IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 20, 2019, accepted August 31, 2019, date of publication September 4, 2019,
date of current version September 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939353

A Multiple Diversity-Driven Brain Storm Optimization Algorithm With Adaptive Parameters

YANG YU[1], SHANGCE GAO[1], (Senior Member, IEEE), YIRUI WANG[1], ZHENYU LEI[1], JIUJUN CHENG[2], AND YUKI TODO[3], (Member, IEEE)

[1]Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan
[2]Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China
[3]Faculty of Electrical, Information, and Computer Engineering, Kanazawa University, Kanazawa 9201192, Japan

Corresponding authors: Shangce Gao (gaosc@eng.u-toyama.ac.jp) and Jiujun Cheng (chengjj@tongji.edu.cn)

This work was supported in part by the JSPS KAKENHI under Grant JP19K12136, and in part by the National Natural Science Foundation of China under Grant 61872271.

ABSTRACT Brain storm optimization (BSO) is a swarm intelligence optimization algorithm which is proven to have practical values in various fields. During these years, many modifications have been facilitated to effectively improve BSO's search performance. So far, these modifications focus on improving the solution quality by applying different clustering methods and learning strategies, in which the population diversity is often neglected. However, in recent studies, population diversity plays a more significant role in designing optimization algorithm. A population that maintains its diversity in a high level can easily obtain better solutions than the one with low level of diversity. Therefore, this paper proposes a control method that evaluates the population diversity of BSO to improve its performance. Two diversity measures, which are known as distance-based diversity and fitness-based diversity, are implemented to realize the adaptation of algorithm parameters. The new algorithm is called multiple diversity-driven BSO (MDBSO). Its performance is verified by CEC2017 benchmark function suit and a neuron model training task. The results demonstrate the effectiveness and efficiency of MDBSO.

INDEX TERMS Population diversity, parameter control, swarm intelligence and adaptive parameters.

I. INTRODUCTION

In recent years, various swarm intelligence (SI) algorithms have been proposed for solving diverse optimization problems. The main property of this kind of algorithms is that they mimic the social behaviors of nature creatures. As far as we know, it is full of wisdom and intelligence when animals are hunting, foraging and navigating in nature. Survival instincts drive them to improve search ability for creating more suitable living environment. Their behaviors gradually arouse great interests among researchers in the field of artificial intelligence [1]. Particle swarm optimization (PSO) which is one of the most popular SI algorithms is modeled based on the social behaviors of flocks of birds and schools of fish [2]. It supposes that a swarm of particles fly randomly in a multidimensional search space. Each of them represents a candidate solution for the optimization task. Their trajectories change according to the best position of the individual and the global best position of the whole population. Particles can effectively search for better solutions by taking advantage of this mechanism.

In addition to PSO, more and more SI optimization algorithms progressively spring into our view. Ant colony optimization (ACO) [3], fireworks algorithm (FA) [4], gravitational search algorithm (GSA) [5], artificial bee colony algorithm (ABC) [6] and brain storm optimization (BSO) [7] are some powerful optimization algorithms. These SI algorithms can be roughly divided into three categories according to the types of behaviors they take inspiration from.

The first category is called bio-inspired. Classical algorithms in this category such as ACO and ABC emulate the foraging behaviors of ant colony and bee colony, respectively. In ACO, individuals utilize a special chemical substance called pheromone to mark their search trajectory.

The associate editor coordinating the review of this manuscript and approving it for publication was Shahab Shamshirband.

The trajectory with more pheromone is considered as a preferred path to the global optimum, and further attracts other individuals [8]. ABC simulates the organizational structure of bees to categorize individuals into three groups: employed artificial bees, onlookers and scouts. The employed artificial bees represent candidate solutions and the onlookers are responsible for sharing the information of employed bees. After these steps, scouts are sent to diverse search area for discovering new solutions. This sophisticated idea of giving different functions to individuals makes the search procedure of ABC efficient and effective [6].

The second category can be named as physics-inspired. The algorithms belong to this category such as FA and GSA straightly take inspiration from physical phenomena or laws. For examples, the explosion processes of fireworks are utilized to design the search mechanism of FA, in which the distribution of individuals is analogized by the sparks in firework explosion. In GSA, the law of gravity is used to depict the relationship among individuals in search space. They are attracted by each other and the gravitational force is directly proportional to their fitness and inversely proportional to the square of the distance between them. The performance of GSA in different kinds of problems implies its powerful search ability [9]–[11].

The last category is called sociology-inspired. The major property of the algorithms in this category is that they are inspired by human social behaviors. BSO is very notable among SI algorithms and has already achieved great success in various applications [12]. Its operations of generating new individuals adopt the brainstorming process in human social behaviors. In reality, a group of people should be called together to figure out a solution when we encounter problems that can not be solved alone. This brainstorming process needs repetitive discussions and debates. BSO is enlightened by this feature and obtains an elaborated search process. At the rudimentary stage of optimization, individuals are divided into multiple clusters, then each cluster selects the best individual as the center. BSO has four independent individual generation methods and the selections of corresponding method are depending on three preset parameters p_1, p_2 and p_3. p_1 decides the usage of one or two clusters. In the condition of using one cluster, p_2 is adopted to choose the center or one random individual in the selected cluster. Otherwise, when two clusters are selected, p_3 determines the adoption of two centers or two random individuals. Being beneficial from this sophisticated selection mechanism, BSO can avoid sticking into local optima and outperform other optimization algorithms when dealing with multimodal problems [12]. However, the inherent feature of BSO that can not maintain good diversity reduces its robustness and deteriorates the performance of solving different problems. In the meanwhile, the parameter adjustment is very important in designing algorithms but it generally costs much time to find an acceptable parameter set. Therefore, more and more researchers prefer making parameters adaptive or self-adaptive to enhance the robustness and performance of algorithms [13]–[17].

Many modifications have been facilitated to improve the optimization performance of BSO but little work tries to make parameters be adaptive and keep the diversity staying in a high stage. BSO in objective space (BSO-OS) [18] aims to accelerate its convergent speed by replacing k-means clustering method with an elitist selection mechanism. Its mutation operation focuses on one-dimension objective space instead of the whole solution space. In [19], a random grouping BSO (RGBSO) is proposed to balance exploration and exploitation via adopting a new dynamic parameter in the generation of step-size. Besides, it replaces k-means clustering by a random grouping strategy so that the time complexity is decreased. Global-best BSO (GBSO) [20] tries to improve the performance of BSO from multiple aspects, including the clustering method, individual selection and mutation. Different from the k-means and mentioned random grouping methods, GBSO ranks the population according to their fitness and makes good and bad individuals equally distribute in different clusters. In original BSO, at most two individuals participate in generating new individuals, while in GBSO, more individuals can contribute to enhance the information exchange in this step. GBSO also adopts the global-best guidance strategy in PSO to modify its mutation mechanism. In our previous work [21], a chaotic local search method is combined with BSO (CBSO) to enhance its search ability and improve the solution quality. Besides the mentioned works, there are many other effective modifications for BSO, In [22], a self-adaptive multiobjective BSO (SMOBSO) is proposed. It adopts an adaptive mutation method to give an uneven distribution of solutions, but parameters still need to be set according to empirical data. Similarly, other works [23], [24] mainly focus on the adaptations of search step length in the mutation operator.

TABLE 1. The main parameters in BSO and MDBSO.

BSO	*Parameters*	n	p_0	p_1	p_2	p_3	k	μ	δ
	Values	5	0.2	0.8	0.4	0.5	20	1	0.5
MDBSO	*Parameters*	n	μ						
	Values	5	0.5						

Overall, most existing works that improve the performance of BSO focus on the adjustments of search and mutation strategies, but as we emphasized before, one drawback of BSO is that it has too many user-defined parameters. Presetting these parameters is a nontrivial task and generally difficult to find the best parameter set for solving different problems. Table 1 lists the main parameters of BSO and their corresponding values. It's widely accepted that the variation in parameter values of an algorithm could cause considerable fluctuation in performance [25]. Taking differential evolution (DE) as an example [26], the number of control parameters in DE is very few, including the scaling factor F, crossover rate CR and population size NP. The effects of these parameters on the performance of DE are well studied and it is reported that different value set for F and CR could obtain significant performance variations [27]. The most successful modifications for DE, such as JADE [13] and SHADE [28], employ parameter adaptation strategy to automatically update the control parameters. Besides, some researches indicate that diversity plays a significant role in improving search performance of SI algorithms [29]–[31].

In the design of optimization algorithms, the balance between exploration and exploitation is a crucial factor for the search performance. A good balance can make the algorithms fast converge and avoid local optimal solutions. Contrarily, the solution quality could be badly deteriorated when the relation is unbalanced. Therefore, the researches about keeping the balance between exploration and exploitation become crucial in recent years [32]–[34]. The key point of keeping balance is the preservation of population diversity in optimization process [31]. The population diversity can be explained as the extent of variation in the population based on the distribution or fitness performance obtained by individuals [35]. There are various methods that can be used to calculate the population diversity [30], [31]. The diversity is named as distance-based when it is measured according to the distance between each individual in decision space. While the fitness-based diversity is obtained by evaluating the performance of individuals in the objective space.

Both kinds of diversity have been incorporated into other techniques to improve the performance of corresponding algorithms. In [32], [33], the distance-based diversity is considered as an explicit objective. In other words, diversity and fitness are combined as a multi-objective problem to be solved. In this way, the balance between exploration and exploitation can be well maintained by searching for Pareto optimal solutions. The experimental results [32], [33] also demonstrate controlling diversity can evidently improve the performance of algorithm. With regard to the fitness-based diversity, it is mainly used to obtain good fitness spread among individual solutions. In [36], a variable relocation technique based on fitness diversity is applied to make the converged population restart convergence from another promising location. A fast adaptive memetic algorithm is proposed in [37] and the fitness diversity is investigated to control the utilization of local search strategies. Other techniques such as fitness sharing [38] and adaptive grid [39] also apply fitness diversity to improve the performance of algorithms [31]. Motivated by these prior studies, it can be expected to make the parameters adaptive via diversity control. Therefore, a multiple diversity-driven BSO (MDBSO) with well-balanced diversity and adaptive parameters is proposed.

The main contributions of this study can be summarized as: (1) We make the first attempt to use both distance-based diversity D_d and fitness-based diversity D_f to control the mutation process to the best of our knowledge. (2) Two new mutation strategies are adopted, including a local search strategy called BLX-α [40], [41] and a Gaussian mutation strategy. Additionally, D_f is utilized to adjust the standard deviation δ in Gaussian distribution. (3) Both D_d and D_f participate in generating new individuals. (4) Extensive experiments are conducted to verify the performance of MDBSO based on CEC2017 [42] benchmark function test suit and a neuron model training task [43], [44]. The results indicate that MDBSO has much better search ability than its peers.

The organization of this paper is arranged as follows. Section II briefly introduces the BSO. The proposed MDBSO is presented in Section III. In Section IV, the experimental results of benchmark function suit and neuron model training data set are reported to show the performance of MDBSO in comparison with other SI algorithms. Some discussions are given in Section V. Finally, we conclude this paper in Section VI.

Algorithm 1 Flowchart of BSO

```
Randomly generate a population with N individuals;
Calculate the fitness of each individual;
while maximum number of function evaluations is not reached do
    Use k-means to divide N individuals into n clusters;
    Choose the best individual in each cluster as the center;
    if random(0, 1) < p0 = 0.2 then
        replace one cluster center by a randomly generated individual
    end
    if random(0, 1) < p1 = 0.8 then
        select one cluster;
        if random(0, 1) < p2 = 0.4 then
            choose the cluster center as X_selected
        else
            randomly choose an individual in the cluster as X_selected
        end
    else
        randomly select two clusters;
        if random(0, 1) < p3 = 0.5 then
            choose the combination of two centers as X_selected
        else
            choose the combination of two randomly selected individuals in two clusters as X_selected
        end
    end
    Generate new individual by adding step length generated by Eqs. (1) and (2) to X_selected;
    if the new individual is better than the old one then
        replace the old individual
    end
end
```

II. BRAIN STORM OPTIMIZATION

BSO is an SI algorithm which mimics the human brainstorming in social behaviors. Algorithm 1 gives its optimization procedure. The main difference between BSO and other SI algorithms is that BSO divides the population into n clusters and the individual with the best fitness in each cluster is selected as the center. Then, $X_{selected}$ is selected to generate new individuals according to the process controlled by p_1, p_2 and p_3. If a random number is smaller than p_1 (= 0.8 [7]), one cluster will be selected. Otherwise, two clusters are applied to generate new individuals. In the condition of using one cluster, p_2 (= 0.4 [7]) decides the usage of the center or one random individual in the selected cluster. In addition, p_3 makes the centers and random individuals in two selected clusters have equal chance to participate in generating new individuals. Besides, BSO has another parameter p_0 to control the operation that replaces one cluster center by a randomly generated individual to avoid premature convergence. Finally, the population is updated based on the elite survival rule, i.e., the old individual will be replaced by the generated individual when the old one's fitness is worse. The mutation operator of BSO is shown as:

$$X_{generated} = X_{selected} + \xi \cdot N(0, 1) \tag{1}$$

where $X_{selected}$ and $X_{generated}$ are the selected and newly generated individuals, respectively. $N(0, 1)$ is the Gaussian distribution with mean 0 and variance 1. ξ is a search step length which is calculated by Eq. (2).

$$\xi = logsig((0.5 * iteration_{max} - t)/k) \cdot rand \tag{2}$$

where $logsig()$ means a logarithmic sigmoid transfer function, and its interval is $(0, 1)$. $iteration_{max}$ and t are the maximum iteration and current iteration count, respectively. k (= 20 [7]) is a scale factor to control the slope of $logsig()$ function.

