IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 16, 2018, accepted February 21, 2018, date of publication March 2, 2018, date of current version April 18, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2811542*

# Locating Multiple Optima via Brain Storm Optimization Algorithms

**SHI CHENG[1], (Member, IEEE), JUNFENG CHEN[2], (Member, IEEE), XIUJUAN LEI[1], AND YUHUI SHI[3], (Fellow, IEEE)**

[1]School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
[2]College of IOT Engineering, Hohai University, Changzhou 213022, China
[3]Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

Corresponding author: Junfeng Chen (chen-1997@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672334, Grant 61773119, Grant 61403121, and Grant 61771297, in part by the Shenzhen Science and Technology Innovation Committee under Grant ZDSYS201703031748284, and in part by the Fundamental Research Funds for the Central Universities under Grant GK201703062 and Grant 2015B20214.

**ABSTRACT** Locating multiple optima/peaks in a single run and maintaining these found optima until the end of a run is the goal of multimodal optimization. Three variants of brain storm optimization (BSO) algorithms, which include original BSO algorithm, BSO in objective space algorithm with Gaussian random variable, and BSO in objective space algorithm with Cauchy random variable, were utilized to solve multimodal optimization problems in this paper. The experimental tests were conducted on eight benchmark problems and its applications in seven nonlinear equation system problems. The performance and effectiveness of various BSO algorithms on solving multimodal optimization problems were validated based on the experimental results. The conclusions could be made that the global search ability and solutions maintenance ability of an algorithm needs to be balanced simultaneously on solving multimodal optimization problems.

**INDEX TERMS** Brain storm optimization, swarm intelligence, brain storm optimization in objective space algorithm, multimodal optimization, nonlinear equation systems.

## NOMENCLATURE

| | |
|---|---|
| BSO | brain storm optimization algorithm; |
| BSO-OS | brain storm optimization in objective space algorithm; |
| DSI | developmental swarm intelligence algorithms; |
| FWA | fireworks algorithm; |
| PSO | particle swarm optimization algorithm; |
| NES | nonlinear equation system. |

## I. INTRODUCTION

Optimization is concerned with finding the optimum feasible solution(s) for a given optimized problem. An optimization problem is a mapping from decision space to objective space. The solutions are searched in the decision space, while the function value (objective) is evaluated in the objective space. For swarm intelligence or evolutionary computation algorithms, the solutions in the search space are represented by individuals in the swarm. The position of an individual is corresponded with decision variables of a solution in the decision space, while the fitness value of an individual corresponds with the objective value of the solution in the objective space. Individuals are guided toward the better and better search areas through the cooperation and competition among individuals until some stopping conditions are met.

Different algorithms could be summarized into a framework to analyze their common properties. Based on the framework, it could give a better understanding of algorithms and guide designing or implementing a new strategy. There are several most used frameworks, such as memetic computing methodologies [1], cultural algorithms, and developmental swarm intelligence (DSI) algorithms [2], *etc.* Developmental swarm intelligence algorithm is defined as a swarm intelligence algorithm with both capability learning ability and capacity developing ability [2].

Capability learning is a micro level learning ability, which focuses on its actual search from the current solution for single point based optimization algorithms and the current population for population-based swarm intelligence algorithms. Capability learning describes the ability of an algorithm to find the better solution(s) from current solution(s) with the learning capacity it possesses. This learning ability is focused on the data-driven approach. The aim of capability learning is to solve a problem more effectively based on the obtained solutions (data points). Capacity developing is a macro level learning ability, which focuses on moving the algorithm's search to the area(s) where higher search potential may exist. The capacity developing describes the learning ability of an algorithm to adaptively change its parameters, structures, and/or its learning potential according to the search states of the problem to be solved. In other words, the capacity developing is the search strength possessed by an algorithm. This learning ability is focused on the model-driven approach. The aim of capacity developing is to solve different problems through the parameter/structure adaptation.

