.1
Collaborative service	$½_d$	0.8533	0.4933	0.4667	0.3867	0.3200
	$¾^2$	0.3	0.15	0.15	0.1	0.1

C. *Performance Metric*

Several performance indicators [57], [58] accounting for convergence and diversity of non-dominated solution sets have been proposed. In this work, hypervolume (HV), set coverage (C-metric) and D-metric are adopted for evaluation of the solution sets obtained by the compared algorithms. In addition, diversity measure [59] is employed to assess the population diversity.

30 independent runs are performed for each instance. The aforementioned performance metrics are described in Appendix B of Supplementary material.

In the following, the performance of BSCA is compared with SGEA, HEIA, NNIA and NSGA-II for both single and collaborative service strategies.

D. *Performance Comparisons for Single Service*

Fig. 4 shows the HV results of five algorithms for the single service strategy. Fig. 4 plots the convergence profiles of the average HV metric values over 30 independent runs. From these results, we can observe that BSCA shows the best HV among the compared algorithms on all five request categories (Case 1 to Case 5 in Fig. S1). Note that NNIA is also very competitive, as it is catching up with BSCA at the end in Case 4 and Case 5. It is also observed that HEIA overtakes NSGA-II in the latter generations in each environment in four of the five cases. This might be attributed to the cooperation of multiple evolutionary strategies in HEIA. In addition, the starting point of HV value of SGEA is the highest in Case 1, Case 3 and Case 4, indicating that SGEA is able to respond to environmental changes very quickly in these cases. This can mainly be attributed to the generational selection and the effectiveness of the guided population reinitialization. However, the performance of SGEA does not improve significantly in the later search stage for low-dimensional and discrete problems. Overall, BSCA performs the best, which benefits not only from hierarchical structures, but also from the reuse of the previous solutions throughout the search process.

