

## Research Article

# Optimization on Black Box Function Optimization Problem

Jin-ke Xiao,<sup>1</sup> Wei-min Li,<sup>1</sup> Wei Li,<sup>2</sup> and Xin-rong Xiao<sup>3</sup>

<sup>1</sup>Air Force Engineering University, Xi'an 710051, China

<sup>2</sup>Northwestern Polytechnical University, Xi'an 710072, China

<sup>3</sup>South China University of Technology, Guangzhou 510640, China

Correspondence should be addressed to Jin-ke Xiao; xjk521125@126.com

Received 27 June 2015; Revised 16 September 2015; Accepted 16 September 2015

Academic Editor: Anna Vila

Copyright © 2015 Jin-ke Xiao 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.

There are a large number of engineering optimization problems in real world, whose input-output relationships are vague and indistinct. Here, they are called black box function optimization problem (BBFOP). Then, inspired by the mechanism of neuroendocrine system regulating immune system, BP neural network modified immune optimization algorithm (NN-MIA) is proposed. NN-MIA consists of two phases: the first phase is training BP neural network with expected precision to confirm input-output relationship and the other phase is immune optimization phase, whose aim is to search global optima. BP neural network fitting without expected fitting precision could be replaced with polynomial fitting or other fitting methods within expected fitting precision. Experimental simulation confirms global optimization capability of MIA and the practical application of BBFOP optimization method.

## 1. Introduction

Black box function optimization problem (BBFOP), inspired from black box concept which originated from cybernetics, is one kind of mathematic optimization functions. Interior function and characteristic of BBFOP could only be understood through exterior observation and experimentation for some condition restriction reasons. Usually, the exterior impacts on BBFOP are called inputs and the function feedback from inputs is called outputs.

A large number of complex engineering BBFOPs present ambiguity relationships between inputs and outputs, much optimization computational complexity issue in real manufacturing industry, in aerospace industry, and in computing industry. BBFOP is hard to be solved for the reason that input-output relationships are vague and indistinct. Therefore, finding the global optima or the local optima nearest to the global optima (if we cannot find the global optima) would provide the best solutions to the decision-makers.

When we solve BBFOP, two difficulties must be given top priority: one is input-output relationship and the other one is the optimization method. There are two solutions to

input-output relationship problem: one is giving BBFOP expression directly through studying interior structure and exploring interior controlling mechanism, which is almost impossible, and the other one is using fitting function as an indirect description of input-output relationship. The latter method enjoys wide application in the engineering practice for fitting function could be born from plentiful input-output sample data. The representative fitting method includes BP neural network fitting and polynomial fitting.

For BBFOP optimization method, population-based optimization algorithms enjoy high popularity and have extensive applications in recent years in engineering optimization problems, including genetic algorithms (GAs) [1, 2], ant colony optimization (ACO) [3, 4], particle swarm optimization (PSO) [5, 6], artificial bee colony (ABC) [7, 8], bacterial foraging optimization (BFO) [9, 10], big bang-big crunch algorithm [11, 12], and teaching-learning-based optimization (TLBO) [13–15] and immune optimization algorithm (IA) [16–19]. These population-based optimization algorithms have one or more populations consisting of a certain number of individuals, which presents a solution of the problem to be solved, respectively. Fitness of individuals in population

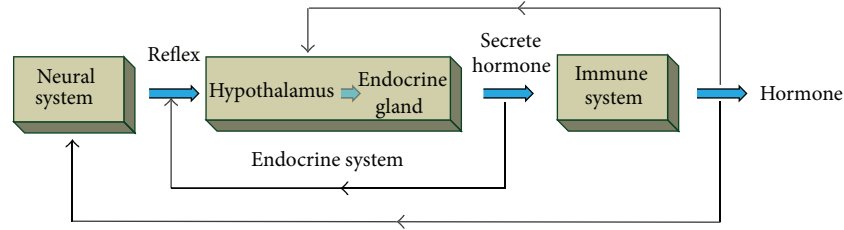


FIGURE 1: INE system regulation concept view.

moves towards better one by applying some operators in the search space. Immune optimization algorithm (IA), a vital research and promising direction of Artificial Immune System (AIS), is a novel population-based intelligent algorithm based on the theoretical immune principles [20] inspired by adaptive immune system. Due to the advantages of simplicity and ease of implementation and the remarkable optimization capability, IA and its variants have attracted the attention of researchers and have been widely exploited to solve a great variety of related engineering optimization problems [19, 21–25].

Body neural system, immune system, and endocrine system (Immune-Neuro-Endocrine, INE) have super information processing mechanism with characteristic of distribution and robustness. The interaction mechanism among neural system, endocrine system, and immune system gives us some inspirations for intelligent application on optimization [26, 27] while the mechanism where neural system and endocrine system enhance or suppress immune system function may be another useful source of inspirations for population-based algorithm for BBFOP.

Here, the single object optimization BBFOP is under research for the reason that the single object optimization problem is the base for multiobject optimization problem. BBFOP optimization method is inspired from the mechanism of how immune system is regulated by neural system and endocrine system. BP neural network or other fitting methods are used to fit input-output sample data with ideal precision while IA and its variants are used to optimize input-output relationship.

The remaining text in this paper is organized as follows. Section 2 gives a detailed biological mechanism description and its inspiration of neuroendocrine system regulating immune system mechanism; Section 3 develops BBFOP algorithm statement and design; in Section 4, three experiments will be done to attest method validity; in Section 5, conclusions on solving BBFOP and modified optimization method directions are given.