III. MULTIPLE DIVERSITY-DRIVEN BSO (MDBSO)

A. DIVERSITY-DRIVEN STRATEGY

Although the distance-based diversity and fitness-based diversity are investigated in some researches, they are for the first time to be studied simultaneously as control parameters in this study. Before introducing the specific roles of two kinds of diversity in MDBSO, their formulas are given as follows.

$$D_d^j = \frac{1}{N_j}\sqrt{\sum_{i=1}^{N_j}(||X_i^j - X_{center}^j||)^2} \tag{3}$$

where j ($j = 1, 2, \ldots, n$) refers to the cluster number and D_d^j is the distance-based diversity of the jth cluster. N_j is the number of individuals in the jth cluster, X_i^j and X_{center}^j are the ith individual and the center in the current cluster, respectively.

The fitness-based diversity D_f is calculated as

$$D_f^j = \frac{1}{N_j}\sqrt{\sum_{i=1}^{N_j}(||F_i^j - F_{center}^j||)^2} \tag{4}$$

where F_i^j and F_{center}^j are the fitness of the ith individual and the center in the jth cluster, respectively. It should be noticed that we choose the centers and their fitness values instead of using the mean values as the subtrahends to calculate corresponding diversities. The intention is to increase convergence rate during optimization process as the centers are the best individuals in the population.

In MDBSO, D_d is applied as a control parameter to replace p_1, p_2 and p_3. It is adaptive via a normalization operation and the formula is shown in Eq. (5).

$$p_d^j(t) = \frac{D_d^j(t) - min\{D_d(t)\}}{max\{D_d(t)\} - min\{D_d(t)\}} \tag{5}$$

where $p_d^j(t)$ decides which mutation strategy is called to generate new individuals in the jth cluster at the tth iteration. $max(D_d(t))$ and $min(D_d(t))$ refer to the maximum and minimum values of D_d of n clusters at tth iteration, respectively. It's obvious that p_d^j values in the interval of [0, 1], and it controls a switch between two mutation strategies: BLX-α and Gaussian mutation. If a random value generated in (0, 1) is smaller than $p_d^j(t)$, it indicates the jth cluster may have a good distance diversity. Therefore, a local search method BLX-α is applied to speed up its convergence. Conversely, if the random value is greater than $p_d^j(t)$, the bad distance diversity in the jth cluster may eventually deteriorate solution quality and cause a premature convergence. Thus, the function of the Gaussian mutation is used to improve the distance diversity.

B. MUTATION STRATEGIES

1) BLX-α

BLX-α is a local search operator to adjust the population density [45]. Firstly, two individuals $X_1 = (x_1^1 \ldots x_1^{dim})$ and $X_2 = (x_2^1 \ldots x_2^{dim})$ are selected (dim is the dimension number). Then, a new individual is generated from the interval of $[min\{X_1, X_2\} - Y \times \alpha, max\{X_1, X_2\} + Y \times \alpha]$, where $Y = max\{X_1, X_2\} - min\{X_1, X_2\}$. α is a control parameter used to limit the search space. According to [41], BLX-α can increase the distribution of individuals when $\alpha > \frac{\sqrt{3}-1}{2}$, otherwise the distribution will be decreased. In particular, BLX-0 makes the variance of the distribution decrease and reduces the distance diversity. Therefore, we use BLX-0 in MDBSO because it is a local search operator that can improve solutions' quality when a cluster maintains a good distance diversity.

Particularly, the individuals X_1 and X_2 are selected via a novel mechanism, in which the centers of the top two clusters with the highest fitness diversity D_f are specified as X_1 and X_2, respectively. The reasons we use the fitness diversity instead of distance diversity here are as follows: (1) Even if the individuals have a close distance, their fitness can vary widely as they are in different peaks of a multi-modal problem. Thus, fitness diversity is more suitable for selections of mutation operators. (2) High fitness diversity means that the cluster manages good fitness spread among individual solutions. Thus, it can avoid premature convergence to a great extent. (3) Meanwhile, centers are the best individuals in the population. They are of strong reliability and promising to enable BLX-α to generate individuals with good fitness. The formula of generating individuals is shown in Eq. (6)

$$X_{generated} = rand \times (max\{X_1, X_2\} - min\{X_1, X_2\}) \quad (6)$$

where *rand* is a random value generated in (0, 1).

2) GAUSSIAN MUTATION

BLX-α is applied for the situation that the cluster stays in a good diversity. But it is not capable of improving diversity when the solution quality is poor. Therefore, we use a mutation strategy to generate new individuals when the distance diversity is relative low. In this part, the adopted Gaussian mutation is presented in details. A common formula which uses Gaussian distribution to generate new individuals can be described as follows.

$$X_{generated} = X_i + N(\mu, \delta) \cdot (X_{selected1} - X_{selected2}) \quad (7)$$

where X_i is the ith individual to be updated in the population. In MDBSO, $X_{selected1}$ and $X_{selected2}$ are two randomly selected individuals in the top two clusters with the highest distance diversity D_d, respectively. It should be pointed out that, different from the utilization of D_f in BLX-α, we use D_d because we want to increase the distance diversity here. Moreover, δ is adaptive in MDBSO and it is controlled by D_f. The adaptation mechanism is given as follows.

$$\delta = \frac{1}{e^{\omega}} \quad (8)$$

where e is the base of natural logarithm and ω is calculated according to Eq. (9).

$$\omega = \left|\frac{D_f^j(t) - mean\{D_f(t)\}}{max\{D_f(t)\} - min\{D_f(t)\}}\right| \quad (9)$$

where $mean\{D_f(t)\} = \frac{1}{n}\sum_{j=1}^{n} D_f^j(t)$.

It is worth emphasizing that we use $mean\{D_f(t)\}$ as the subtrahend to control ω. It is clear that ω and δ are negatively correlated, which means that the clusters with higher (or lower) D_f obtain smaller (or bigger) δ. Generally, an individual in the cluster with poor population diversity may need greater δ to provide a larger search step size in the aim of increase diversity. But too high diversity could cause the algorithm fail to converge. A contrast experiment is conducted, in which we use $min\{D_f(t)\}$ instead of $mean\{D_f(t)\}$ as the subtrahend. In this way, the cluster with the lowest fitness diversity would generate new individuals with $N(\mu, 1)$ and it's predictable that this method could obtain considerable population diversity due to the increase in δ. However, its optimization result is not as well as it of using $mean\{D_f(t)\}$. The reason is that too high population diversity undermines the performance of algorithm. Therefore, a moderate value is more suitable for not only maintaining population diversity, but also obtaining good results.

C. MDBSO

The structure of BSO is simplified by replacing its parameters with $p_d^j(t)$. We can find the number of parameters are substantially reduced due to the proposal of adaptive parameters, as shown in Table 1. The number of clusters stays the same as 5 in BSO and μ is set to 0.5. Regarding the values of these two parameters, some discussions are given in Section V.

Algorithm 2 Flowchart of MDBSO

```
Randomly generate a population with N individuals;
Calculate the fitness of each individual;
while maximum number of function evaluations is not reached do
    Use k-means to divide N individuals into n clusters;
    Choose the best individual in each cluster as the center;
    Calculate the D_d and D_f of each cluster;
    for the individual in the jth cluster do
        if random(0, 1) < p_d^j then
            use BLX-α strategy to generate new individuals in the jth cluster;
        else
            use Gaussian mutation strategy to generate new individuals in the jth cluster;
        end
    end
    if the new individual is better than the old one then
        replace the old individual
    end
end
```

The primary procedures of MDBSO is presented in Algorithm 2. In the first step, MDBSO randomly generates N individuals and calculates their fitness. If the termination is not satisfied, k-means is applied to divide the population into n clusters. The best individual in each cluster is selected as the *center*. Then, MDBSO has its specific step in which the distance diversity D_d and fitness diversity D_f of each cluster are calculated. The p_d^j calculated by D_d decides the selection of mutation strategies for each cluster. The $BLX-\alpha$ strategy would have a high possibility to be applied to generate individuals in the cluster with good distance diversity. In the opposite, the Gaussian mutation strategy is utilized by the cluster with bad distance diversity. The new generated individual with better fitness will replace the old at the end of each iteration.

Fig. 1 illustrates the functions of D_d and D_f in the specific steps in MDBSO. Each of them has very important role in the generation of new solutions and would be used for more than once. In most literature, population diversity is usually

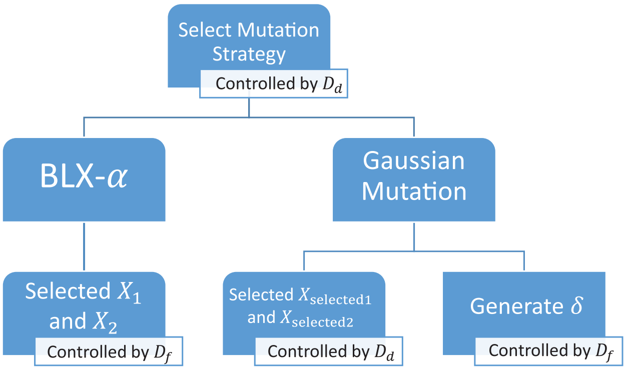


FIGURE 1. The functions of distance diversity and fitness diversity in MDBSO.

applied as an optimization objective rather than an approach. They focus on the maintenance of population diversity but it does not participate in the search process. Innovatively, in this study, the distance and fitness diversity are simultaneously utilized and have been proven to be very effective in enhancing the performance of BSO.

Compared with the original BSO presented in Algorithm 1, MDBSO has essential modifications in two aspects. One is that the diversity in BSO is well maintained so that the search ability is enhanced. The other is the adaptations of parameters. Most steps in MDBSO are controlled by adaptive parameters, which enhances its robustness and makes it can be applied into more diverse application scenarios.

IV. EXPERIMENTAL RESULTS

A. BENCHMARK FUNCTION TEST SUIT

In this section, CEC2017 benchmark function suit is implemented to test the performance of MDBSO. It should be noticed that F2 in CEC2017 has been excluded because it shows unstable behavior especially for higher dimensions, and significant performance variations for the same algorithm implemented in Matlab and C [42]. This benchmark function suit includes 2 unimodal, 7 simple unimodal, 10 hybrid and 10 composition functions. Hence, it is very suitable for testing the search ability and robustness of optimization algorithms. The population size N is 100, and the dimension for the problems dim is 30. Each problem is run for 30 times to reduce random errors. The maximum number of function evaluations (MFE) is 10000*dim. All experiments are implemented on a PC with 3.10GHz Intel(R) Core(TM) i5-4440 CPU and 8GB of RAM using MATLAB R2013b. All parameters for the contrast SI algorithms are set up according to the values provided in the corresponding literature.

1) MDBSO *VS*. BSO VARIANTS

In this part, MDBSO is compared with BSO [7] and its variants, including CBSO [21], BSO-OS [18], RGBSO [19], GBSO [20] and ASBSO [23]. The results including mean and standard deviation (Std Dev) are listed in Tables 2 and 3. The values in boldface represent the best results among compared algorithms.

TABLE 2. Experimental results of MDBSO versus BSO variants on CEC'17 benchmark functions (1).