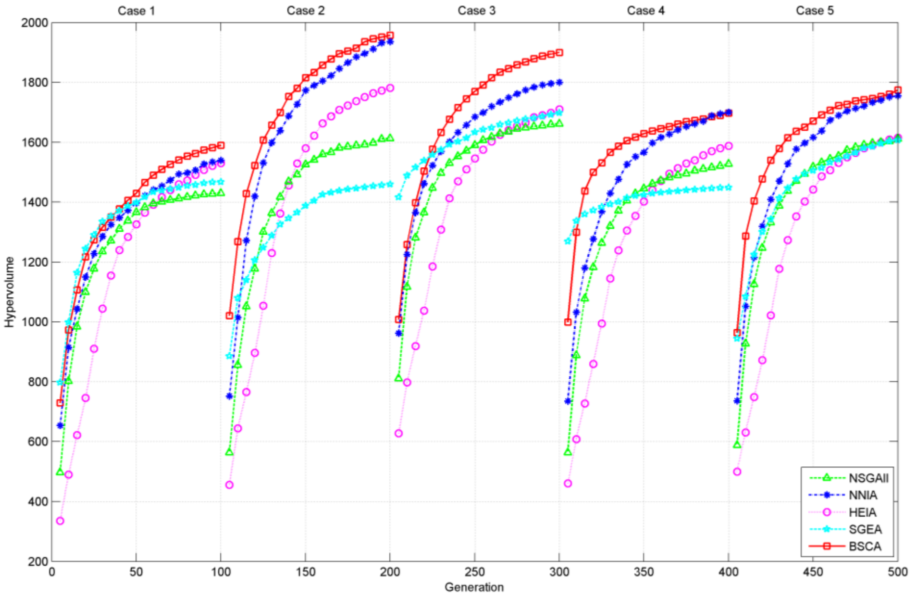


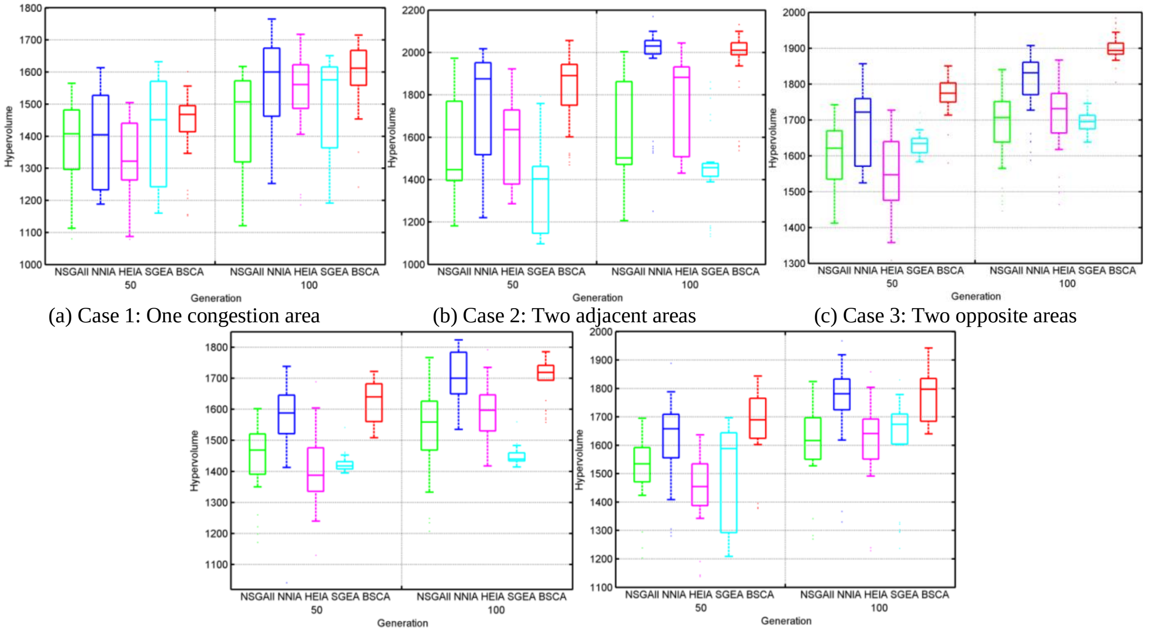
Fig. 4. Convergence profiles in terms of the average HV in five distributions cases for the single service strategy.

To further compare the performance of each algorithm in 30 independent runs, Fig. 5 presents the box plots of the HV results obtained by five algorithms in generations 50 and 100 in five distribution cases. As a whole, BSCA performs remarkably better than the other compared algorithms. The only exception is Case 2 in generation 100, where NNIA performs the best. These boxplots also indicate that BSCA performs more consistently compared to other four algorithms.

E. Performance Comparisons for Collaborative Service

Fig. 6 shows the HV results of the five algorithms for the collaborative service strategy. From these results, it is observed that BSCA clearly outperforms other four algorithms in all cases except for Case 1, where NNIA performs the best, although BSCA is going to catch up with NNIA at the end of generation. This may be due to the fact that the requests in Case 1 distribute in a very concentrated area, which reduces the benefit of explorative search. We also note that the average HV of NSGA-II, NNIA and HEIA are similar to the results for the single service strategy. NSGA-II is outperformed by HEIA during the later stage of evolution. Nevertheless, overall, SGEA performs better in collaborative service than in single service. This confirms that SGEA outperforms NSGA-II, NNIA and HEIA when handling high-dimensional problems. As to collaborative service strategy, the starting point of the HV value of BSCA is always the highest except in Case 2, indicating that BSCA has superior performance on higher dimensional optimization problems since it is able to quickly respond to environmental changes in most cases. By comparing the results in Figs. 4 and 6, it is noted that BSCA outperforms the compared algorithms more significantly in Cases 2-5 in collaborative service, implying that BSCA is more competitive for dealing with high-dimensional problems.

Fig. 7 presents the box plots of the HV results obtained by the five algorithms in generations 50 and 100 for the five distribution cases, from which we can see that BSCA performs clearly better in four of the five cases. Another dynamic multiobjective optimization algorithm, SGEA, ranks the second



(a) Case 1: One congestion area (b) Case 2: Two adjacent areas (c) Case 3: Two opposite areas

(d) Case 4: Three adjacent areas (e) Case 5: Entire area

Fig. 5. Box plots of the HV results in representative generations (50 and 100) on five distributions cases under the strategy of single service.