## 2. Neuroendocrine System Regulating Immune System Mechanism

INE system is stereonetwork regulation system with high precision and high complexity through cytokine, hormone, and other chemistry substances [28, 29]. Physical and chemical biological characteristic of medium and hormone secreted by neuroendocrine system and lymph gene, monocyte secreted by immune system, and existing receptor in INE system cell

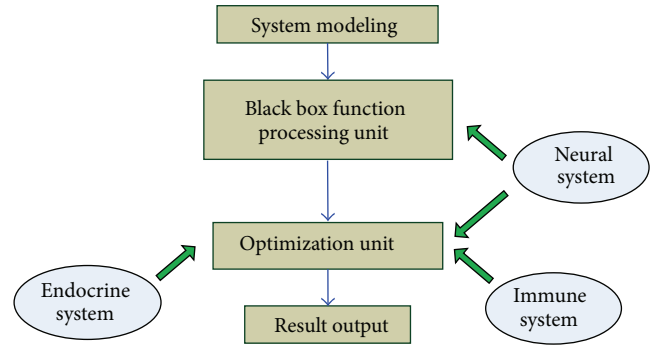


FIGURE 2: BBFOP optimization flow.

surface are gradually discovered with the molecule-biology development.

Immune system could eliminate the invading antigen through the immune response at first once the body is attacked by exterior antigen. If one antibody matches the antigen with higher affinity, it reproduces and eliminates antigen; else, other antibody populations coevolve to produce antibody matching the antigen with high affinity. Meanwhile, the real-time status of immune system is reflected to neuroendocrine system; then neuroendocrine system coregulates immune system to eliminate antigen faster through neurotransmitter and multihormone. Visual concept view is in Figure 1.

Neuroendocrine system strengthens immune effect through neurotransmitter and multihormone. This mechanism gives a source inspiration for solving BBFOP, which inspires us to propose the optimization method in Figure 2.

In Figure 2, black box function processing unit consists of input-output relationship which is calculated by neural network; optimization unit is the mechanism simulation where neuroendocrine system regulates immune system eliminating invading antigen and searches ideal solution based on input-output relationship.

## 3. Algorithm Statement and Design

**3.1. Algorithm Formulation.** As the minimal optimization problem could be converted into corresponding maximal one, the following nonlinear optimization problem of form in (1) is considered:

$$\max f(X) = f(x_1, x_2, \dots, x_n), \quad (1)$$

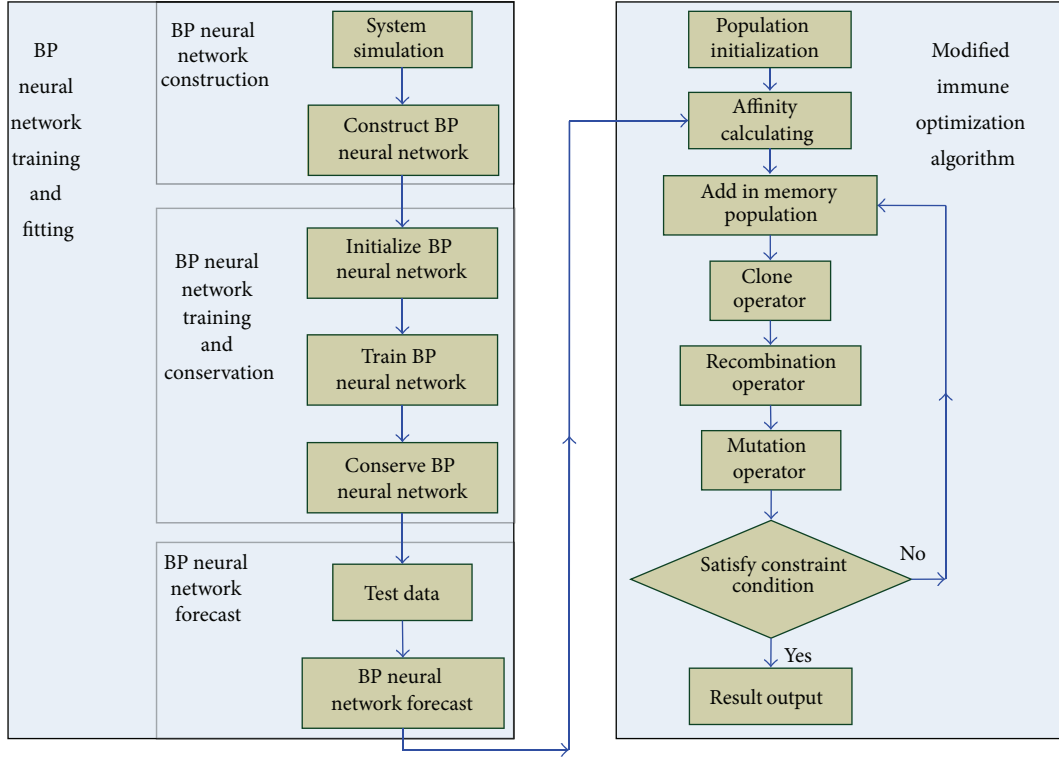


FIGURE 3: Algorithm flow.

where  $f(X)$  is unknown single function optimization BBFOP and also is a time continuous real function with  $n$ -dimension argument  $X = (x_1, x_2, \dots, x_n)$ ;  $x_i$  ( $i = 1, 2, \dots, n$ ) is in the interval  $[l_i, h_i]$  ( $i = 1, 2, \dots, n$ ), respectively.

Here, a flowchart of NN-MIA is given in Figure 3. Figure 3 consists of BP neural network training and fitting and modified immune optimization algorithm (MIA), where MIA has a loop deciding the procedure ends.

In Box 1, Steps 1 and 2 are BP neural network training and fitting and a loop of optimization, respectively. Particularly, in Step 2.5, mutation operator is to be implemented on antibody population  $[C^c, P^r]$  instead of antibody population  $P^r$ , which could make full use of antibody information before combination; in Step 2.6, selecting the best antibody with the highest affinity from antibody population  $[C^c, P^r, P^m, P^d]$  and adding the best antibody into memory antibody population could make full use of antibody information before combination and mutation.

### 3.2. Algorithm Illustration

**3.2.1. BP Neural Network Construction Phase.** Input-output layer and hidden network layer of BP neural network should be constructed according to experimental conditions and “ $2N + 1$ ” principle proposed by Hecht-Nielsen [30].