Fun.	MDBSO Mean ±Std Dev	BSO Mean ±Std Dev	CBSO Mean ±Std Dev	BSO-OS Mean ±Std Dev
F1	3.17E+03 ± 4.68E+03	2.47E+03 ± 1.95E+03 ≈	3.99E+03 ± 3.07E+03 +	**1.96E+03 ± 1.57E+03** ≈
F3	6.13E+02 ± 6.28E+02	5.34E+02 ± 2.66E+02 ≈	**3.03E+02 ± 3.56E+00** −	8.78E+03 ± 3.04E+03 +
F4	4.62E+02 ± 3.28E+01	4.67E+02 ± 2.19E+01 ≈	4.94E+02 ± 2.04E+01 +	4.66E+02 ± 2.29E+01 ≈
F5	**6.02E+02 ± 3.10E+01**	6.87E+02 ± 4.05E+01 +	6.98E+02 ± 3.91E+01 +	6.82E+02 ± 2.88E+01 +
F6	6.13E+02 ± 7.18E+00	6.52E+02 ± 7.01E+00 +	6.47E+02 ± 7.67E+00 +	6.50E+02 ± 5.86E+00 +
F7	8.88E+02 ± 6.43E+01	1.15E+03 ± 9.65E+01 +	1.13E+03 ± 8.76E+01 +	1.12E+03 ± 6.90E+01 +
F8	9.13E+02 ± 3.75E+01	9.47E+02 ± 2.82E+01 +	9.41E+02 ± 2.57E+01 +	9.37E+02 ± 2.95E+01 +
F9	1.31E+03 ± 5.38E+02	3.98E+03 ± 6.95E+02 +	3.87E+03 ± 6.65E+02 +	3.74E+03 ± 4.78E+02 +
F10	7.37E+03 ± 1.29E+03	5.30E+03 ± 5.16E+02 −	5.37E+03 ± 5.82E+02 −	5.20E+03 ± 8.49E+02 −
F11	**1.21E+03 ± 5.06E+01**	1.23E+03 ± 4.05E+01 ≈	1.23E+03 ± 4.31E+01 ≈	1.23E+03 ± 4.18E+01 ≈
F12	**4.39E+04 ± 2.26E+04**	1.77E+06 ± 1.25E+06 +	1.91E+06 ± 1.36E+06 +	1.88E+06 ± 1.30E+06 +
F13	**1.46E+04 ± 1.61E+04**	5.36E+04 ± 2.85E+04 +	5.82E+04 ± 4.12E+04 +	5.84E+04 ± 3.64E+04 +
F14	7.71E+03 ± 5.29E+03	6.40E+03 ± 4.66E+03 ≈	3.51E+03 ± 2.20E+03 −	9.92E+03 ± 8.88E+03 ≈
F15	**7.51E+03 ± 8.01E+03**	2.95E+04 ± 1.61E+04 +	2.97E+04 ± 1.67E+04 +	3.23E+04 ± 1.81E+04 +
F16	**2.44E+03 ± 4.43E+02**	3.20E+03 ± 4.24E+02 +	2.83E+03 ± 2.73E+02 +	3.10E+03 ± 2.70E+02 +
F17	**1.98E+03 ± 1.94E+02**	2.48E+03 ± 2.56E+02 +	2.20E+03 ± 2.10E+02 +	2.42E+03 ± 2.96E+02 +
F18	9.28E+04 ± 5.38E+04	1.21E+05 ± 1.03E+05 ≈	8.35E+04 ± 4.27E+04 ≈	1.51E+05 ± 1.21E+05 ≈
F19	9.70E+03 ± 9.19E+03	1.52E+05 ± 6.43E+04 +	9.59E+04 ± 5.88E+04 +	1.15E+05 ± 4.55E+04 +
F20	**2.20E+03 ± 1.19E+02**	2.67E+03 ± 1.74E+02 +	2.50E+03 ± 1.33E+02 +	2.72E+03 ± 2.10E+02 +
F21	**2.40E+03 ± 4.16E+01**	2.50E+03 ± 4.48E+01 +	2.48E+03 ± 4.08E+01 +	2.51E+03 ± 3.67E+01 +
F22	4.78E+03 ± 3.01E+03	6.03E+03 ± 1.77E+03 ≈	5.36E+03 ± 2.21E+03 ≈	6.42E+03 ± 1.54E+03 ≈
F23	**2.73E+03 ± 2.30E+01**	3.29E+03 ± 1.27E+02 +	3.02E+03 ± 1.27E+02 +	3.31E+03 ± 9.91E+01 +
F24	**2.92E+03 ± 5.27E+01**	3.50E+03 ± 1.13E+02 +	3.13E+03 ± 1.42E+02 +	3.47E+03 ± 2.09E+02 +
F25	2.89E+03 ± 1.20E+01	2.89E+03 ± 1.46E+01 −	2.89E+03 ± 6.78E+00 −	**2.88E+03 ± 8.40E+00** −
F26	**4.78E+03 ± 6.31E+02**	8.16E+03 ± 1.57E+03 +	6.26E+03 ± 2.12E+03 +	7.65E+03 ± 1.89E+03 +
F27	**3.23E+03 ± 2.01E+01**	3.82E+03 ± 2.91E+02 +	3.39E+03 ± 1.94E+02 +	3.86E+03 ± 2.64E+02 +
F28	3.19E+03 ± 5.78E+01	3.21E+03 ± 2.57E+01 ≈	3.19E+03 ± 4.02E+01 ≈	3.21E+03 ± 1.35E+01 ≈
F29	**3.70E+03 ± 1.92E+02**	4.38E+03 ± 2.79E+02 +	4.24E+03 ± 2.85E+02 +	4.37E+03 ± 2.71E+02 +
F30	**8.49E+03 ± 3.24E+03**	5.74E+05 ± 3.50E+05 +	3.91E+05 ± 2.07E+05 +	7.73E+05 ± 4.79E+05 +
Rank	1	6	4	7
+/ ≈ /−	- /- /-	19/ 8 /2	21/ 4/4	20/ 7 /2

TABLE 3. Experimental results of MDBSO versus BSO variants on CEC'17 benchmark functions (2).

Fun.	RGBSO Mean ±Std Dev	GBSO Mean ±Std Dev	ASBSO Mean ±Std Dev
F1	2.50E+03 ± 2.90E+03 ≈	3.43E+03 ± 3.60E+03 ≈	2.21E+03 ± 2.00E+03 ≈
F3	2.39E+04 ± 7.78E+03 +	**3.00E+02± 2.11E-04** −	3.95E+02 ± 1.10E+02 ≈
F4	4.75E+02 ± 1.10E+01 +	**4.58E+02± 3.13E+01** ≈	4.72E+02 ± 2.92E+01 ≈
F5	6.37E+02 ± 2.49E+01 +	6.79E+02± 3.44E+01 +	6.86E+02 ± 3.45E+01 +
F6	**6.01E+02 ± 8.14E-01** −	6.43E+02± 8.74E+00 +	6.51E+02 ± 7.77E+00 +
F7	9.10E+02 ± 3.11E+01 +	**8.36E+02± 2.89E+01** −	1.16E+03 ± 9.94E+01 +
F8	**9.08E+02 ± 1.89E+01** ≈	9.31E+02 ±2.62E+01 +	9.41E+02 ± 3.19E+01 +
F9	2.35E+03 ± 5.33E+02 +	**1.08E+03± 2.56E+02** −	3.93E+03 ± 6.39E+02 +
F10	**4.22E+03 ± 5.39E+02** −	5.00E+03± 6.55E+02 −	5.20E+03 ± 5.67E+02 −
F11	1.22E+03 ± 4.44E+01 ≈	1.25E+03 ±4.45E+01 +	1.23E+03 ± 4.75E+01 ≈
F12	1.73E+06 ± 1.06E+06 +	2.52E+06± 1.66E+06 +	1.41E+06 ± 8.00E+05 +
F13	1.76E+04 ± 1.66E+04 ≈	7.94E+04 ±4.60E+04 +	5.04E+04 ± 2.64E+04 +
F14	1.55E+04 ± 1.65E+04 +	**1.92E+03 ±4.01E+02** −	7.08E+03 ± 5.23E+03 ≈
F15	4.57E+03 ± 5.42E+03 −	5.61E+04 ±4.76E+04 +	3.01E+04 ± 2.25E+04 +
F16	2.93E+03 ± 3.46E+02 +	2.87E+03 ±1.95E+02 +	3.01E+03 ± 2.25E+02 +
F17	2.40E+03 ± 2.67E+02 +	2.22E+03± 1.99E+02 +	2.40E+03 ± 2.44E+02 +
F18	1.43E+05 ± 8.64E+04 +	**6.82E+04 ±3.51E+04** −	1.23E+05 ± 1.21E+05 ≈
F19	**6.17E+03 ± 4.20E+03** ≈	1.01E+05± 6.16E+04 +	1.25E+05 ± 6.31E+04 +
F20	2.58E+03 ± 2.44E+02 +	2.68E+03± 2.11E+02 +	2.67E+03 ± 2.17E+02 +
F21	2.45E+03 ± 3.67E+01 +	2.48E+03 ±4.44E+01 +	2.49E+03 ± 3.14E+01 +
F22	**3.54E+03 ± 1.82E+03** ≈	3.72E+03± 2.26E+03 ≈	5.79E+03 ± 2.04E+03 ≈
F23	2.90E+03 ± 8.97E+01 +	3.01E+03± 9.52E+01 +	3.26E+03 ± 1.24E+02 +
F24	3.32E+03 ± 1.27E+02 +	3.17E+03 ±1.14E+02 +	3.49E+03 ± 9.56E+01 +
F25	2.89E+03 ± 1.76E+01 −	2.90E+03± 2.25E+01 ≈	2.89E+03 ± 1.25E+01 −
F26	5.47E+03 ± 1.89E+03 +	6.10E+03 ±1.61E+03 +	7.84E+03 ± 1.80E+03 +
F27	3.29E+03 ± 3.33E+01 +	3.24E+03 ±6.39E+01 ≈	3.85E+03 ± 2.17E+02 +
F28	3.23E+03 ± 2.24E+01 +	3.21E+03± 3.68E+01 ≈	**3.18E+03 ± 3.60E+01** ≈
F29	3.90E+03 ± 2.39E+02 +	4.27E+03 ±2.85E+02 +	4.40E+03 ± 3.39E+02 +
F30	5.06E+04 ± 4.62E+04 +	7.66E+05 ±4.05E+05 +	5.16E+05 ± 2.90E+05 +
Rank	2	3	5
+/ ≈ /−	19 /6 /4	17/ 6 /6	19/ 8/2

We can intuitively find that MDBSO obtains much more number of the best results in comparison with other competitors from these tables. It should be emphasized that MDBSO outperforms BSO on hybrid and composition functions (F11-F30) except for F14, suggesting that the drawback of BSO's poor robustness is mitigated and the search ability of BSO is greatly improved via diversity controlled parameters. A non-parametric statistical analysis called Friedman test is employed to give the ranking that each algorithm obtained in the current comparison [46]. The lower ranking indicates the better performance. As observed, MDBSO is the algorithm with the best performance among the compared BSO variations. To more precisely analyze its performance, a non-parametric statistical test called Wilcoxon rank-sum test is implemented [47]. Each $+/\approx/-$ indicates the performance of MDBSO is significantly better (+), not significantly better and worse (≈) or worse (−) than its peers. According to the statistical results, the number of times MDBSO wins to others is 19 (BSO), 21 (CBSO), 20 (BSO-OS), 19 (RGBSO), 17 (GBSO) and 19 (ASBSO) out of 29 tested problems, respectively. Moreover, there are at most six problems where MDBSO underperforms another algorithm (i.e. GBSO). Considering the tested algorithms are state-of-the-art BSO variations, MDBSO has verified its superior and it executes an effective search process by the adaptive parameter system.

In addition, box-and-whisker diagrams and convergence graphs are given in Fig. 2 and Fig. 3 to directly exhibit the difference in performance between MDBSO and its peers, respectively. The box-and-whisker diagrams can illustrate the quality of solutions on 30 runs. There are five values are conventionally used: the extremes, the upper and lower hinges (quartiles), and the median. The interval between the upper and lower hinges of the box is called interquartile range (IQR) and it indicates the degree of dispersion and skewness in the results. Symbol + indicates the outliers. As observed in Fig. 2, MDBSO obtains the best performance on F5, F12, F13, F23, F24, and F26. Fig. 3 depicts the convergent performance during the whole search procedure. The horizontal axis represents the number of function evaluations, and the vertical axis denotes the average values of optimization results on 30 runs. It's obvious that the convergence speed of MDBSO is much faster than its peers. In addition, it generally obtains better results. Although RGBSO shows an ability of avoiding premature on F13, F23, and F26, its slow convergent speed deteriorates the solutions' quality. The good performance of MDBSO profits from the diversity controlled search mechanism which keeps the balance between exploration and exploitation.

Moreover, to show the population diversity obtained by each contrast algorithm graphically, four plots on F12, F13, F23, and F24 are illustrated in Fig. 4, respectively.

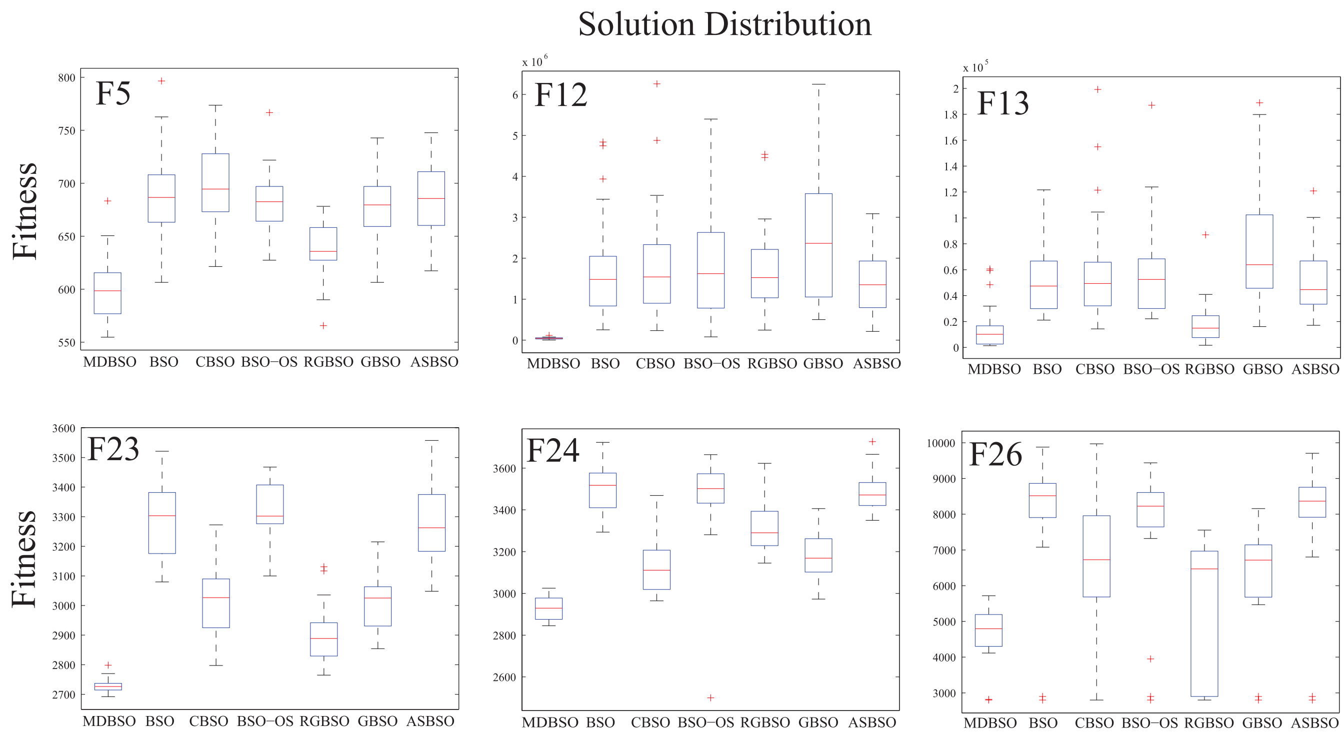


FIGURE 2. The box-and-whisker diagrams of optimal solutions obtained by seven kinds of BSOs on F5, F12, F13, F23, F24, F26.