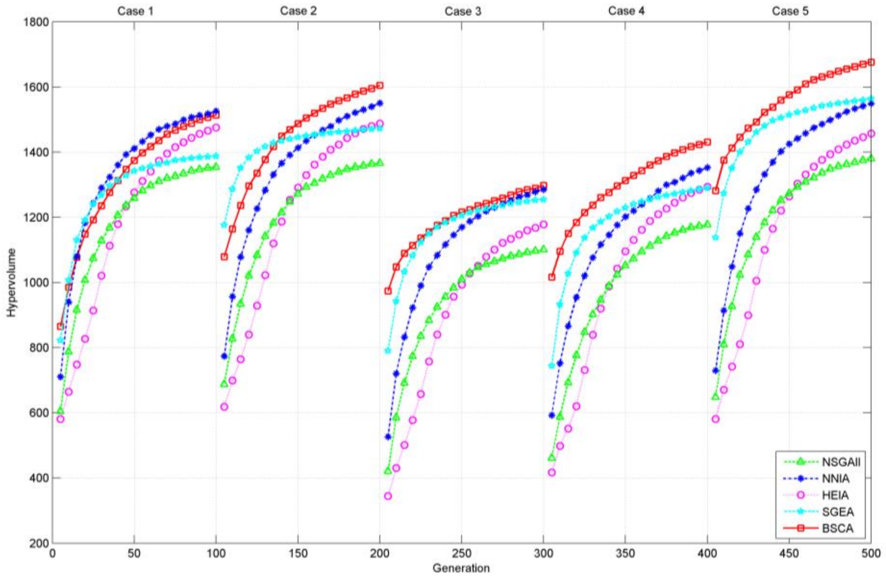
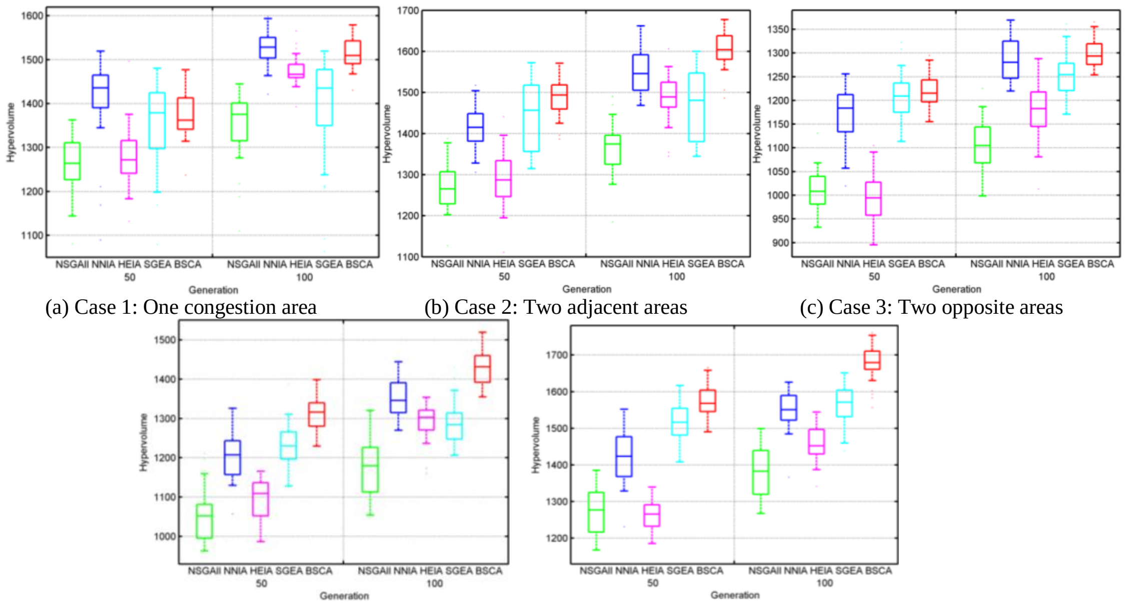


Fig. 6. Convergence profiles in terms of the average HV values in five distributions cases for the collaborative service strategy.



(a) Case 1: One congestion area (b) Case 2: Two adjacent areas (c) Case 3: Two opposite areas

(d) Case 4: Three adjacent areas (e) Case 5: Entire area

Fig. 7. Box plots of the HV results in representative generations (50 and 100) on five distributions cases under the strategy of collaborative service.

on median value in generation 50, yet in generation 100, its performance has declined and is not so prominent compared with its earlier search stage. Comparing the results in Fig. 5 and Fig. 7, we find that the performance superiority of BSCA over the compared algorithms is more significant in terms of the maximum value, median, and minimum value. This further confirms the superiority of BSCA in dealing with high-dimensional problems.