### 3.2.2. Immune Optimization Phase

(1) *Initialization Operator.* The algorithm generates populations corresponding to the solutions in the search space. The

solutions are produced randomly within the range of the boundaries of the variables in

$$x_{i,j} = x_j^{\min} + \text{rand} \cdot (x_j^{\max} - x_j^{\min}), \quad (2)$$

$$(i = 1, 2, \dots, P; j = 1, 2, \dots, n).$$

$P$  is the individual number in population,  $n$  is the dimension of optimization problem, representing the number of parameters to be optimized, and  $x_j^{\min}$  and  $x_j^{\max}$  are lower and upper bounds of the  $j$ th parameter.

(2) *Clone Operator.* Antibody clone scale is adjusted adaptively and dynamically according to the affinity between antibodies, which is inspired from the inhibitory characteristic mechanism of immune response.

Here is the definition of antibody affinity  $\Phi_i$  between antibody  $i$  and the other antibody  $j$  in

$$\Phi_i = \min \left( \exp \left( \|X_i - X_j\| \right) \right)$$

$$= \min \left( \exp \left( \sqrt{\sum_{ii=1}^n (x_{ii}^i - y_{ii}^j)^2} \right) \right), \quad (3)$$

$$(i \neq j, i, j = 1, 2, \dots, P),$$

where  $\|X_i - X_j\|$  is the Euclidean distance between antibody  $i$  and antibody  $j$ , which is in the interval  $[0, 1]$  after normalization processing. The higher the affinity between antibody and the other antibody is, the higher their comparability value is

**Step 1. BP neural network training and fitting**

Step 1.1. draw out mathematic optimization model from engineering problem and produce plenty of sample data

Step 1.2. construct appropriate BP neural network based on the sample data and the model complexity

Step 1.3. initialize BP neural network, train and conserve neural network

Step 1.4. test BP neural network through some sample data to forecast the output, that is affinity in the following MIA

**Step 2. modified immune optimization algorithm**

Step 2.1. population initialization: initialize antibody population scale  $P$ , memory population scale  $M$ , clone population scale  $C$  and antibody population  $P^p$

Step 2.2. affinity calculating: calculate antibody-antigen affinity based on the trained BP neural network

Step 2.3. clone operator: clone antibody population  $P^p$  to population scale  $C$  and produce antibody population  $C^c$

Step 2.4. combination operator: combine antibody population  $C^c$  to produce antibody population  $P^r$

Step 2.5. mutation operator: mutate antibody population  $[C^c, P^r]$  to produce antibody population  $P^m$

Step 2.6. antibody population updating: produce antibody population  $P^d$  randomly with population scale  $d$

Step 2.7. add in memory antibody population: select  $M$  antibody with the highest affinity from  $[C^c, P^r, P^m, P^d]$  and add in memory antibody population

Step 2.8. if constrained condition satisfaction is true, skip to Step 8; else skip to Step 2

Step 2.9. select the antibody with highest affinity from the antibody population as the satisfying output result

Box 1: Pseudoprocedure flow.

and the smaller the value of  $\Phi_i$  is, which means the stronger inhibitory effect.  $C_i$  stands for the clone scale of antibody  $i$  and is calculated in

$$C_i = \text{ceil} \left( C * \Phi_i * \frac{f(X_i)}{\sum_{i=1}^n f(X_i)} \right), \quad (4)$$

where  $C$  is the clone scale initialized before and function  $\text{ceil}(x)$  stands for the elements of  $x$  to the nearest integers which can be seen in MATLAB 7.11.0.

(3) *Recombination Operator*. Suppose  $X^{(1)}$  and  $X^{(2)}$  are two vectors in  $n$ -dimension space. Then, select a dot between two vector regions randomly to produce offspring antibody  $X^{\text{new}}$ , where  $X^{\text{new}} = a \cdot X^{(1)} + (1 - a) \cdot X^{(2)}$ ,  $a = \text{rand}$ .

Antibody combination operator executing flow is in Box 2.

(4) *Mutation Operator*. Hypermutation, altering on some genes with light probability, leads to affinity maturation. Usually, uniform mutation is used, a mutation combining the result of mutation operator with iterative times. Mutation range is larger relatively at the early stage while the latter is; mutation range is smaller and smaller, which will have fine-tuning impact on evolution system. Uniform mutation is described as follows.

Suppose father antibody  $X = (x_1, x_2, \dots, x_n)$   $x_k \in [a_k, b_k]$  is selected to mutate, where  $a_k$  and  $b_k$  are lower and upper bounds of  $x_k$ ; offspring after mutation is in

$$X' = (x_1, \dots, x_{k-1}, x'_k, \dots, x_n),$$

$$x'_k = \begin{cases} x_k + \Delta(it, b_k - x_k), & \text{rnd}(2) = 0, \\ x_k - \Delta(it, b_k - x_k), & \text{rnd}(2) = 1, \end{cases} \quad (5)$$

where  $\text{rnd}(2)$  is the modulus after division 2,  $it$  is the current iteration, and range of function  $\Delta(it, y)$  is  $[0, y]$ . The bigger it is, the larger probability that  $\Delta(it, y)$  is close to 0, which

enables the algorithm search across a large scale in the early evolution stage and search across a local scale in the later evolution stage.  $r \in [0, 1]$  is random number,  $T$  is the maximum iteration, and  $\lambda \in [2, 5]$  is the parameter that decides uniform mutation degree, which adjusts the local search region.

Main steps of uniform mutation operator are in Box 3, where  $b_u(i, k)$  and  $b_d(i, k)$  are lower and upper bounds of  $k$ th component in  $i$ th antibody, respectively.

## 4. Experiments

Experimental procedures and results will be presented and discussed in this section. All algorithms were executed in MATLAB 7.11.0 using computer with Intel Pentium CPU G640, 2.80 GHZ, 2 GB RAM. The operating system of the computer is Windows 7.

The global optimization performance of MIA is tested and compared with other optimization algorithms on several well-known benchmark functions including (A) Rosenbrock function, (B) Step function, (C) Quadric function, (D) Schwefel's function, (E) Rastrigin function, (F) Ackley's function, (G) Griewank function, and (H) Rotate hyperellipsoid function in the Appendix.