The calculation of population diversity is shown in Eq. (10)

$$Div = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(||X_i - X_{mean}||)^2} \tag{10}$$

where Div is the population diversity. N is population size and X_i is the ith individual. X_{mean} which is calculated as $X_{mean} = \frac{1}{N}\sum_{i=1}^{N} X_i$ is the average of the population.

As observed, MDBSO and CBSO are the best two algorithms that maintain population diversity at a good level in the whole process. This is owing to that MDBSO implements the diversity-driven strategy and CBSO uses a chaotic local search mechanism that can disturb the search trajectory of the individual. The diversity of GBSO keeps stable at the early stage but rapidly deteriorates, which makes it lose the capability of further improving solutions' quality. It should be emphasized that the diversity of RGBSO keeps fluctuating, which means that the dynamic step-size parameter control strategy has the efficacy of improving population diversity, but RGBSO lacks a mechanism to maintain it. Based on these analyses, one conclusion can be drawn that MDBSO can significantly preserve the population diversity in a good level during the search process.

2) MDBSO *VS*. GSA VARIANTS

Besides the BSO variants, more SI algorithm are applied to further testify the effectiveness of MDBSO. GSA has been proposed for nearly a decade, and its developments are more matured than BSO's. Thus, the comparison between MDBSO and GSA variants can reflect the position of MDBSO in the whole SI algorithms. GSA uses gravity to mimic the search mechanism of individuals and it has obtained many successes in various research aspects. In this part, GSA [5] and its variants (IGSA [14], GGSA [48], MGSA [49], PSOGSA [50], HGSA [51], DNLGSA [52]) are implemented and their experimental results are presented in Tables 4 and 5. The results including mean and standard deviation (Std Dev) are exhibited and the values in boldface represent the best results among compared algorithms. The *Rank* refers to the rank of each algorithm obtained in the Friedman test. Each $+/\approx/-$ indicates the performance of MDBSO is significantly better ($+$), not significantly better and worse ($\approx$) or worse ($-$) than its peers.

In the statistical results obtained by MDBSO and GSA variants, the number of times MDBSO wins to others is 26 (GSA), 17 (IGSA), 18 (GGSA), 20 (MGSA), 25 (PSOGSA), 13 (HGSA) and 25 (DNLGSA) out of 29 problems, respectively. MDBSO can significantly outperform most variations of GSA and is very competitive with HGSA. This indicates that MDBSO obtains a strong competitiveness in comparison with GSA and its peers.

3) MDBSO *VS*. ABC VARIANTS

ABC [6] is another powerful SI algorithm and it has been successfully modified in these years. It has a very special search mechanism against classical SI algorithms like PSO and GSA. The population in ABC is divided into three categories: employed bees, onlooker bees and scout bees.

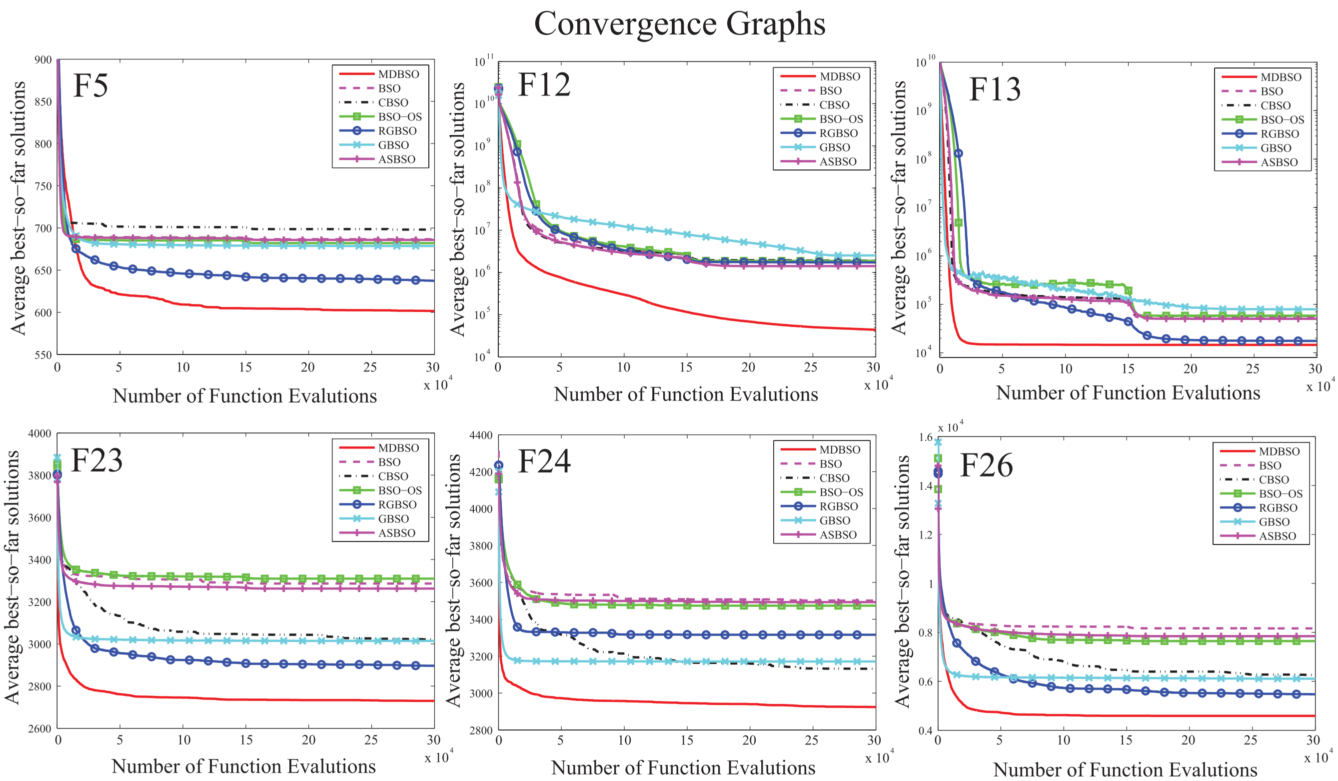


FIGURE 3. The convergence graphs of average best-so-far solutions obtained by seven kinds of BSOs on F5, F12, F13, F23, F24, F26.

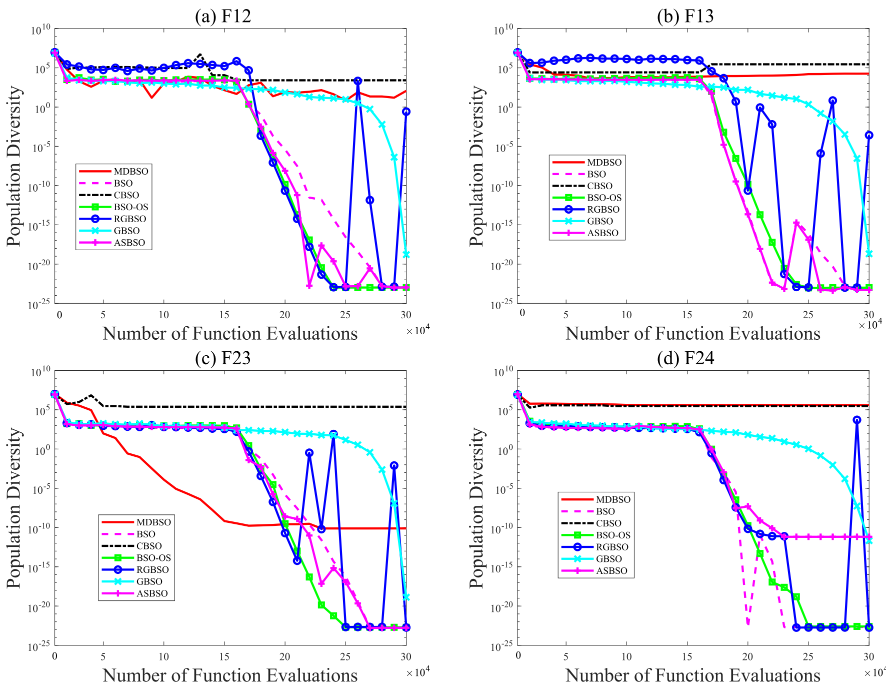


FIGURE 4. Population diversity on F12, F13, F23, and F24.

Each of them has different responsibility during the search process. Therefore, ABC is quite successful in optimizing multivariable and multimodal problems. As we mentioned that the proposed MDBSO is superior than BSO in solving such kinds of problems [53]. Using ABC variants as contrasts, including GABC [54], SeABC [55], MABC [56], RABC [57]

TABLE 4. Experimental results of MDBSO versus GSA variants on CEC'17 benchmark functions (1).

Fun.	MDBSO Mean ±Std Dev	GSA Mean ±Std Dev	IGSA Mean ±Std Dev	GGSA Mean ±Std Dev
F1	3.17E+03 ± 4.68E+03	2.00E+03 ± 1.03E+03 ≈	**1.88E+03 ± 1.37E+03** ≈	2.18E+03 ± 1.12E+03 ≈
F3	**6.13E+02 ± 6.28E+02**	8.30E+04 ± 4.33E+03 +	6.04E+04 ± 7.02E+03 +	6.02E+04 ± 6.73E+03 +
F4	**4.62E+02 ± 3.28E+01**	5.42E+02 ± 1.59E+01 +	5.22E+02 ± 2.10E+01 +	5.33E+02 ± 2.30E+01 +
F5	6.02E+02 ± 3.10E+01	7.26E+02 ± 2.01E+01 +	**5.42E+02 ± 8.18E+00** −	6.11E+02 ± 1.22E+01 +
F6	6.13E+02 ± 7.18E+00	6.50E+02 ± 2.75E+00 +	**6.00E+02 ± 1.29E-02** −	6.09E+02 ± 5.29E+00 −
F7	8.88E+02 ± 6.43E+01	7.87E+02 ± 1.19E+01 −	**7.43E+02 ± 5.60E+00** −	7.37E+02 ± 1.49E+00 −
F8	9.13E+02 ± 3.75E+01	9.51E+02 ± 1.31E+01 +	**8.33E+02 ± 7.72E+00** −	8.88E+02 ± 9.79E+00 −
F9	1.31E+03 ± 5.38E+02	2.93E+03 ± 3.92E+02 +	9.00E+02 ± 2.11E-14 −	**9.00E+02 ± 0.00E+00** −
F10	7.37E+03 ± 1.29E+03	4.87E+03 ± 4.34E+02 −	**3.58E+03 ± 4.60E+02** −	4.38E+03 ± 3.89E+02 −
F11	1.21E+03 ± 5.06E+01	1.45E+03 ± 8.92E+01 +	1.28E+03 ± 7.44E+01 +	1.25E+03 ± 3.23E+01 +
F12	**4.39E+04 ± 2.26E+04**	1.03E+07 ± 1.93E+07 +	1.40E+06 ± 7.32E+05 +	4.83E+05 ± 2.11E+05 +
F13	1.46E+04 ± 1.61E+04	3.10E+04 ± 6.45E+03 +	3.06E+04 ± 7.97E+03 +	1.87E+04 ± 4.70E+03 +
F14	7.71E+03 ± 5.29E+03	4.74E+05 ± 1.31E+05 +	1.96E+05 ± 1.37E+05 +	1.96E+05 ± 7.59E+04 +
F15	7.51E+03 ± 8.01E+03	1.17E+04 ± 1.93E+03 +	1.31E+04 ± 3.65E+03 +	4.12E+03 ± 1.57E+03 ≈
F16	**2.44E+03 ± 4.43E+02**	3.18E+03 ± 2.84E+02 +	2.71E+03 ± 2.16E+02 +	2.88E+03 ± 3.22E+02 +
F17	**1.98E+03 ± 1.94E+02**	2.90E+03 ± 1.70E+02 +	2.22E+03 ± 2.14E+02 +	2.67E+03 ± 2.06E+02 +
F18	9.28E+04 ± 5.38E+04	3.20E+05 ± 1.76E+05 +	3.81E+05 ± 3.85E+05 +	1.68E+05 ± 7.28E+04 +
F19	9.70E+03 ± 9.19E+03	1.42E+04 ± 5.13E+03 +	1.57E+04 ± 8.10E+03 +	5.93E+03 ± 1.46E+03 ≈
F20	**2.20E+03 ± 1.19E+02**	3.03E+03 ± 2.36E+02 +	2.41E+03 ± 1.70E+02 +	2.82E+03 ± 1.64E+02 +
F21	2.40E+03 ± 4.16E+01	2.56E+03 ± 1.95E+01 +	**2.35E+03 ± 6.54E+00** −	2.41E+03 ± 2.11E+01 +
F22	4.78E+03 ± 3.01E+03	6.39E+03 ± 1.69E+03 +	**2.30E+03 ± 0.00E+00** −	2.30E+03 ± 2.05E-10 -
F23	**2.73E+03 ± 2.30E+01**	3.56E+03 ± 1.23E+02 +	2.74E+03 ± 2.26E+01 ≈	2.86E+03 ± 3.94E+01 +
F24	2.92E+03 ± 5.27E+01	3.29E+03 ± 5.57E+01 +	**2.82E+03 ± 2.24E+01** −	2.91E+03 ± 3.70E+01 ≈
F25	2.89E+03 ± 1.20E+01	2.93E+03 ± 1.22E+01 +	2.92E+03 ± 9.79E+00 +	2.93E+03 ± 1.03E+01 +
F26	4.78E+03 ± 6.31E+02	6.86E+03 ± 8.95E+02 +	**2.83E+03 ± 4.66E+01** −	2.94E+03 ± 5.28E+02 −
F27	**3.23E+03 ± 2.01E+01**	4.67E+03 ± 3.21E+02 +	3.37E+03 ± 6.87E+01 +	3.39E+03 ± 3.57E+01 +
F28	3.19E+03 ± 5.78E+01	3.31E+03 ± 4.94E+01 +	3.26E+03 ± 3.48E+01 +	3.23E+03 ± 3.28E+01 +
F29	**3.70E+03 ± 1.92E+02**	4.71E+03 ± 2.10E+02 +	4.03E+03 ± 2.22E+02 +	4.25E+03 ± 2.30E+02 +
F30	**8.49E+03 ± 3.24E+03**	1.70E+05 ± 1.24E+05 +	3.34E+05 ± 3.68E+05 +	4.39E+04 ± 1.91E+04 +
Rank	1	6	3	4
+/ ≈ /−	- /- /-	26/ 1 /2	17/ 2/10	18/ 4 /7

and SFABC [58], is a very suitable conduct to prove the promising search ability of MDBSO.