F. Comparison of Computation Time

To compare the complexity of the five compared algorithms, we list the average computation time in Table V, where ‘Ss’ means single service and ‘Cs’ denotes collaborative service. It is noted that computation time fluctuates with the number of requests received. As the dimension of decision space increases, the computation time will increase correspondingly. From the results, it is clear that NNIA is the most efficient, computationally, and NSGA-II the least. On average, BSCA takes more time than NNIA, HEIA and SGEA since it employs the multi-layered method and spends time on back and forth communication among layers.

TABLE V. AVERAGE COMPUTATION TIME (S)

Case		NSGA-II	NNIA	HEIA	SGEA	BSCA
Case 1	Ss	77.1371	**2.0969**	7.7961	13.2580	18.1971
	Cs	76.9155	**2.5591**	23.2601	15.1368	46.9831
Case 2	Ss	67.6911	**1.9032**	7.4899	13.9466	8.0299
	Cs	77.6679	**2.7740**	30.6288	15.3266	30.5047
Case 3	Ss	68.2179	**2.0832**	7.6818	14.0726	9.6710
	Cs	75.7962	**2.5811**	18.6727	15.2799	34.4382
Case 4	Ss	68.2100	**2.1993**	7.7973	14.2318	13.2992
	Cs	76.9695	**2.5226**	25.1278	15.3558	40.4372
Case 5	Ss	68.2252	**2.1292**	7.5390	14.0765	11.4517
	Cs	71.2038	**2.6201**	20.6591	15.3561	32.7730
Mean	/	72.8034	**2.3469**	15.6653	14.6041	24.5785

VI. CONCLUSION

In this paper, a bio-inspired self-learning coevolutionary algorithm (BSCA) having a three-layer progressive structure is presented for dynamic multiobjective optimization of IoT services to minimize service costs and service time. BSCA is inspired by the mechanisms found in human nervous, endocrine and immune systems to quickly track the moving Pareto optimal solutions in the presence of changing requests. The simulation results demonstrate that the proposed algorithm is competitive in dynamic optimization of agricultural IoT services. In practice, IoT service system may select one of the extreme solutions or other Pareto optimal solutions on the front according to the service strategy specified by the decision-maker.

One important topic yet to study is to handle abnormal situations in dynamic IoT environments, such as services failure. Meanwhile, it is essential to consider more practical situations where new requests may arrive before all the previous requests have been handled. Finally, it is also of interest to verify the performance of BSCA on the standard DOP benchmark problems to further identify its strengths and weaknesses for general optimization problems.

ACKNOWLEDGMENT

We are grateful to Prof. Yongsheng Ding for his contributions to the work and for his technical assistance. We are very sad that he passed away unexpectedly during the work.

REFERENCES

[1] R. Want, B. N. Schilit, S. Jenson, “Enabling the Internet of Things,” *Computer*, vol. 48, no. 1, pp. 28-35, Jan. 2015.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, “Internet of Things for smart cities,” *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22-32, Feb. 2014.

[3] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, “Sensing as a service model for smart cities supported by Internet of Things,” *Eur. Trans. Telecomm.*, vol. 25, no. 1, pp. 81-93, Jan. 2014.

[4] L. D. Xu, W. He, S. C. Li, “Internet of Things in industries: A survey,” *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2233-2243, Nov. 2014.

[5] F. Tao, Y. Zuo, L. D. Xu, L. Zhang, “IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing,” *IEEE Trans. Ind. Inform.*, vol. 10, no. 2, pp. 1547-1557, May. 2014.

[6] Y. S. Ding, Y. L. Jin, L. H. Ren, K. R. Hao, “An intelligent self-organization scheme for the Internet of Things,” *IEEE Comput. Intell. Mag.*, vol. 8, no. 3, pp. 41-53, Aug. 2013.

[7] M. Gigli, S. Koo, “Internet of Things: Services and applications categorization,” *Advances Internet Things.*, vol. 1, no. 2, pp. 27-31, Jul. 2011.