In addition, it should be mentioned that, as these population optimization algorithms have stochastic search, which would lead to some changing results in different iterations to some degree, we observe the box plot of optimization function with 30 dimensions (in Section 4.1) obtained by each algorithm based on 60 independent runs.

Section 4.1 reports a comparison between MIA experimental results and three known immune algorithms: SAMCCTLBO, CLONALG, and HPGA; Section 4.2 reports BP neural network-MIA solving BBFO; Section 4.3 reports polynomial fitting-MIA solving BBFO.

*4.1. Experiment 1: Performance Comparisons of MIA with SAMCCTLBO, CLONALG, and HPGA.* In order to examine

Step 1. initialization  $j := 1$   
 Step 2.  
 Step 2.1. produce random number  $a \in [0, 1]$   
 Step 2.2. produce  $X^{\text{new}} = a \cdot X^{(1)} + (1 - a) \cdot X^{(2)}$  after combination  
 Step 3.  $j := j + 1$ , if termination condition reach, iterative terminates;  
 else, skip to Step 2.

Box 2: Antibody combination operator executing flow.

Step 1. initialization  $i := 1$   
 Step 2.  $k = \text{round}(\text{rand} \cdot (n - 1) + 1)$ , where  $n$  is the dimension of decision-making vector  
 Step 2.1. select  $k$ th component in  $i$ th antibody  $A(it)(i, k)$  from antibody population  $A(it)$   
 Step 2.2. select positive integer  $a$  randomly and calculate its modulus after division 2  
 Step 2.3. if  $\text{rnd}(2) == 1$   

$$A(it)(i, k) = A(it)(i, k) + (b_u(i, k) - A(it)(i, k)) \left(1 - r^{(1-it/T)^2}\right)$$
 else  

$$A(it)(i, k) = A(it)(i, k) - (A(it)(i, k) - b_d(i, k)) \left(1 - r^{(1-it/T)^2}\right)$$
  
 Step 3.  $i := i + 1$ , if termination condition reach, iterative terminates;  
 else, skip to Step 2.

Box 3: Main steps of uniform mutation operator.

the proposed MIA, three known algorithms (a multiclass cooperative teaching-learning-based optimization algorithm with simulated annealing (SAMCCTLBO) [31], Clone Selection Algorithm (CLONALG) [32], and Hierarchical Parallel Genetic Algorithms (HPGA) [33]) are selected for direct comparisons.

The number of function evaluations (FEs) is becoming a more and more appalusive and reasonable measurement termination criterion [34–36] for measuring different algorithms as computational complexity taken in each iteration may exhibit difference in different algorithms. Therefore, combined with search result precision, the termination criterion of the algorithms is one of the objectives,  $|f_{\text{best}}| \leq \varepsilon$ ,  $\varepsilon = 10^{-5}$ , where  $f_{\text{best}}$  is the function value obtained by the optimization algorithms or a maximum number of function evaluations (FEs) that have been reached,  $Fes = 10000$ ; here  $\varepsilon = 10^{-5}$  is used for all functions. If  $|f_{\text{best}}| \leq \varepsilon$ , this algorithm is considered the best result that has been obtained.

Box plots [37] are used to illustrate the distribution of these samples obtained from 60 independent runs. The upper and lower ends of the box are the upper and lower quartiles while a line within the box denotes the median and thin appendages summarize the spread and shape of the distribution. The notches represent a robust estimate of the uncertainty about the medians for box-to-box comparison. Symbol “+” denotes outliers. Here, box plots are used to illustrate the distribution of the function values obtained from 60 independent runs.

Detailed parameter settings of SAMCCTLBO, CLONALG, and HPGA can be seen in the respective literature and in MIA. The higher the parameters are, the more excellent

the results are, the more complex the algorithm's space and time complexity are, and the longer the running time is. Therefore, parameters are set to  $P = 100$ ,  $M = 50$ ,  $C = 200$ ,  $r = 3$ ,  $\lambda = 3$ ,  $d = 10$ , and  $T = 500$  according to several experiments' testing. Once they have been all experimentally determined, they are kept the same through all experiments except that MIA parameters are set again. Then, for a summarized performance comparison for different algorithms solving unimodal function, unrotated multimodal function problems are presented in Figure 4.

Then, NN-MIA formulation pseudoprocedure is in Box 1 correspondingly.

From the results in Figure 4 box plot, the upper quartiles, lower quartiles, and medians obtained by MIA are much less than those obtained by CLONALG and HPGA and, in addition, are much less or not much more than those obtained by SAMCCTLBO. Furthermore, for (A) Rosenbrock function, (B) Step function, (E) Rastrigin function, (F) Ackley's function, and (G) Griewank function, MIA and SAMCCTLBO, CLONALG, and HPGA all can obtain global optimal 0 in all the 60 runs with accepted standard deviation, but MIA generates the best results while SAMCCTLBO performs better than CLONALG and HPGA. For (C) Quadric function and (D) Schwefel's function, MIA and SAMCCTLBO can obtain global optimal 0 in all the 60 runs, and CLONALG and HPGA could obtain global results near to global optima and beyond the results precision, while MIA obtains global optima with smaller standard deviation than that of SAMCCTLBO. For (H) Rotate hyperellipsoid function, only MIA can obtain the best optima near the results precision. Here, it can be easily deduced from the box plots that MIA has a much higher capability of finding the global optima.