Tables 6 and 7 exhibit the results obtained by MDBSO and ABC variants. The results including mean and standard deviation (Std Dev) are exhibited and the values in boldface represent the best results among compared algorithms. The *Rank* refers to the rank of each algorithm obtained in the Friedman test. Each $+/\approx/-$ indicates the performance of MDBSO is significantly better ($+$), not significantly better and worse ($\approx$) or worse ($-$) than its peers. To be precise, MDBSO significantly outperforms ABC and its variants on 26 (ABC), 17 (GABC), 25 (MABC), 24 (SeABC), 21 (RABC) and 20 (SFABC) problems, respectively. The result reveals the fact that MDBSO has better performance in solving diverse optimization problems than most state-of-the-art variants of ABC.

B. ARTIFICIAL NEURON NETWORK (ANN) TRAINING DATA SET

Benchmark functions are widely used to test the preliminary performance of the proposed algorithms because of its simple practicality. They can directly exhibit the pros and cons of the tested algorithms. However, it is far from enough to only use benchmark functions as they are surrogate models and can not closely reflect real-world challenges to cross the big gap between academia and industries. Thus, in this section, we make an attempt to apply MDBSO for training a dendritic neuron model (DNM) [43].

DNM is proposed by considering the nonlinearity of synapses and has achieved great success in classification and prediction problems [59]–[62]. It is composed of four layers, including a synaptic layer, a dendrite layer, a membrane layer and a soma layer. Gao *et al*. [44] conclude that DNM with learning algorithms can outperform the traditional multilayer perceptron (MLP) model with the same algorithms. Besides, training a neural network is a complex and tough optimization problem as it requires high computational cost. The goal is to minimize the sum of errors (between the practical and desired values) by optimizing the weight ω and threshold θ [63]. Thus, the application of using MDBSO can closely reflect its practical value in real-world challenges. We use BSO and MDBSO to train DNM for classification, approximation and prediction problems to systemically investigate the effectiveness of MDBSO in ANN training.

Four classification, three function approximation, and three prediction problems are used to testify the effectiveness of MDBSO for training DNM. The classification problems, i.e., XOR, ballon, iris and heart are acquired from the University of California at Irvine Machine Learning Repository [64]. The number of attributes, training samples, test samples and classes for these problems are summarized in Table 8. The function approximation problems include 1-D cosine with

TABLE 5. Experimental results of MDBSO versus GSA variants on CEC'17 benchmark functions (2).

Fun.	MGSA Mean ±Std Dev	PSOGSA Mean ±Std Dev	HGSA Mean ±Std Dev	DNLGSA Mean ±Std Dev
F1	4.63E+03 ± 4.30E+03 +	4.12E+03 ± 3.26E+03 +	2.68E+03 ± 2.50E+03 ≈	1.22E+05 ± 1.81E+05 +
F3	4.23E+04 ± 1.28E+04 +	3.56E+03 ± 7.87E+03 +	4.36E+04 ± 5.49E+03 +	1.49E+04 ± 1.24E+04 +
F4	5.33E+02 ± 5.86E+01 +	1.04E+03 ± 5.05E+02 +	5.19E+02 ± 2.63E+00 +	7.11E+02 ± 1.46E+02 +
F5	6.35E+02 ± 3.01E+01 +	6.46E+02 ± 3.40E+01 +	6.53E+02 ± 1.28E+01 +	6.50E+02 ± 3.67E+01 +
F6	6.27E+02 ± 8.09E+00 +	6.24E+02 ± 8.94E+00 +	6.08E+02 ± 4.54E+00 −	6.41E+02 ± 7.86E+00 +
F7	8.38E+02 ± 2.65E+01 −	9.72E+02 ± 6.32E+01 +	7.41E+02 ± 3.01E+00 −	9.86E+02 ± 6.78E+01 +
F8	9.08E+02 ± 2.29E+01 ≈	9.36E+02 ± 3.25E+01 +	9.00E+02 ± 9.03E+00 ≈	9.16E+02 ± 2.85E+01 ≈
F9	3.41E+03 ± 8.50E+02 +	4.54E+03 ± 1.67E+03 +	9.00E+02 ± 9.67E-14 −	3.93E+03 ± 1.10E+03 +
F10	4.92E+03 ± 8.11E+02 −	4.70E+03 ± 6.23E+02 −	4.21E+03 ± 2.93E+02 −	4.96E+03 ± 8.84E+02 −
F11	1.23E+03 ± 4.53E+01 ≈	1.49E+03 ± 3.14E+02 +	**1.20E+03 ± 2.98E+01** ≈	1.51E+03 ± 2.44E+02 +
F12	5.27E+05 ± 5.78E+05 +	6.00E+07 ± 1.48E+08 +	1.29E+05 ± 8.15E+04 +	1.58E+08 ± 2.63E+08 +
F13	2.81E+05 ± 1.42E+06 +	2.39E+07 ± 7.46E+07 +	**1.46E+04 ± 5.32E+03 +**	1.62E+06 ± 8.72E+06 +
F14	1.87E+04 ± 3.80E+04 +	9.87E+04 ± 2.69E+05 ≈	**6.72E+03 ± 3.05E+03** ≈	6.07E+04 ± 1.02E+05 +
F15	6.08E+03 ± 4.72E+03 ≈	5.31E+05 ± 2.82E+06 +	**2.20E+03 ± 7.21E+02** −	1.29E+04 ± 1.02E+04 +
F16	2.83E+03 ± 2.89E+02 +	3.05E+03 ± 4.59E+02 +	2.83E+03 ± 2.32E+02 +	2.74E+03 ± 3.13E+02 +
F17	2.37E+03 ± 2.07E+02 +	2.27E+03 ± 2.29E+02 +	2.77E+03 ± 1.99E+02 +	2.30E+03 ± 2.31E+02 +
F18	1.44E+05 ± 1.28E+05 ≈	3.07E+05 ± 1.01E+06 ≈	**6.16E+04 ± 1.47E+04** −	1.88E+05 ± 1.86E+05 +
F19	9.28E+03 ± 6.23E+03 ≈	1.43E+04 ± 1.33E+04 +	**5.42E+03 ± 1.25E+03** ≈	1.72E+04 ± 5.34E+04 ≈
F20	2.67E+03 ± 1.86E+02 +	2.57E+03 ± 2.35E+02 +	2.86E+03 ± 2.24E+02 +	2.72E+03 ± 2.15E+02 +
F21	2.44E+03 ± 3.14E+01 +	2.43E+03 ± 3.53E+01 +	2.41E+03 ± 5.90E+01 +	2.43E+03 ± 3.73E+01 +
F22	4.19E+03 ± 2.22E+03 ≈	4.68E+03 ± 1.91E+03 ≈	2.30E+03 ± 3.91E-09 −	4.50E+03 ± 2.32E+03 ≈
F23	3.00E+03 ± 8.12E+01 +	2.93E+03 ± 8.75E+01 +	2.76E+03 ± 1.33E+02 +	3.00E+03 ± 8.74E+01 +
F24	3.27E+03 ± 1.12E+02 +	3.21E+03 ± 1.43E+02 +	2.92E+03 ± 3.58E+01 ≈	3.18E+03 ± 7.29E+01 +
F25	2.92E+03 ± 1.66E+01 +	3.02E+03 ± 7.53E+01 +	**2.89E+03 ± 7.59E+00** −	3.00E+03 ± 4.61E+01 +
F26	5.56E+03 ± 1.63E+03 +	5.70E+03 ± 1.30E+03 +	2.85E+03 ± 5.07E+01 −	5.98E+03 ± 1.26E+03 +
F27	3.52E+03 ± 1.19E+02 +	3.52E+03 ± 1.36E+02 +	3.25E+03 ± 2.08E+01 +	3.43E+03 ± 1.50E+02 +
F28	3.21E+03 ± 7.43E+01 ≈	3.52E+03 ± 2.00E+02 +	**3.11E+03 ± 2.82E+01** −	3.44E+03 ± 9.79E+01 +
F29	4.12E+03 ± 3.03E+02 +	4.24E+03 ± 3.80E+02 +	4.05E+03 ± 1.88E+02 +	4.47E+03 ± 3.17E+02 +
F30	7.95E+04 ± 1.81E+05 +	3.39E+06 ± 1.42E+07 +	1.10E+04 ± 2.60E+03 +	3.60E+06 ± 6.27E+06 +
Rank	5	8	2	7
+/≈/−	20 /7 /2	25/ 3 /1	13/ 6 /10	25/ 3/1

one peak, 1-D sine with four peaks, and 2-D Griewank problems. Their formulas, number of training samples and test samples are listed in Table 9. With regard to the prediction problems, their details are presented in Table 10, including Mackey Glass, Box Jenkins and EEG times series data. Besides, the reasonable combination of three DNM parameters for these tested problems are given in Table 13, respectively.

The experimental results obtained by BSO and MDBSO for training DNM are shown in Table 13 where better results are highlighted. A Wilcoxon rank-sum test is used to analyze the significant difference between the performance of BSO and MDBSO. It can be observed that MDBSO obtains better results than BSO on all test problems, and seven out of ten are significantly better. Thus, the conclusion can be drawn that MDBSO is more superior than BSO in training DNM, which exhibits that it is promising to be applied to more fields.

In addition, Fig. 5 presents a logic circuit (LC) of DNM trained by MDBSO for the heart dataset, after implementing the neuronal pruning function [62], [65]. In LCs, comparator which is an analog-to-digital converter and logical "NOT", "AND" and "OR" gates are the major components. Comparator outputs 1 when the input is greater than the threshold θ, otherwise it outputs 0. More details about the pruning method can be referred in [62], [65]. In this LC, the number of attributes is decreased from 10 to 5, which means the structure is greatly simplified and it can be easily implemented in hardware. By doing so, this model achieves a high computational speed and exhibits its practical value.

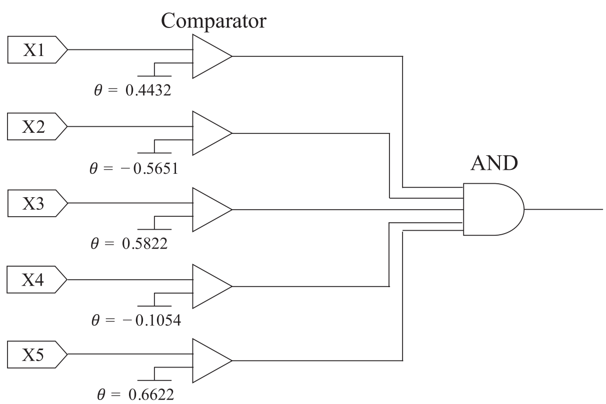


FIGURE 5. LC of Heart dataset trained by MDBSO.

V. DISCUSSION

A. ANALYSIS OF PRESET PARAMETERS

The number of parameters of MDBSO is reduced to two, including the number of cluster n and the mean value of Gaussian distribution μ. This indicates the effectiveness of the proposed concrete structure and adaptive parameters.

TABLE 6. Experimental results of MDBSO versus ABC variants on CEC'17 benchmark functions (1).