[8] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and applications,” *IEEE Commun. Surv. Tutor.*, vol. 17, no. 4, pp. 2347-2376, Nov. 2015.

[9] G. Fortino, C. Savaglio, and M. C. Zhou, “Toward opportunistic services for the industrial Internet of Things,” in *Proc. IEEE Conf. Autom. Sci. Eng (CASE).*, Xi'an, China, Aug. 2017, pp. 825-830.

[10] A. Zaslavsky, C. Perera, D. Georgakopoulos, “Sensing as a service and big data,” in *Proc. Int. Conf. Cloud Comput (ACC).*, Bengaluru, India, Jul. 2012, pp. 21–29.

[11] C. Perera, A. Zaslavsky, C. H. Liu, M. Compton, P. Christen, D. Georgakopoulos, “Sensor search techniques for sensing as a service architecture for the Internet of Things,” *IEEE Sens. J.*, vol. 14, no. 2, pp. 406-420, Feb. 2014.

[12] M. E. Khanouche, Y. Amirat, A. Chibani, M. Kerkar, A. Yachir, “Energy-centered and QoS-aware services selection for Internet of Things,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1256-1269, Jul. 2016.

[13] T. Baker, M. Asim, H. Tawfik, B. Aldawsari, R. Buyya, “An energy-aware service composition algorithm for multiple cloud-based IoT applications,” *J. Netw. Comput. Appl.*, vol. 89, pp. 96-108, Jul. 2017.

[14] L, Li, S. C. Li, S. S. Zhao, “QoS-aware scheduling of services-oriented Internet of Things,” *IEEE Trans. Ind. Inform.*, vol. 10, no. 2, pp. 1497-1505, May. 2014.

[15] V. Angelakis, I. Avgouleas, N. Pappas, E. Fitzgerald, D. Yuan, “Allocation of heterogeneous resources of an IoT device to flexible services,” *IEEE Internet Things J.*, vol. 3, no, 5, pp. 691-700, Oct. 2016.

[16] R. C. Liu, Y. Y. Chen, W. P. Ma, C. H. Mu, L. C. Jiao, “A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model,” *Soft Comput.*, vol. 18, no. 10, pp. 1913-1929, Oct. 2014.

[17] S. Y. Jiang, S. X. Yang, “Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons,” *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 198-211, Jan. 2017.

[18] T. T. Nguyen, S. X. Yang, J. Branke, “Evolutionary dynamic optimization: A survey of the state of the art,” *Swarm Evol. Comput.*, vol. 6, pp. 1-24, Oct. 2012.

[19] C. K. Goh, K. C. Tan, “A competitive-cooperative co-evolutionary paradigm for dynamic multi-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103-127, Feb. 2009.

[20] A. M. Zhou, Y. C. Jin, Q. F. Zhang, “A population prediction strategy for evolutionary dynamic multiobjective optimization,” *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40-53, Jan. 2014.

[21] Y. Wu, Y. C. Jin, X. X. Liu, “A directed search strategy for evolutionary dynamic multiobjective optimization,” *Soft Comput.*, vol. 19, no. 11, pp. 3221-3235, Nov. 2015.

[22] Z. Peng, J. H. Zheng, J. Zou, M. Liu, “Novel prediction and memory strategies for dynamic multiobjective optimization,” *Soft Comput.*, vol. 19, no. 9, pp. 2633-2653, Sep. 2015.

[23] R. C. Liu, J. Fan, L. C. Jiao, “Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm,” *Appl. Intell.*, vol. 43, no. 1, pp. 192-207, Jul. 2015.

[24] M. Mestari, M. Benzirar, N. Saber, M. Khouil, “Solving nonlinear equality constrained multiobjective optimization problems using neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2500-2520, Oct. 2015.

[25] R. H. Shang, L. C. Jiao, Y. J. Ren, J. Wang, Y. Y. Li, “Immune clonal coevolutionary algorithm for dynamic multiobjective optimization,” *Nat. Comput.*, vol. 13, no. 3, pp. 421-445, Sep. 2014.

[26] Y. S. Ding, Z. H. Hu, W. B. Zhang, “Multi-criteria decision making approach based on immune co-evolutionary algorithm with application to garment matching problem,” *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10377-10383, Aug. 2011.