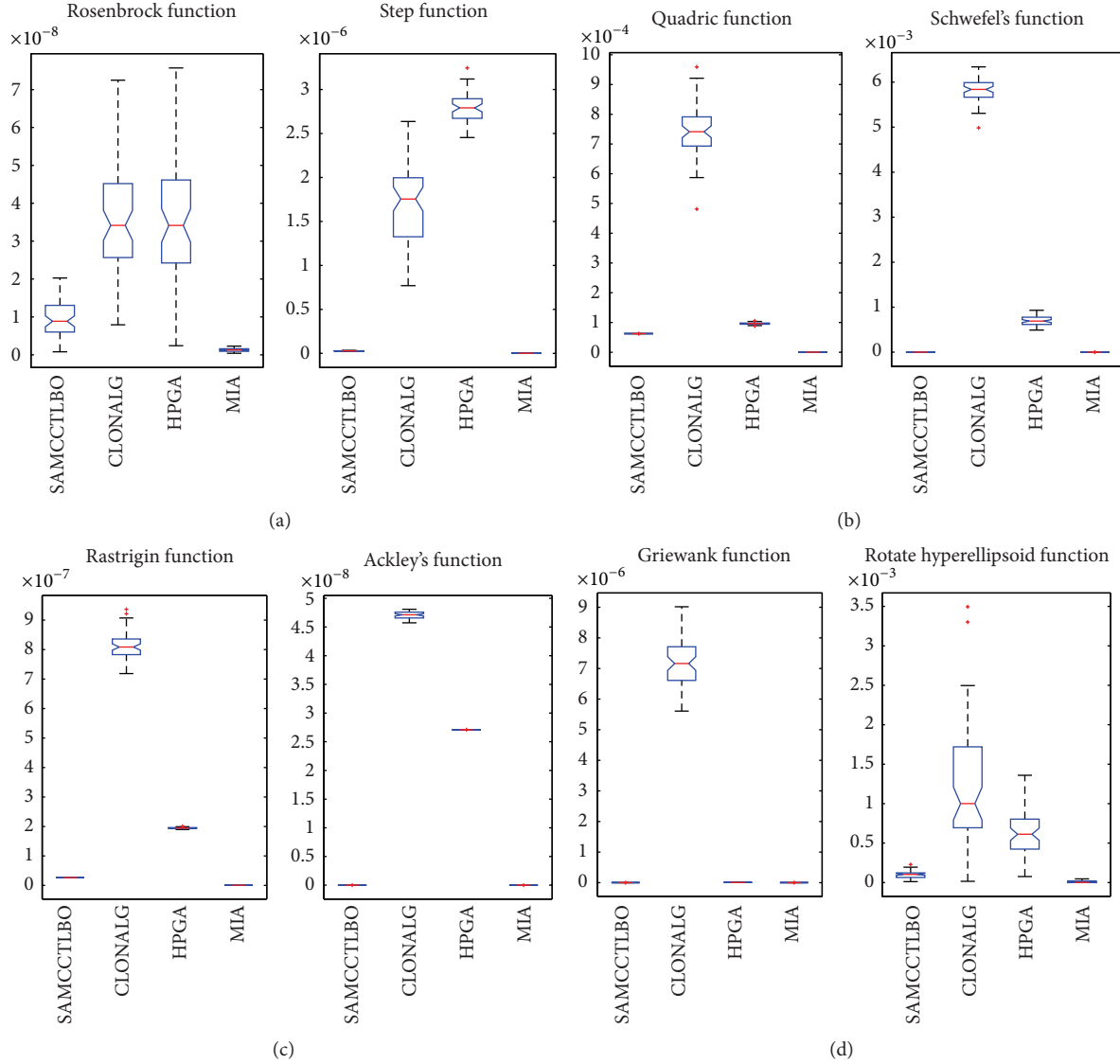


FIGURE 4: Performance comparison among optimization algorithms for the test functions.

4.2. Experiment 2: BP Neural Network-MIA Solving BBFO. Here, NN-MIA is used to optimize the following nonlinear function in

$$\max f(x) = \frac{1}{1 + \sum_{l=1}^n x_l^2}, \quad (6)$$

$$-5 \leq x_l \leq 5, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 1$  with  $[x_1 \dots x_n] = [0 \dots 0]$ .

The function graphic is in Figure 5 when  $n = 2$ .

It is easy to conclude that the function global maximum is 1 and its corresponding coordinate is  $(0, 0)$  from the above function expression in (6) and its function graphic. Function extremum and its corresponding coordinate are easy to be concluded in a known instead of an unknown function expression case.

Suppose  $n = 8$  in (6) which means the corresponding black box function has 8 input parameters and 1 output

parameter. Then, BP neural network is constructed: suppose BP neural network has single hidden network, the neuron number is set to be 17 according to "2N + 1" method, and transferring function is Sigmoid function in every neuron. Finally, topology structure 8-17-1 BP neural network with 8 inputs, 17 hidden layers, and 1 output is demonstrated in Figure 6.

Here, 8000 inputs and outputs are produced according to (6) and 7000 pieces of data are selected randomly to train BP neural network; the other 1000 pieces of data are used to test the fitting accuracy of BP neural network. Then, comparison of forecasting outputs and expected outputs is demonstrated in Figure 7. The accumulative error between forecasting outputs and expected outputs is limited within 1.1495, which is within the error requirement. Therefore, the input-output relationship could be solved by topology structure 8-17-1 BP neural network. Then, the following is optimizing results.

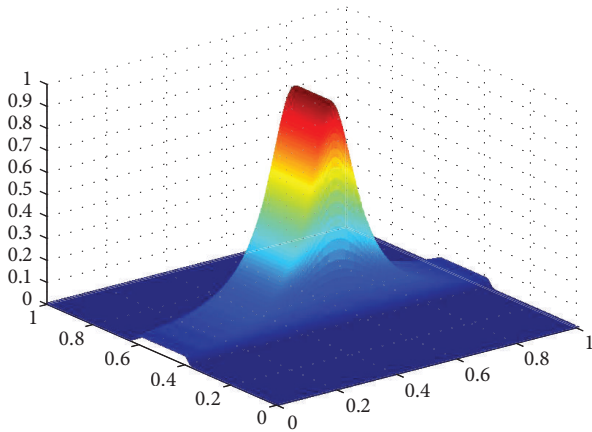


FIGURE 5: Function graphic when  $n = 2$ .

The efficiency of real number code is much higher than that of binary code usually; therefore, real coding is applied in MIA and antibody fitness is the BP neural network output; the larger the fitness is, the super the antibody is. Then, the best fitness is 0.999794 and its best antibody individual is  $[0.0002, -0.00009, -0.0003, 0.0001, 0.0003, -0.0004, -0.0003, -0.0009]$ , which is very close to real maximum 1 and its corresponding coordinate  $(0, 0)$ . The method is confirmed to be valid.