Fun.	MDBSO Mean ±Std Dev	ABC Mean ±Std Dev	GABC Mean ±Std Dev	MABC Mean ±Std Dev
F1	3.17E+03 ± 4.68E+03	4.72E+04 ± 7.74E+04 +	5.27E+03 ± 5.71E+03 +	2.01E+03 ± 1.82E+03 ≈
F3	**6.13E+02 ± 6.28E+02**	1.03E+05 ± 1.09E+04 +	9.20E+04 ± 1.02E+04 +	9.71E+04 ± 1.28E+04 +
F4	**4.62E+02 ± 3.28E+01**	5.19E+02 ± 2.79E+00 +	4.82E+02 ± 3.32E+01 +	5.17E+02 ± 2.31E+00 +
F5	6.02E+02 ± 3.10E+01	7.18E+02 ± 9.44E+00 +	**5.96E+02 ± 2.01E+01** ≈	7.19E+02 ± 1.48E+01 +
F6	6.13E+02 ± 7.18E+00	6.00E+02 ± 6.69E-03 −	6.00E+02 ± 1.15E-01 −	6.00E+02 ± 7.55E-04 −
F7	**8.88E+02 ± 6.43E+01**	9.43E+02 ± 9.71E+00 +	8.38E+02 ± 3.40E+01 −	9.39E+02 ± 9.74E+00 +
F8	9.13E+02 ± 3.75E+01	1.02E+03 ± 1.16E+01 +	**8.91E+02 ± 2.13E+01** −	1.02E+03 ± 1.08E+01 +
F9	1.31E+03 ± 5.38E+02	1.90E+03 ± 4.45E+02 +	2.08E+03 ± 1.08E+03 +	1.41E+03 ± 3.36E+02 +
F10	**7.37E+03 ± 1.29E+03**	8.10E+03 ± 3.19E+02 +	8.20E+03 ± 2.16E+02 +	8.15E+03 ± 3.09E+02 +
F11	**1.21E+03 ± 5.06E+01**	4.37E+03 ± 7.31E+02 +	1.71E+03 ± 7.64E+02 +	4.31E+03 ± 6.19E+02 +
F12	**4.39E+04 ± 2.26E+04**	1.17E+08 ± 2.66E+07 +	1.30E+06 ± 1.06E+06 +	7.79E+07 ± 2.79E+07 +
F13	1.46E+04 ± 1.61E+04	8.02E+07 ± 3.32E+07 +	**8.32E+03 ± 7.15E+03** ≈	8.46E+07 ± 2.90E+07 +
F14	**7.71E+03 ± 5.29E+03**	3.04E+05 ± 1.25E+05 +	1.88E+05 ± 9.66E+04 +	3.62E+05 ± 1.64E+05 +
F15	7.51E+03 ± 8.01E+03	2.08E+07 ± 8.65E+06 +	**7.32E+03 ± 7.67E+03** ≈	1.96E+07 ± 7.41E+06 +
F16	**2.44E+03 ± 4.43E+02**	3.76E+03 ± 1.87E+02 +	2.48E+03 ± 2.19E+02 ≈	3.68E+03 ± 1.57E+02 +
F17	**1.98E+03 ± 1.94E+02**	2.49E+03 ± 1.19E+02 +	2.05E+03 ± 1.34E+02 +	2.50E+03 ± 1.17E+02 +
F18	**9.28E+04 ± 5.38E+04**	6.34E+06 ± 3.20E+06 +	5.10E+06 ± 2.12E+06 +	6.77E+06 ± 2.63E+06 +
F19	9.70E+03 ± 9.19E+03	2.39E+07 ± 1.03E+07 +	**6.00E+03 ± 5.19E+03** ≈	2.67E+07 ± 1.04E+07 +
F20	**2.20E+03 ± 1.19E+02**	2.74E+03 ± 8.15E+01 +	2.72E+03 ± 8.80E+01 +	2.75E+03 ± 1.06E+02 +
F21	**2.40E+03 ± 4.16E+01**	2.52E+03 ± 1.18E+01 +	2.40E+03 ± 2.35E+01 ≈	2.51E+03 ± 1.18E+01 +
F22	4.78E+03 ± 3.01E+03	2.64E+03 ± 2.08E+02 ≈	2.30E+03 ± 1.56E+00 −	2.52E+03 ± 1.76E+02 ≈
F23	**2.73E+03 ± 2.30E+01**	2.89E+03 ± 1.60E+01 +	2.78E+03 ± 3.12E+01 +	2.88E+03 ± 1.65E+01 +
F24	**2.92E+03 ± 5.27E+01**	3.04E+03 ± 1.17E+01 +	2.95E+03 ± 3.66E+01 ≈	3.04E+03 ± 1.19E+01 +
F25	2.89E+03 ± 1.20E+01	2.89E+03 ± 1.73E-01 −	2.90E+03 ± 1.45E+01 +	**2.89E+03 ± 1.29E-01** −
F26	**4.78E+03 ± 6.31E+02**	5.74E+03 ± 1.28E+02 +	5.08E+03 ± 7.13E+02 +	5.71E+03 ± 1.13E+02 +
F27	3.23E+03 ± 2.01E+01	3.46E+03 ± 3.87E+01 +	3.25E+03 ± 1.52E+01 +	3.46E+03 ± 2.89E+01 +
F28	**3.19E+03 ± 5.78E+01**	3.26E+03 ± 2.59E+01 +	3.22E+03 ± 2.59E+01 +	3.23E+03 ± 1.95E+01 +
F29	**3.70E+03 ± 1.92E+02**	4.93E+03 ± 1.31E+02 +	3.73E+03 ± 1.69E+02 ≈	4.86E+03 ± 1.75E+02 +
F30	**8.49E+03 ± 3.24E+03**	2.67E+07 ± 1.04E+07 +	1.08E+04 ± 2.81E+03 +	2.38E+07 ± 7.73E+06 +
Rank	1	7	3	6
+/≈/−	- /- /-	26/ 1 /2	17/ 8/4	25/ 2 /2

Moreover, to be more precise, n and μ need to be analyzed to find the best parameter set for MDBSO.

1) ANALYSIS OF THE NUMBER OF CLUSTERS

In this part, n is first investigated. Four values, 3, 7, 9 and the original number 5 in BSO, are tested. The Wilcoxon rank-sum result is given in Table 13 and we can find that 5 is the value with the best performance.

2) ANALYSIS OF μ

Besides $\mu = 0.5$, we also investigate the values of -0.5, 0 and 1 to decide the best parameter for Gaussian distribution. In Table 15, $\mu = 0.5$ is significantly better than 0 and 1. Although $\mu = -0.5$ has similar performance with 0.5, $\mu = 0.5$ is still the best choice for MDBSO.

B. ANALYSIS OF POPULATION DIVERSITY

The experimental results have proven that the MDBSO has superior performance than BSO and other SI algorithms. It is owing to the multiple diversity-driven strategy which keeps a well balance between exploration and exploitation. In this part, the graphs of population diversity are illustrated to deeply analyze its effectiveness in optimization process. Figs. 6 and 7 are the curves of distance and fitness diversities of BSO and MDBSO on F12, F13, F23 and F24, respectively. In each subgraph, the population diversity of each cluster is depicted based on the average value of 30 runs. The horizontal axis is the number of function evaluations and the vertical axis is the average population diversity (including distance diversity D_d and fitness diversity D_f).

It should be noticed that in Fig. 6 the population diversity of the last cluster in BSO is always decreasing fast and stays in a very low order of magnitude until the end of search process. It indicates that only a few individuals remain in this cluster. As we introduced in Section II, the best individual in each cluster is selected as the center after k-means clustering. The situation of individuals keeping emigrating reports that the center in the last cluster obtains the worst quality in comparison with others. Thus, it is not competitive in attracting other individuals, which reflects the deficiency of the original individual generation strategy of BSO. If a cluster in BSO can not generate individuals with better fitness at the early stage of optimization, its size will continue to decease and never has a chance to rebound. However, in Fig. 7, the curves of MDBSO have more coordination than these of BSO. The population diversity of the last cluster keeps the same level with other cluster from the beginning to the end. Even if there is a deterioration in the middle, it will rebound quickly after several generations. This is due to the proposed mutation strategies in MDBSO which can efficiently generate new individuals with good fitness and endow the cluster with considerable attraction.

TABLE 7. Experimental results of MDBSO versus ABC variants on CEC'17 benchmark functions (2).

Fun.	SeABC Mean ±Std Dev	RABC Mean ±Std Dev	SFABC Mean ±Std Dev
F1	2.87E+09 ± 6.67E+09 +	4.36E+03 ± 5.73E+03 ≈	**1.40E+03 ± 1.26E+03** ≈
F3	9.90E+04 ± 1.15E+04 +	7.73E+04 ± 1.01E+04 +	9.82E+04 ± 1.38E+04 +
F4	5.06E+02 ± 3.15E+01 +	4.93E+02 ± 1.96E+01 +	5.05E+02 ± 2.44E+01 +
F5	6.58E+02 ± 5.81E+01 +	6.55E+02 ± 1.89E+01 +	6.98E+02 ± 1.51E+01 +
F6	6.29E+02 ± 2.66E+01 +	**6.00E+02 ± 7.35E-06** −	6.00E+02 ± 7.36E-06 −
F7	9.46E+02 ± 1.86E+02 +	8.91E+02 ± 1.80E+01 +	9.27E+02 ± 1.12E+01 +
F8	9.50E+02 ± 7.14E+01 +	9.59E+02 ± 2.36E+01 +	9.96E+02 ± 1.48E+01 +
F9	4.19E+03 ± 3.90E+03 +	9.58E+02 ± 1.22E+02 −	**9.00E+02 ± 1.18E-07** −
F10	8.07E+03 ± 2.47E+02 +	7.89E+03 ± 3.37E+02 +	8.11E+03 ± 3.22E+02 +
F11	2.90E+03 ± 1.70E+03 +	1.30E+03 ± 4.74E+01 +	1.82E+03 ± 4.48E+02 +
F12	2.59E+08 ± 6.64E+08 +	2.37E+06 ± 2.08E+06 +	1.23E+06 ± 7.26E+05 +
F13	2.34E+08 ± 6.04E+08 ≈	2.04E+04 ± 2.86E+04 +	1.06E+04 ± 6.52E+03 ≈
F14	2.56E+05 ± 1.95E+05 +	1.44E+05 ± 8.34E+04 +	1.93E+05 ± 1.00E+05 +
F15	3.63E+06 ± 1.58E+07 ≈	1.11E+04 ± 2.13E+04 ≈	1.41E+04 ± 3.07E+04 ≈
F16	2.86E+03 ± 6.08E+02 +	3.10E+03 ± 2.13E+02 +	3.45E+03 ± 1.55E+02 +
F17	2.37E+03 ± 2.86E+02 +	2.13E+03 ± 1.53E+02 +	2.33E+03 ± 1.28E+02 +
F18	4.25E+06 ± 3.33E+06 +	3.94E+06 ± 1.64E+06 +	5.93E+06 ± 3.06E+06 +
F19	2.21E+07 ± 7.50E+07 ≈	7.38E+05 ± 1.83E+06 +	1.37E+04 ± 2.86E+04 ≈
F20	2.74E+03 ± 8.62E+01 +	2.51E+03 ± 8.88E+01 +	2.76E+03 ± 7.80E+01 +
F21	2.48E+03 ± 7.97E+01 +	2.45E+03 ± 2.60E+01 +	2.49E+03 ± 1.15E+01 +
F22	3.02E+03 ± 1.68E+03 ≈	2.31E+03 ± 4.19E+00 ≈	**2.30E+03 ± 1.53E-08** −
F23	2.84E+03 ± 1.17E+02 +	2.77E+03 ± 2.89E+01 +	2.84E+03 ± 2.05E+01 +
F24	3.12E+03 ± 2.01E+02 +	2.99E+03 ± 2.24E+01 +	3.01E+03 ± 1.15E+01 +
F25	3.04E+03 ± 3.94E+02 ≈	2.89E+03 ± 7.25E+00 −	2.89E+03 ± 6.79E-01 −
F26	6.07E+03 ± 1.80E+03 +	4.90E+03 ± 3.44E+02 ≈	5.29E+03 ± 2.74E+02 +
F27	3.33E+03 ± 1.59E+02 +	**3.22E+03 ± 8.70E+00** −	3.25E+03 ± 1.80E+01 +
F28	3.42E+03 ± 4.18E+02 +	3.22E+03 ± 1.98E+01 +	3.21E+03 ± 1.08E+01 ≈
F29	4.05E+03 ± 4.12E+02 +	3.92E+03 ± 2.17E+02 +	4.32E+03 ± 2.23E+02 +
F30	1.83E+07 ± 5.10E+07 +	3.63E+05 ± 2.42E+05 +	2.33E+05 ± 3.07E+05 +
Rank	5	2	4
+/ ≈ /−	24 /5 /0	21/ 4 /4	20/ 5/4

C. MDBSO WITH FITNESS-BASED GROUPING

It is introduced above that many attempts are made to improve the efficiency of BSO in clustering as the k-means is a time-cost clustering method. The modifications in BSO-OS, RGBSO and GBSO include improvements to clustering methods. For example, BSO-OS and RGBSO apply random grouping to minimize the clustering overhead. Although this method divides the population into elitists and normal individuals, it can not provide any specific measures for clustering. Thus, it is not suitable for MDBSO since the population diversity of each cluster is not available. In GBSO, a fitness-based grouping strategy is presented in which the individuals are ranked according to their fitness. The individuals with good and bad fitness are equally distributed into different groups. Fitness-based grouping is proven to be a less time-cost and effective method in [20]. Therefore, in this part, we will discuss whether the fitness-based grouping method could further improve the performance of MDBSO by replacing k-means clustering. The combination is named as MDBSO-FG.

Table 15 provides the Wilcoxon rank-sum test result and one conclusion can be drawn that the fitness-based grouping method is not suitable for MDBSO. In current situation, although k-means clustering is a computational cost method, it still has good performance in optimization results. Finding a method which can outperform k-means in both efficiency and effectiveness becomes a challenge in our future research.

TABLE 8. Details of the classification data sets.

Classification data sets	# of attributes	# of training samples	# of test samples	# of classes
3-bits XOR	3	8	8	2
Ballon	4	16	16	2
Iris	4	150	150	2
Heart	10	297	297	2

D. COMPARISON WITH MIIBSO

BSO with multi-information interactions (MIIBSO) [66] is a newly proposed state-of-the-art BSO variation. It proposes a multi-information interaction (MII) strategy which contains three patterns to enhance the information interaction capability between individuals. Moreover, it uses a random grouping strategy to replace the k-means clustering method. The comparison result between MDBSO and MIIBSO is shown in Table 16. The results including mean and standard deviation (Std Dev) are exhibited and the values in boldface represent the better results. Each $+/\approx/-$ indicates the performance of MDBSO is significantly better (+), not significantly better and worse (≈) or worse (−) than MIIBSO.

As observed, MDBSO can significantly outperform MIIBSO on 19 out of the total of 29 functions. Specifically, the search ability of MDBSO is much better than that of MIIBSO on F1-F10 as they are unimodal and simple multimodal functions. The good performance of MIIBSO on F14-F20 reveals its effect for solving hybrid functions.

TABLE 9. Details of the function approximation data sets.