[27] Z. H. Hu, Y. S. Ding, Q. Shao, “Immune co-evolutionary algorithm based partition balancing optimization for tobacco distribution system,” *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5248-5255, Apr. 2009.

[28] S. Nguyen, M. J. Zhang, M. Johnston, K. C. Tan, “Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 193-208, Apr. 2014.

[29] G. S. Yao, Y. S. Ding, Y. C. Jin, K. R. Hao, “Endocrine-based coevolutionary multi-swarm for multi-objective workflow scheduling in a cloud system,” *Soft Comput.*, vol. 21, no. 15, pp. 4309-4322, Aug. 2017.

[30] S. Y. Jiang, S. X. Yang, “A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65-82, Feb. 2017.

[31] M. Jiang, Z. Q. Huang, L. M. Qiu, W. Z. Huang, G. G. Yen, “Transfer learning based dynamic multiobjective optimization algorithms,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501-514, Aug. 2018.

[32] Q. Kang, X. Y. Song, M. C. Zhou, L. Li, “A collaborative resource allocation strategy for decomposition-based multi-objective evolutionary algorithms,” *IEEE Trans. Syst. Man Cybern. -Syst.*, pp. 1-8, May. 2018. [Online]. DOI: 10.1109/TSMC.2018.2818175.

[33] S. Das, A. Abraham, U. K. Chakraborty, A. Konar, “Differential evolution using a neighborhood-based mutation operator,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526-553, Jun. 2009.

[34] S. Das, A. Mandal, R. Mukherjee, “An adaptive differential evolution algorithm for global optimization in dynamic environment,” *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 966-978, Jun. 2014.

[35] H. Zhang, A. M. Zhou, S. M. Song, Q. F. Zhang, X. Z. Gao, J. Zhang, “A self-organizing multiobjective evolutionary algorithm,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 792-806, Oct. 2016.

[36] G. H. Nguyen, J. H. Shin, W. H. Kim, “Autotuning controller for motion control system based on intelligent neural network and relay feedback approach,” *IEEE-ASME Trans. Mechatron.*, vol. 20, no. 3, pp. 1138-1148, Jun. 2015.

[37] Q. L. Wei, D. R. Liu, X. Yang, “Infinite horizon self-learning optimal control of nonaffine discrete-time nonlinear systems,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 866-879, Apr. 2015.

[38] D. Wang, C. X. Mu, H. B. He, D. R. Liu, “Event-driven adaptive robust control of nonlinear systems with uncertainties through NDP strategy,” *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 47, no. 7, pp. 1358-1370, Jul. 2017.

[39] Y. S. Ding, N. Xu, S. F. Dai, L. H. Ren, K. R. Hao, B. Huang, “An immune system-inspired reconfigurable controller,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 5, pp. 1875-1882, Sep. 2016.

[40] Y. Peng, B. L. Lu, “Hybrid learning clonal selection algorithm,” *Inf. Sci.*, vol. 296, pp. 128-146, Mar. 2015.

[41] P. Campigotto, A. Passerini, R. Battiti, “Active learning of Pareto fronts,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 506-519, Mar. 2014.

[42] M. Zuluaga, A. Krause, M. Püschel, “ε-PAL: An active learning approach to the multi-objective optimization problem,” *J. Mach. Learn. Res.*, vol. 17, pp. 1-32, Aug. 2016.

[43] T. Alpcan, I. Shames, “An information-based learning approach to dual control,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2736-2748, Nov. 2015.

[44] Z. Z. Liu, D. H. Chu, C. Song, X. Xue, B. Y. Lu, “Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition,” *Inf. Sci.*, vol. 326, pp. 315-333, Jan. 2016.

[45] D. Barkoczi, M. Galesic, “Social learning strategies modify the effect of network structure on group performance,” *Nat. Commun.*, vol. 7, pp. 1-8, Oct. 2016.

[46] M. A. J. Apps, J. Sallet, “Social learning in the medial prefrontal cortex,” *Trends Cogn. Sci.*, vol. 21, no. 3, pp. 151-152, Mar. 2017.