4.3. Experiment 3: Polynomial Fitting-MIA Solving BBFO. However, in experiment 2, this method has to be modified to some extent. Here, modified Rosenbrock function is in (7) and the function graphic is in Figure 8 when  $n = 2$ :

$$\max f(x) = \frac{1}{1 + \sum_{l=1}^{n-1} (100(x_{l+1} - x_l^2)^2 + (x_l - 1)^2)}, \quad (7)$$

$$-5 \leq x_l \leq 5, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 1$  with  $[x_1 \dots x_n] = [1 \dots 1]$ .

The same BP neural network in experiment is used here and the comparison of forecasting output and expected output is demonstrated in Figure 9. The accumulative error between forecasting output and expected output reaches 7.9212, which has gone beyond error requirement.

In this case, the fitness results of MIA will not be satisfied. Therefore, polynomial fitting method is used instead of BP neural network and the comparison of forecasting output and expected output is demonstrated in Figure 10. The accumulative error between forecasting output and expected output is limited within 1.2032, which accords with error requirement.

Here, MIA is used for optimization and the best fitness 0.9999981 is produced, which is very close to real maximum 1. The method of combining polynomial fitting with MIA is confirmed to be valid here.

### 5. Conclusion

BBFOP optimization algorithm is proposed in this paper, which is inspired by the mechanism of neuroendocrine

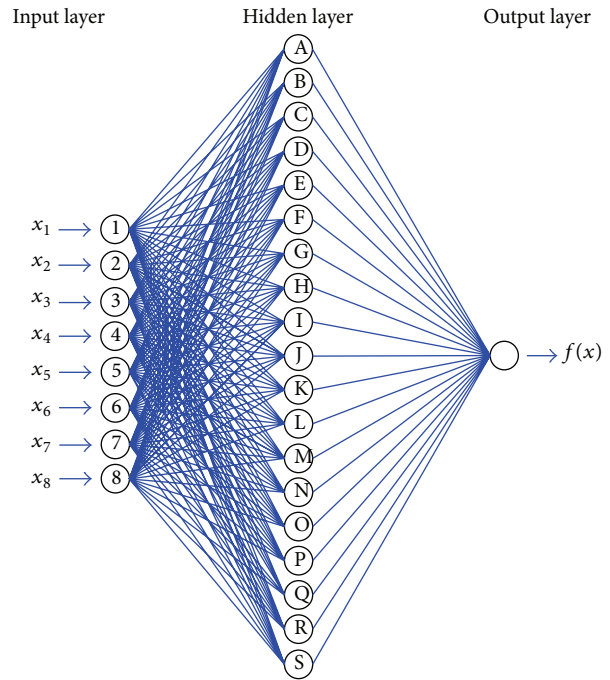


FIGURE 6: BP neural network designed to model in experimentation 1.

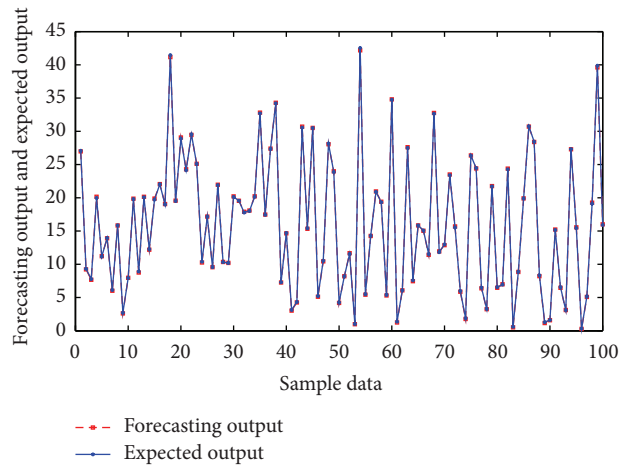


FIGURE 7: Comparison of forecasting output and expected output.

system regulating immune system. In this algorithm, BP neural network is used to fit this input-output relationship based on sample data. If the fitness precision is not reached, the polynomial fitting and other fitting methods are adopted, and then MIA are to optimize the fitting function.

(1) Engineering Value of Black Box Function and Its Optimization Method. BBFOP optimizing method enjoys important applications in some kinds of engineering problem. For example, one experiment aim is to conclude the experiment conditions according to the optimal experiment results; however, such experiment could only repeat numbered times and experiment conditions could not be reached just based

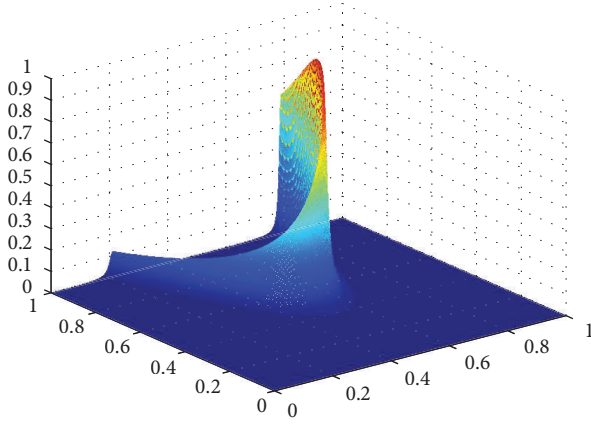
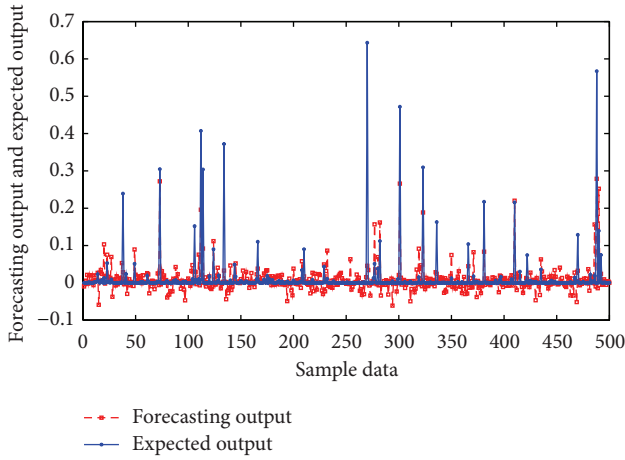
FIGURE 8: Function graphic when  $n = 2$ .