Function approximation datasets	# of Training Samples	# of Test Samples
Cosine: $y = cos(x\pi/2)^7$	31: $x \in [1.25 : 0.05 : 2.75]$	38: $x \in [1.25 : 0.04 : 2.75]$
Sine: $y = sin(2x)$	126: $x \in [-2\pi : 0.1 : 2\pi]$	252: $x \in [-2\pi : 0.05 : 2\pi]$
Griewank: $z = \frac{1}{4000}\sum_{i=1}^{2} x_i^2 - \prod_{i=1}^{2} cos(\frac{x_i}{\sqrt{i}}) + 1$, $x = x_1$ and $y = x_2$	21× 21: $x, y \in [-2 : 0.1 : 2]$	41× 41: $x, y \in [-4 : 0.05 : 2]$

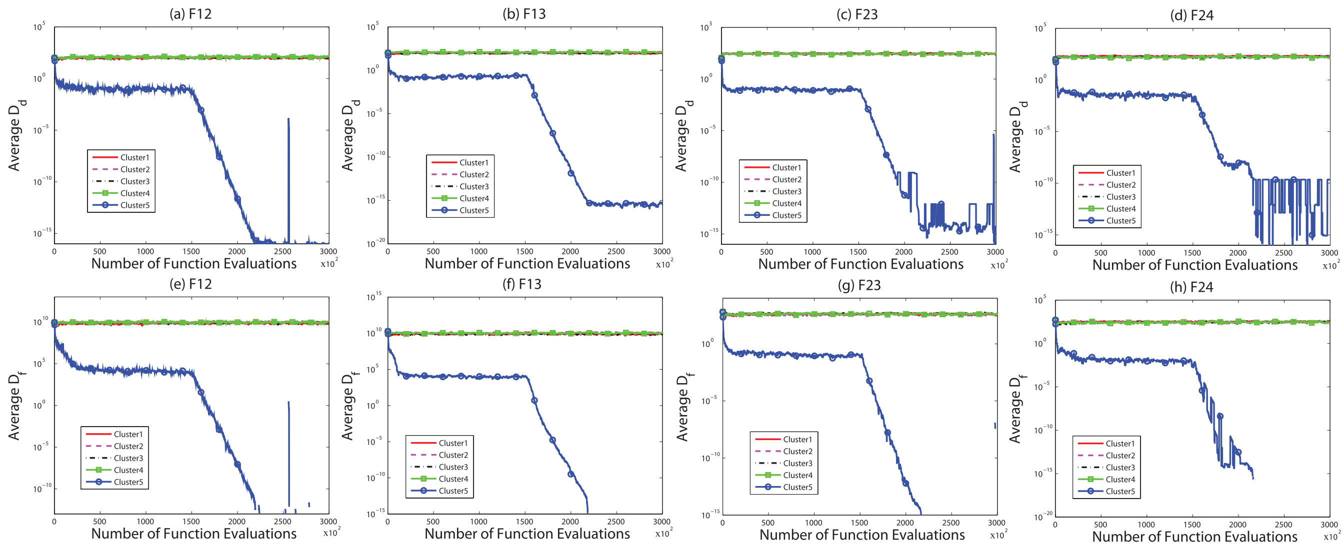


FIGURE 6. The curves of distance diversity and fitness diversity of BSO on F12, F13, F23 and F24.

TABLE 10. Details of the prediction data sets.

Classification data sets	# of training samples	# of test samples
Mackey Glass	450	550
Box Jenkins	140	156
EEG	1000	1500

TABLE 11. Reasonable combination of three parameters for nine tested problems, respectively.

Problems	M	k	θ_s
XOR	6	3	0.5
Balloon	7	3	0.5
Iris	7	3	0.5
Heart	14	3	0.5
Cosine	23	3	0.5
Sine	22	3	0.5
Griewank	15	3	0.5
Mackey Glass	12	3	0.5
Box Jenkins	15	3	0.5
EEG	12	3	0.5

TABLE 12. Experimental results of DNM training by MDBSO and BSO, respectively.

	MDBSO	BSO
	Mean (Std Dev)	Mean (Std Dev)
XOR	**2.89E-02** (**2.08E-02**)	1.42E-01 + (3.86E-02)
Balloon	**2.00E-02** (**8.84E-06**)	2.41E-02 + (5.86E-03)
Iris	**1.11E-02** (**7.83E-09**)	1.11E-02 + (3.97E-05)
Heart	**5.97E-02** (**1.60E-02**)	8.89E-02 + (2.30E-02)
Cosine	**5.41E-03** (**5.84E-04**)	2.92E-02 ≈ (6.40E-02)
Sine	**2.38E-01** (**7.43E-03**)	2.38E-01 ≈ (1.42E-02)
Griewank	**8.18E-02** (**1.62E-02**)	9.97E-02 + (2.10E-02)
Mackey Glass	**3.54E-04** (**1.66E-04**)	3.99E-04 ≈ (2.01E-04)
Box Jenkins	**4.22E-03** (**1.23E-04**)	4.39E-03 + (9.38E-05)
EEG	**5.59E-03** (**1.80E-04**)	5.64E-03 + (1.46E-04)

This encourages us to further enhance the search ability of MDBSO on complex problems.

E. COMPUTATIONAL COMPLEXITY

In this part, we compare the computational complexity of MDBSO with BSO to show its efficiency.

The time complexity of BSO is calculated as follows:

(1) The initialization of population and parameters in BSO needs the time complexity $O(N)$ where N is the population size.
(2) The population evaluation process needs $O(N)$.

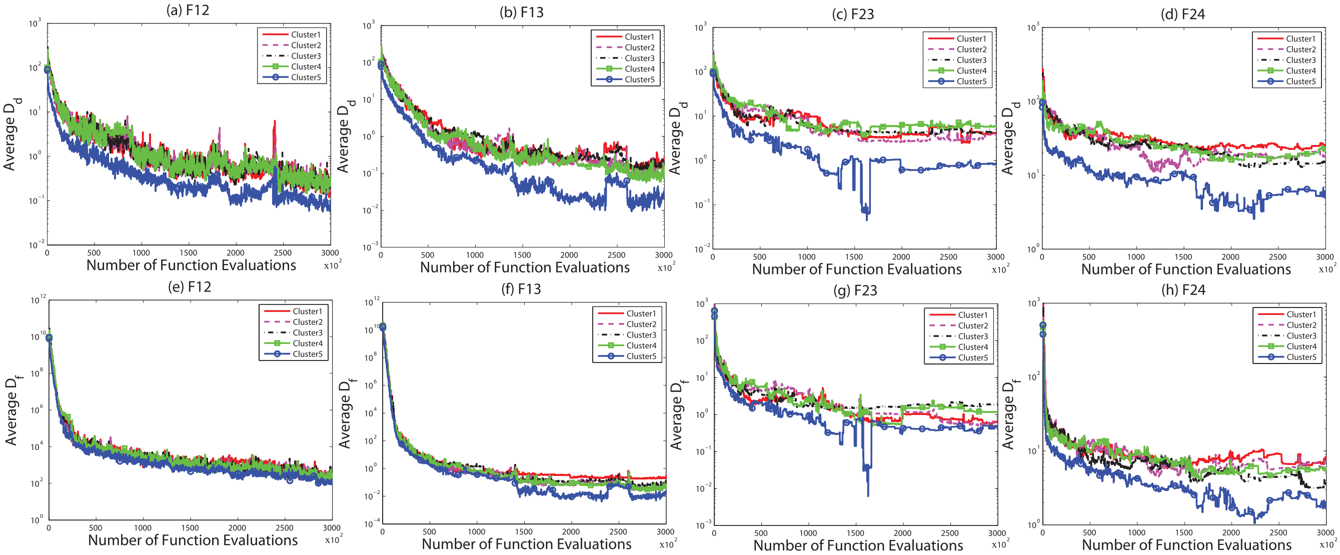


FIGURE 7. The curves of distance diversity and fitness diversity of MDBSO on F12, F13, F23 and F24.

TABLE 13. Wilcoxon rank-sum test results of different numbers of clusters in MDBSO.

n=5 *vs.*	n=3	n=7	n=9
$+/\approx/-$	21 /8 /0	10/ 19 /0	12/ 17/0

TABLE 14. Wilcoxon rank-sum test results of different numbers of μ in MDBSO.

$\mu = 0.5$ *vs.*	$\mu = -0.5$	$\mu = 0$	$\mu = 1$
$+/\approx/-$	5 / 24/0	10 / 19 /0	14/ 14/1

TABLE 15. Wilcoxon rank-sum test result of MDBSO *vs.* MDBSO-FG.

MDBSO *vs.*	MDBSO-FG
$+/\approx/-$	28 / 1 /0

(3) Using k-means clustering method to divide the population into 5 clusters needs $O(5N^2)$.
(4) The process of individual selection and generation of step length both cost $O(N^2)$.
(5) The generation of new individuals and the fitness calculation needs the time complexity $O(N^2)$, respectively.

Therefore, the overall time complexity of BSO is

$$O(N) + O(N) + O(5N^2) + O(N^2) + O(N^2) = 2O(N^2) + O(5N^2) + 2O(N) \quad (11)$$

To be simplified, its overall time complexity is $O(N^2)$.

The time complexity of MDBSO is

(1) The initialization process is $O(N)$.
(2) The fitness evaluation is $O(N)$.
(3) Using k-means clustering method to divide the population into 5 clusters needs $O(5N^2)$.
(4) Calculating the distance diversity (D_d) and fitness diversity (D_f) needs $O(N^2)$, respectively.
(5) Selection of mutation strategies costs $O(N^2)$.

TABLE 16. Experimental results of MDBSO versus MIIBSO on CEC'17 benchmark functions.

Fun.	MDBSO Mean ±Std Dev	MIIBSO Mean ±Std Dev
F1	**3.17E+03 ± 4.68E+03**	1.95E+10 ± 5.48E+09 +
F3	**6.13E+02 ± 6.28E+02**	3.29E+04 ± 1.10E+04 +
F4	**4.62E+02 ± 3.28E+01**	1.35E+03 ± 5.67E+02 +
F5	**6.02E+02 ± 3.10E+01**	6.49E+02 ± 2.44E+01 +
F6	**6.13E+02 ± 7.18E+00**	6.23E+02 ± 4.66E+00 +
F7	**8.88E+02 ± 6.43E+01**	9.84E+02 ± 5.30E+01 +
F8	**9.13E+02 ± 3.75E+01**	9.27E+02 ± 2.03E+01 +
F9	**1.31E+03 ± 5.38E+02**	2.42E+03 ± 7.60E+02 +
F10	**7.37E+03 ± 1.29E+03**	8.04E+03 ± 5.05E+02 +
F11	**1.21E+03 ± 5.06E+01**	1.45E+03 ± 1.65E+02 +
F12	**4.39E+04 ± 2.26E+04**	7.61E+08 ± 7.02E+08 +
F13	**1.46E+04 ± 1.61E+04**	3.50E+04 ± 2.67E+04 +
F14	7.71E+03 ± 5.29E+03	**1.48E+03 ± 3.85E+01** −
F15	7.51E+03 ± 8.01E+03	**2.98E+03 ± 9.77E+02** −
F16	2.44E+03 ± 4.43E+02	**2.24E+03 ± 2.60E+02** −
F17	1.98E+03 ± 1.94E+02	**1.83E+03 ± 4.03E+01** −
F18	9.28E+04 ± 5.38E+04	**2.87E+03 ± 1.12E+03** −
F19	9.70E+03 ± 9.19E+03	**2.30E+03 ± 7.12E+02** −
F20	2.20E+03 ± 1.19E+02	**2.20E+03 ± 9.89E+01** ≈
F21	**2.40E+03 ± 4.16E+01**	2.43E+03 ± 2.31E+01 +
F22	**4.78E+03 ± 3.01E+03**	8.65E+03 ± 1.81E+03 +
F23	**2.73E+03 ± 2.30E+01**	2.86E+03 ± 6.81E+01 +
F24	**2.92E+03 ± 5.27E+01**	3.04E+03 ± 4.59E+01 +
F25	**2.89E+03 ± 1.20E+01**	3.20E+03 ± 1.70E+02 +
F26	**4.78E+03 ± 6.31E+02**	5.86E+03 ± 5.94E+02 +
F27	3.23E+03 ± 2.01E+01	**3.20E+03 ± 2.03E-04** −
F28	**3.19E+03 ± 5.78E+01**	3.30E+03 ± 2.13E-04 +
F29	3.70E+03 ± 1.92E+02	**3.41E+03 ± 1.20E+02** −
F30	8.49E+03 ± 3.24E+03	**4.67E+03 ± 9.39E+02** −
w /t/ l	- /- /-	19/1/9

(6) The generation of new individuals and the fitness calculation needs the time complexity $O(N^2)$, respectively.

Thus, the overall time complexity of MDBSO is

$$O(N) + O(N) + O(5N^2) + 2O(N^2) + O(N^2) + O(N^2) = 4O(N^2) + O(5N^2) + 2O(N) \quad (12)$$

The time complexity of MDBSO can be simplified as $O(N^2)$.

Although calculating D_d and D_f need more cost than BSO, the same overall time complexities of $O(N^2)$ indicate that MDBSO achieves the same computational efficiency in comparison with BSO. In other words, the multiple diversity-driven strategy can improve the performance of MDBSO and maintain efficiency. Thus, the utilization of D_d and D_f is promising to be applied to more SI algorithms.

VI. CONCLUSION

In this paper, a novel multiple diversity-driven BSO is proposed to improve the search ability of BSO. Two diversity measures, including distance diversity and fitness diversity, are collaborated to control the generation of adaptive parameters in optimization procedure. The diversities of each cluster in BSO is calculated to control the utilization of mutation strategies. Moreover, new individuals are generated according to these diversities. In this way, the number of parameters in original BSO is greatly decreased.