[47] J. Haaker, J. Yi, P. Petrovic, A. Olsson, “Endogenous opioids regulate social threat learning in humans,” *Nat. Commun.*, vol. 8, pp. 1-9, May. 2017.

[48] M. S. Reed, A. C. Evely, G. Cundill, I. Fazey, J. Glass, A. Laing, J. Newig, B. Parrish, C. Prell, C. Raymond, L. C. Stringer, “What is social learning,” *Ecol. Soc.*, vol. 15, no. 4, pp. 1-10, Dec. 2010.

[49] A. N. Meltzoff, P. K. Kuhl, J. Movellan, T. J. Sejnowski, “Foundations for a new science of learning,” *Science*, vol. 325, no. 5938, pp. 284-288, Jul. 2009.

[50] D. Kumaran, A. Banino, C. Blundell, D. Hassabis, P. Dayan, "Computations underlying social hierarchy learning: distinct neural mechanisms for updating and representing self-relevant information," *Neuron*, vol. 92, no. 5, pp. 1135-1147, Dec. 2016.

[51] Z. H. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Appl. Soft. Comput.*, vol. 8, no. 2, pp. 959-971, Mar. 2008.

[52] Y. Liu, Y. Peng, B. L. Wang, S. R. Yao, Z. H. Liu, "Review on cyber-physical systems," *IEEE-CAA J. Autom. Sinica*, vol. 4, no. 1, pp. 27-40, Jan. 2017.

[53] T. C. Fung, C. A. Olson, E. Y. Hsiao, "Interactions between the microbiota, immune and nervous systems in health and disease," *Nature Neuroscience*, vol. 20, no. 2, pp. 145-155, Feb. 2017.

[54] M. G. Gong, L. C. Jiao, H. F. Du, L. F. Bo, "Multi-objective immune algorithm with non-dominated neighbor-based selection," *Evol. comput.*, vol. 16, no. 2, pp. 225-255, Jun. 2008.

[55] Q. Z. Lin, J. Y. Chen, Z. H. Zhan, W. N. Chen, C. A. Coello Coello, Y. L. Yin, etc., "A hybrid evolutionary immune algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 711-729, Oct. 2016.

[56] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, Aug. 2002.

[57] E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117-132, Apr. 2003.

[58] Q. F. Zhang, H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712-731, Dec. 2007.

[59] R. Cheng, Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191-204, Feb. 2015.

Zhen Yang is currently pursuing the Ph.D. degree at College of Information Sciences and Technology, Donghua University, Shanghai, China. She received the M.S. degree in computer application from the Faculty of Information Engineering, China University of Geosciences, Wuhan, China in 2007. Her research interests include computational intelligence, evolutionary optimization, and intelligent Internet of Things.

Yaochu Jin (M'98-SM'02-F'16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Bochum, Germany, in 2001.

He is a Professor of Computational Intelligence with the Department of Computing, University of Surrey, Guildford, U.K., where he is the Head of the Nature Inspired Computing and Engineering Group. He is also a Finland Distinguished Professor funded by the Finnish Agency for Innovation (Tekes), Helsinki, Finland, and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, Beijing, China. He has co-authored over 200 peer-reviewed journal and conference papers and has been granted eight patents on evolutionary optimization. His research interests include computational intelligence, computational neuroscience, computational systems biology, nature-inspired, and real-world-driven problem-solving.

He is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Complex & Intelligent Systems. He is an IEEE Distinguished Lecturer and was the Vice President for Technical Activities of the IEEE Computational Intelligence Society (2014-2015). He is the recipient of the 2014 and 2016 IEEE Computational Intelligence Magazine Outstanding Paper Award, and the 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Award. He is a Fellow of IEEE.

Kuangrong Hao (M'17) is currently a Professor at the College of Information Sciences and Technology, Donghua University, Shanghai, China. She obtained her B.S. degree in Mechanical Engineering from Hebei University of Technology, Tianjin, China in 1984, her M.S. degree in applied mathematics from Ecole Normale Supérieur de Cachan, Paris, France in 1991, and her Ph.D. degree in Mathematics and Computer Science from Ecole Nationale des Ponts et Chaussées, Paris, France in 1995. She has published more than 100 technical papers, and three research monographs. Her scientific interests include machine vision, image processing, robot control, intelligent control, and digitized textile technology.