FIGURE 9: Comparison of forecasting output and expected output.

on experiment results. Similar engineering problems could be considered as BBFOP.

(2) *Function Fitting Methods Analysis on BBFOP.* The forecasting result precision of BP neural network determines final optimizing results to some extent. With higher forecasting result precision, the final optimizing results are closer to the real results, which requires more training sample data. However, not all black box function inputs and outputs could be expressed by BP neural network with expected precision. Therefore, other function fitting methods including polynomial fitting method could be applied to meet expected precision and the following steps are almost the same as the above.

In order to promote the development on the theory and application of BBFOP and its optimization method, our future research will focus on basic theory of BBFOP: biology theory including neural system and immune system should be explored at length to enhance neural network fitting capability and MIA capability; therefore, deeper researches should be carried out by means of theoretical analysis and related experiments.

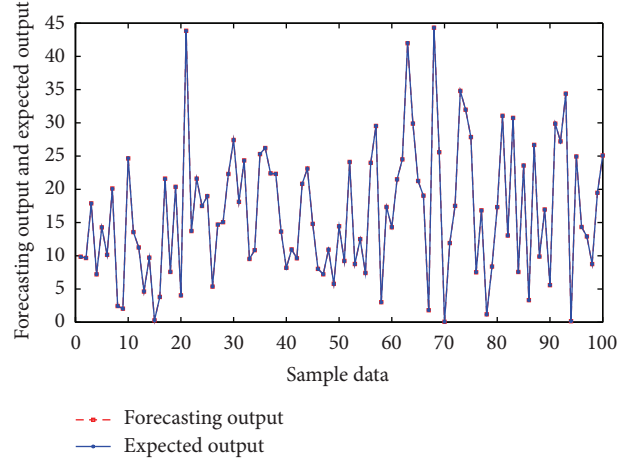


FIGURE 10: Comparison of forecasting output and expected output.

## Appendix

(A) Rosenbrock function is

$$\min f(x) = \sum_{l=1}^{n-1} \left( 100(x_{l+1} - x_l^2)^2 + (x_l - 1)^2 \right), \quad (\text{A.1})$$

$$-30 \leq x_l \leq 30, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [1 \cdots 1]$ .

(B) Step function is

$$\min f(x) = \sum_{l=1}^n \left( \lfloor (x_l + 0.5)^2 \rfloor \right), \quad (\text{A.2})$$

$$-100 \leq x_l \leq 100, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [0 \cdots 0]$ .

(C) Quadric function is

$$\min f(x) = \sum_{l=1}^n \left( \sum_{j=1}^l x_j \right)^2, \quad (\text{A.3})$$

$$-10 \leq x_l \leq 10, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [0 \cdots 0]$ .

(D) Schwefel's function is

$$\min f(x) = 418.9829 - \sum_{l=1}^n \left( x_l \sin \sqrt{|x_l|} \right), \quad (\text{A.4})$$

$$-500 \leq x_l \leq 500, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [420.9687 \cdots 420.9687]$ .

(E) Rastrigin function is

$$\min f(x) = \sum_{l=1}^n \left( x_l^2 - 10 \cos(2\pi x_l) + 10 \right), \quad (\text{A.5})$$

$$-5.12 \leq x_l \leq 5.12, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [0 \cdots 0]$ .



(F) Ackley's function is

$$\begin{aligned} \min f(x) = & -20e^{-0.2(\sqrt{(\sum_{l=1}^n x_l^n)/n})} - e^{(\sum_{l=1}^n \cos(2\pi x_l))/n} \\ & + 20 + e, \end{aligned} \quad (\text{A.6})$$

$$-32 \leq x_l \leq 32, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [0 \cdots 0]$ .

(G) Griewank function is

$$\begin{aligned} \min f(x) = & \frac{\sum_{l=1}^n x_l^2}{4000} - \prod_{l=1}^n \cos\left(\frac{x_l}{\sqrt{l}}\right) + 1, \end{aligned} \quad (\text{A.7})$$

$$-600 \leq x_l \leq 600, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [0 \cdots 0]$ .

(H) Rotate hyperellipsoid function is

$$\begin{aligned} \min f(x) = & \sum_{l=1}^n \left( \sum_{l=1}^l x_{ll} \right)^2, \end{aligned} \quad (\text{A.8})$$

$$-100 \leq x_l \leq 100, \quad \forall l = 1, 2, \dots, n,$$

where  $f(x^*) = 0$  with  $[x_1 \cdots x_n] = [0 \cdots 0]$ .

## Conflict of Interests

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

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (nos. 61102109, 61473309, and 61472443).