Experiments on CEC2017 benchmark function suit and ANN training task are utilized to investigate the performance of the proposed MDBSO. The results demonstrate that MDBSO obtains superior effectiveness, efficiency and robustness. In the comparison with the latest BSO variations, MDBSO exhibits overwhelming advantages, which indicates MDBSO is the current best BSO variation. Besides, diversity is a significant part in optimization search. Several researchers have realized its importance but we creatively take it into parameter adaptation and obtain desired result. It suggests that the proposed multiple diversity-driven strategy deserves more attention in SI research field.

In our future research, we will consistently concentrate on the parameter adaptation and the diversity management. Some other combinations can be attempted to reveal the effect of diversity in SI optimization algorithms.

REFERENCES

[1] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and Biological Systems: Towards a New Bionics?* Berlin, Germany: Springer, 1993, pp. 703–712.
[2] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Berlin, Germany: Springer, 2011, pp. 760–766.
[3] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
[4] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Proc. Int. Conf. Swarm Intell.* Berlin, Germany: Springer, 2010, pp. 355–364.
[5] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.
[6] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Apr. 2007.
[7] Y. Shi, "Brain storm optimization algorithm," in *Proc. Int. Conf. Swarm Intell.* Berlin, Germany: Springer, 2011, pp. 303–309.
[8] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang, "Ant colony optimization with clustering for solving the dynamic location routing problem," *Appl. Math. Comput.*, vol. 285, pp. 149–173, Jul. 2016.
[9] S. Gao, Y. Todo, T. Gong, G. Yang, and Z. Tang, "Graph planarization problem optimization based on triple-valued gravitational search algorithm," *IEEJ Trans. Elect. Electron. Eng.*, vol. 9, no. 1, pp. 39–48, 2014.
[10] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," *Appl. Math. Comput.*, vol. 231, pp. 48–62, Mar. 2014.
[11] Z. Song, S. Gao, Y. Yu, J. Sun, and Y. Todo, "Multiple chaos embedded gravitational search algorithm," *IEICE Trans. Inf. Syst.*, vol. 100, no. 4, pp. 888–900, 2017.
[12] S. Cheng, Q. Qin, J. Chen, and Y. Shi, "Brain storm optimization algorithm: A review," *Artif. Intell. Rev.*, vol. 46, no. 4, pp. 445–458, 2016.
[13] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
[14] J. Ji, S. Gao, S. Wang, Y. Tang, H. Yu, and Y. Todo, "Self-adaptive gravitational search algorithm with a modified chaotic local search," *IEEE Access*, vol. 5, pp. 17881–17895, 2017.
[15] C. Vairappan, H. Tamura, S. Gao, and Z. Tang, "Batch type local search-based adaptive neuro-fuzzy inference system (ANFIS) with self-feedbacks for time-series prediction," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1870–1877, 2009.
[16] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb. 2008.
[17] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.
[18] Y. Shi, "Brain storm optimization algorithm in objective space," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 1227–1234.
[19] Z. Cao, Y. Shi, X. Rong, B. Liu, Z. Du, and B. Yang, "Random grouping brain storm optimization algorithm with a new dynamically changing step size," in *Proc. Int. Conf. Swarm Intell.* Berlin, Germany: Springer, 2015, pp. 357–364.
[20] M. El-Abd, "Global-best brain storm optimization algorithm," *Swarm Evol. Comput.*, vol. 37, pp. 27–44, Dec. 2017.
[21] Y. Yu, S. Gao, S. Cheng, Y. Wang, S. Song, and F. Yuan, "CBSO: A memetic brain storm optimization with chaotic local search," *Memetic Comput.*, vol. 10, no. 4, pp. 353–367, 2018.
[22] X. Guo, Y. Wu, L. Xie, S. Cheng, and J. Xin, "An adaptive brain storm optimization algorithm for multiobjective optimization problems," in *Proc. Int. Conf. Swarm Intell.* Berlin, Germany: Springer, 2015, pp. 365–372.
[23] Y. Yu, S. Gao, Y. Wang, J. Cheng, and Y. Todo, "ASBSO: An improved brain storm optimization with flexible search length and memory-based selection," *IEEE Access*, vol. 6, pp. 36977–36994, 2018.
[24] M. El-Abd, "Brain storm optimization algorithm with re-initialized ideas and adaptive step size," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 2682–2686.
[25] Á. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, Jul. 1999.
[26] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[27] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
[28] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2013, pp. 71–78.
[29] Y. Shi and R. C. Eberhart, "Population diversity of particle swarms," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.) (CEC)*, Jun. 2008, pp. 1063–1067.
[30] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "Enhancing artificial bee colony algorithm using more information-based search equations," *Inf. Sci.*, vol. 270, pp. 112–133, Jun. 2014.
[31] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591–607, Oct. 2011.
[32] C. Segura, C. A. C. Coello, E. Segredo, and A. H. Aguirre, "A novel diversity-based replacement strategy for evolutionary algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3233–3246, Dec. 2016.
[33] C. Segura, A. Hernández-Aguirre, F. Luna, and E. Alba, "Improving diversity in evolutionary algorithms: New best solutions for frequency assignment," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 539–553, Aug. 2017.
[34] S. Cheng, Y. Shi, Q. Qin, T. O. Ting, and R. Bai, "Maintaining population diversity in brain storm optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 3230–3237.

[35] E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor, "Advanced population diversity measures in genetic programming," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Springer, 2002, pp. 341–350.

[36] Y. G. Woldesenbet and G. G. Yen, "Dynamic evolutionary algorithm with variable relocation," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 500–513, Jun. 2009.

[37] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner, "A fast adaptive memetic algorithm for online and offline control design of PMSM drives," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 28–41, Feb. 2007.

[38] C. C. Coello, G. T. Pulido, and E. M. Montes, "Current and future research trends in evolutionary multiobjective optimization," in *Information Processing with Evolutionary Algorithms*. Berlin, Germany: Springer, 2005, pp. 213–231.

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

[40] D. Molina, M. Lozano, C. García-Martínez, and F. Herrera, "Memetic algorithms for continuous optimisation based on local search chains," *Evol. Comput.*, vol. 18, no. 1, pp. 27–63, 2010.

[41] T. Nomura and K. Shimohara, "An analysis of two-parent recombinations for real-valued chromosomes in an infinite population," *Evol. Comput.*, vol. 9, no. 3, pp. 283–308, 2001.

[42] N. Awad, M. Ali, J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," NTU, Singapore, Tech. Rep., 2016.

[43] Y. Todo, H. Tamura, K. Yamashita, and Z. Tang, "Unsupervised learnable neuron model with nonlinear interaction on dendrites," *Neural Netw.*, vol. 60, pp. 96–103, Dec. 2014.

[44] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.

[45] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis," *Artif. Intell. Rev.*, vol. 12, no. 4, pp. 265–319, 1998.

[46] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 Special Session on Real Parameter Optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[47] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, 2010.

[48] S. Mirjalili and A. Lewis, "Adaptive gbest-guided gravitational search algorithm," *Neural Comput. Appl.*, vol. 25, nos. 7–8, pp. 1569–1584, 2014.

[49] B. Gu and F. Pan, "Modified gravitational search algorithm with particle memory ability and its application," *Int. J. Innov. Comput., Inf. Control*, vol. 9, no. 11, pp. 4531–4544, 2013.

[50] S. Mirjalili and S. Z. M. Hashim, "A new hybrid PSOGSA algorithm for function optimization," in *Proc. Int. Conf. Comput. Inf. Appl. (ICCIA)*, Dec. 2010, pp. 374–377.

[51] Y. Wang, Y. Yu, S. Gao, H. Pan, and G. Yang, "A hierarchical gravitational search algorithm with an effective gravitational constant," *Swarm Evol. Comput.*, vol. 46, pp. 118–139, May 2019.

[52] A. Zhang, G. Sun, J. Ren, X. Li, Z. Wang, and X. Jia, "A dynamic neighborhood learning-based gravitational search algorithm," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 436–447, Jan. 2018.

[53] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009.

[54] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Appl. Math. Comput.*, vol. 217, pp. 3166–3173, Dec. 2010.

[55] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Comput.*, vol. 22, no. 9, pp. 2935–2952, May 2017.

[56] X. Li and G. Yang, "Artificial bee colony algorithm with memory," *Appl. Soft Comput.*, vol. 41, pp. 362–372, Apr. 2016.

[57] G. Li, L. Cui, X. Fu, Z. Wen, N. Lu, and J. Lu, "Artificial bee colony algorithm with gene recombination for numerical function optimization," *Appl. Soft Comput.*, vol. 52, pp. 146–159, Mar. 2017.

[58] J. Ji, S. Song, C. Tang, S. Gao, Z. Tang, and Y. Todo, "An artificial bee colony algorithm search guided by scale-free networks," *Inf. Sci.*, vol. 473, pp. 142–165, Jan. 2019.

[59] T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, and Z. Tang, "Financial time series prediction using a dendritic neuron model," *Knowl.-Based Syst.*, vol. 105, pp. 214–224, Aug. 2016.

[60] T. Jiang, S. Gao, D. Wang, J. Ji, Y. Todo, and Z. Tang, "A neuron model with synaptic nonlinearities in a dendritic tree for liver disorders," *IEEJ Trans. Elect. Electron. Eng.*, vol. 12, no. 1, pp. 105–115, 2017.

[61] W. Chen, J. Sun, S. Gao, J. Cheng, J. Wang, and Y. Todo, "Using a single dendritic neuron to forecast tourist arrivals to Japan," *IEICE Trans. Inf. Syst.*, vol. 100, no. 1, pp. 190–202, 2017.

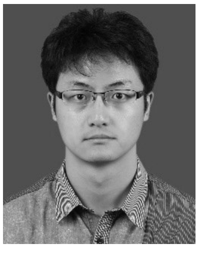
[62] Y. Tang, J. Ji, S. Gao, H. Dai, Y. Yu, and Y. Todo, "A pruning neural network model in credit classification analysis," *Comput. Intell. Neurosci.*, vol. 2018, Feb. 2018, Art. no. 9390410.

[63] Y.-J. Gong, J. Zhang, and Y. Zhou, "Learning multimodal parameters: A bare-bones niching differential evolution approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2944–2959, Jul. 2018.

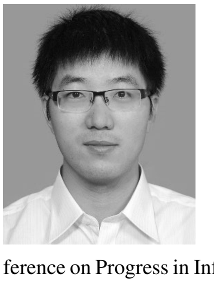
[64] C. L. Blake. (1998). *Uci Repository of Machine Learning Databases, Irvine, University of California*. [Online]. Available: http://archive.ics.uci.edu/ml/index.php

[65] J. Ji, S. Gao, J. Cheng, Z. Tang, and Y. Todo, "An approximate logic neuron model with a dendritic structure," *Neurocomputing*, vol. 173, pp. 1775–1783, Jan. 2016.

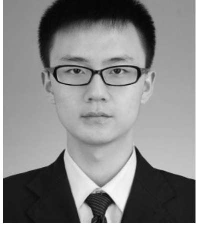
[66] C. Li, Z. Song, J. Fan, Q. Cheng, and P. X. Liu, "A brain storm optimization with multi-information interactions for global optimization problems," *IEEE Access*, vol. 6, pp. 19304–19323, 2018.



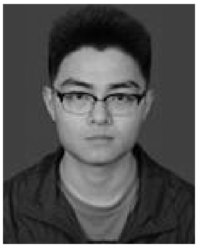
YANG YU received the B.E. degree in civil engineering from the Yancheng Institute of Technology, Yancheng, China, in 2014, and the M.S. degree from the University of Toyama, Toyama, Japan, in 2017, where he is currently pursuing the Ph.D. degree. His current research interests include data mining, optimization problems, and nature-inspired algorithms.



SHANGCE GAO (M'11–SM'16) received the Ph.D. degree in innovative life science from the University of Toyama, Toyama, Japan, in 2011, where he is currently an Associate Professor. He has authored over 100 publications in refereed journals and conference proceedings. His current research interests include nature-inspired technologies, mobile computing, machine learning, and neural networks. He was a recipient of the Best Paper Award at the IEEE International Conference on Progress in Informatics and Computing, the Shanghai Rising-Star Scientist Award, the Chen-Guang Scholar of Shanghai Award, the Outstanding Academic Performance Award of IEICE, and the Outstanding Academic Achievement Award of IPSJ.



YIRUI WANG received the B.S. and M.S. degrees from the College of Information Sciences and Technology, Donghua University, Shanghai, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the University of Toyama, Japan. His current research interests include computational intelligence, swarm intelligent algorithms, and combinatorial optimizations.



ZHENYU LEI received the B.E. degree from the Shandong University of Science and Technology, Qingdao, China, in 2017. He is currently pursuing the M.S. degree with the University of Toyama, Toyama, Japan. His current research interests include machine learning, optimization problems, and data mining.

JIUJUN CHENG received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2006. In 2009, he was a Visiting Professor with Aalto University, Espoo, Finland. He is currently a Professor with Tongji University, Shanghai, China. He has over 50 publications including conference and journal articles. His current research interests include mobile computing, complex networks with a focus on mobile/Internet interworking, complex networks, and Internet of vehicles.

YUKI TODO (M'13) received the B.S. degree from Zhejiang University, Zhejiang, China, in 1983, the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 1986, and the D.E. degree from Kanazawa University, Kanazawa, Japan, in 2005. From 1987 to 1989, she was an Assistant Professor with the Institute of Microelectronics, Shanghai Jiaotong University, Shanghai, China. From 1989 to 1990, she was a Research Student with Nagoya University, Nagoya, Japan. From 1990 to 2000, she was a Senior Engineer with Sanwa Newtech Inc., Miyazaki, Japan. From 2000 to 2011, she was with Tateyama Systems Institute, Toyama, Japan. In 2012, she joined Kanazawa University, where she is currently an Associate Professor with the School of Electrical and Computer Engineering. Her current research interests include multiple-valued logic, neural networks, and optimization.