## References

- [1] D. Doorly, "Parallel genetic algorithms for optimization in CFD," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Periaux, M. Galan, and P. Cuesta, Eds., pp. 251–270, Wiley, Chichester, UK, 1995.
- [2] B. Peng and L. Li, "An improved localization algorithm based on genetic algorithm in wireless sensor networks," *Cognitive Neurodynamics*, vol. 9, no. 2, pp. 249–256, 2015.
- [3] D. V. Speranskii, "Ant colony optimization algorithms for digital device diagnostics," *Automatic Control and Computer Sciences*, vol. 49, no. 2, pp. 82–87, 2015.
- [4] K.-S. Yoo and S.-Y. Han, "Modified ant colony optimization for topology optimization of geometrically nonlinear structures," *International Journal of Precision Engineering and Manufacturing*, vol. 15, no. 4, pp. 679–687, 2014.
- [5] J. A. Tenreiro Machado, E. J. Solteiro Pires, and M. S. Couceiro, "Reply to: comments on 'particle swarm optimization with fractional-order velocity,'" *Nonlinear Dynamics*, vol. 77, no. 1, pp. 435–436, 2014.
- [6] V. Muthukumar, A. Suresh Babu, R. Venkatasamy, and N. Senthil Kumar, "An accelerated particle swarm optimization algorithm on parametric optimization of WEDM of die-steel," *Journal of The Institution of Engineers (India): Series C*, vol. 96, no. 1, pp. 49–56, 2015.
- [7] H.-C. Tsai, "Integrating artificial bee colony and bees algorithm for solving numerical function optimization," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 635–651, 2014.
- [8] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.
- [9] X. Xu and H.-L. Chen, "Adaptive computational chemotaxis based on field in bacterial foraging optimization," *Soft Computing*, vol. 18, no. 4, pp. 797–807, 2014.
- [10] S. Gholami-Boroujeny and M. Eshghi, "Active noise control using an adaptive bacterial foraging optimization algorithm," *Signal, Image and Video Processing*, vol. 8, no. 8, pp. 1507–1516, 2014.
- [11] A. Rezaee Jordehi, "A chaotic-based big bang–big crunch algorithm for solving global optimisation problems," *Neural Computing & Applications*, vol. 25, no. 6, pp. 1329–1335, 2014.
- [12] M. Sedighzadeh, M. Ghalambor, and A. Rezazadeh, "Reconfiguration of radial distribution systems with fuzzy multi-objective approach using modified big bang-big crunch algorithm," *Arabian Journal for Science and Engineering*, vol. 39, no. 8, pp. 6287–6296, 2014.
- [13] T. Dede, "Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures," *KSCE Journal of Civil Engineering*, vol. 18, no. 6, pp. 1759–1767, 2014.
- [14] H. S. Keesari and R. V. Rao, "Optimization of job shop scheduling problems using teaching-learning-based optimization algorithm," *Opsearch*, vol. 51, no. 4, pp. 545–561, 2013.
- [15] A. Tiwary, L. D. Arya, R. Arya, and S. C. Choube, "Inspection-repair based availability optimization of distribution systems using teaching learning based optimization," *Journal of The Institution of Engineers (India): Series B*, 2015.
- [16] Z. Zhang, L. Wang, and F. Long, "Immune optimization approach solving multi-objective chance-constrained programming," *Evolving Systems*, vol. 6, no. 1, pp. 41–53, 2015.
- [17] X. Xiao, T. Li, and R. Zhang, "An immune optimization based real-valued negative selection algorithm," *Applied Intelligence*, vol. 42, no. 2, pp. 289–302, 2015.
- [18] R. Shang, L. Jiao, Y. Ren, J. Wang, and Y. Li, "Immune clonal coevolutionary algorithm for dynamic multiobjective optimization," *Natural Computing*, vol. 13, no. 3, pp. 421–445, 2014.
- [19] P. K. Mahapatra, S. Ganguli, and A. Kumar, "A hybrid particle swarm optimization and artificial immune system algorithm for image enhancement," *Soft Computing*, vol. 19, no. 8, pp. 2101–2109, 2015.
- [20] J. Xiao, W. Li, B. Liu, and P. Ni, "A novel multi-population coevolution immune optimization algorithm," *Soft Computing*, 2015.
- [21] P. K. Mahapatra, S. Sethi, and A. Kumar, "Comparison of artificial immune system and particle swarm optimization techniques for error optimization of machine vision based tool movements," *Journal of The Institution of Engineers (India): Series C*, vol. 96, no. 4, pp. 363–372, 2015.
- [22] W. Shu, W. Wang, and Y. Wang, "A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, article 64, 2014.
- [23] R. Shang, H. Ma, J. Wang, L. Jiao, and R. Stolkin, "Immune clonal selection algorithm for capacitated arc routing problem," *Soft Computing*, 2015.

- [24] A.-M. Karimi-Majd, M. Fathian, and B. Amiri, "A hybrid artificial immune network for detecting communities in complex networks," *Computing*, vol. 97, no. 5, pp. 483–507, 2015.
- [25] R. J. Kuo, S. S. Chen, W. C. Cheng, and C. Y. Tsai, "Integration of artificial immune network and K-means for cluster analysis," *Knowledge and Information Systems*, vol. 40, no. 3, pp. 541–557, 2014.
- [26] D. Chen, J. Wang, F. Zou, W. Yuan, and W. Hou, "Time series prediction with improved neuro-endocrine model," *Neural Computing & Applications*, vol. 24, no. 6, pp. 1465–1475, 2014.
- [27] C. Rotar, "The Endocrine Control Evolutionary Algorithm: an extensible technique for optimization," *Natural Computing*, vol. 13, no. 1, pp. 97–117, 2014.
- [28] P. Jiang, L. Zhang, W. Zhu, H. Li, R. Dang, and M. Tang, "Chronic stress causes neuroendocrine-immune disturbances without affecting renal vitamin D metabolism in rats," *Journal of Endocrinological Investigation*, vol. 37, no. 11, pp. 1109–1116, 2014.
- [29] F. Cavalla, A. C. Araujo-Pires, C. C. Bigueti, and G. P. Garlet, "Cytokine networks regulating inflammation and immune defense in the oral cavity," *Current Oral Health Reports*, vol. 1, no. 2, pp. 104–113, 2014.
- [30] Z.-Z. Zhang, "Dynamic adaptive modular neural network architecture design," *Control and Decision*, vol. 29, no. 1, pp. 64–70, 2014.
- [31] D. Chen, F. Zou, J. Wang, and W. Yuan, "SAMCCTLBO: a multi-class cooperative teaching-learning-based optimization algorithm with simulated annealing," *Soft Computing*, pp. 1–23, 2015.
- [32] L. N. de Castro and F. J. von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, 2002.
- [33] C. K. Loo, W. S. Liew, M. Seera, and E. Lim, "Probabilistic ensemble Fuzzy ARTMAP optimization using hierarchical parallel genetic algorithms," *Neural Computing & Applications*, vol. 26, no. 2, pp. 263–276, 2015.
- [34] H. A. Pham, "Reduction of function evaluation in differential evolution using nearest neighbor comparison," *Vietnam Journal of Computer Science*, vol. 2, no. 2, pp. 121–131, 2015.
- [35] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [36] W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 459796, 16 pages, 2010.
- [37] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.



# Hindawi

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