Several swarm intelligence algorithms could be categorized as the developmental swarm intelligence algorithms. Brain storm optimization (BSO), fireworks algorithm (FWA) [3], [4], and particle swarm optimization algorithm are three typical DSI algorithms [2].

The brain storm optimization (BSO) algorithm is based on the collective behavior of human being, that is, the brainstorming process [5], [6]. The individuals in brain storm optimization are diverging into several clusters. The new individuals are generated based on the mutation of one existed individual or a combination of two individuals. In the original BSO algorithm, the clustering strategy is performed at each iteration. The computational resources are consumed a lot on the clustering operation. Thus, to reduce the computational burden, the clustering strategy needs to be modified. The brain storm optimization in objective space (BSO-OS) algorithm was proposed and the clustering strategy was replaced by a simple elitist strategy based on the fitness values [7]. For BSO algorithms, the "good enough" optimum could be obtained through solutions' diverging and converging in the search space. Since the invention of the brain storm optimization algorithm in 2011 [5], [6], it has attracted many attentions in the swarm intelligence research community. An analysis of BSO algorithm from the data analytics perspective is introduced in [8]. A comprehensive survey of BSO algorithm was given in [9], and a simple brain storm optimization algorithm with a periodic quantum learning strategy is proposed in [10], just to name a few.

The aim of multimodal optimization is locating multiple global optima in a single run and maintaining these found optima until the end of a run [11]–[13]. Two performance criteria could be used to measure the success of search algorithms. One is whether an optimization algorithm could find all desired optima including global and/or local optima, and the other is whether it can maintain multiple candidate solutions stably over a run [13]. A framework is proposed for locating and tracking multiple optima in [14]. Population diversity of swarm intelligence could be a good way to measure the average distance among candidate solutions, which could reflect the algorithm's ability of solutions maintenance [15].

The principal contributions presented in this work can be summarized as follow:

- The brain storm optimization algorithms have been utilized on solving eight multimodal optimization benchmark problems and seven nonlinear equation system problems.
- The analysis on properties of different variant of BSO algorithms to solve multimodal optimization problems.
- A comparison of different algorithms on different function was given to show the efficiency and effectiveness of the test algorithms.

The remaining of the paper is organized as follows. The basic concept of brain storm optimization algorithms is introduced in Section II. Section III introduces the concepts and performance criteria of multimodal optimization. Experimental study of three BSO variants (original BSO algorithm, BSO-OS-Gaussian, and BSO-OS-Cauchy algorithm), fireworks algorithm, and particle swarm optimization (PSO) algorithms on eight multimodal optimization benchmark functions are conducted in Section IV. An application of multimodal optimization, solving the nonlinear equation system, is given in Section V. Finally, Section VI concludes with some remarks and future research directions.

## II. BRAIN STORM OPTIMIZATION ALGORITHMS

### A. BACKGROUND

The developmental swarm intelligence algorithm is defined that a swarm intelligence algorithm has two kinds of ability: capability learning and capacity developing [2]. The capability learning is a micro level learning ability, which focuses on its actual search from the current solution for single point based optimization algorithms and the current population for population-based swarm intelligence algorithms. While the capacity developing is a macro level learning ability, which focuses on moving the algorithm's search to the area(s) where higher search potential may exist.

### B. ORIGINAL BRAIN STORM OPTIMIZATION

The brain storm optimization algorithm is based on the collective behavior of human being, that is, the brainstorming process [5], [6]. The individuals (solutions) in BSO are converging into several clusters. The best solution in the population will be kept if the newly generated solution at the same index is not better. The new individual can be generated based on the mutation of one or two individuals in clusters. The exploration ability of algorithm is enhanced when the new individual is generated randomly or generated based on the combination of two individuals in two clusters. While the exploitation ability is enhanced when the new individual is generated close to the best solution founded.

It is simple in concept and easy in implementation for the original BSO algorithm [5], [6]. The procedure of BSO algorithm is given in Algorithm 1. There are three strategies in this algorithm: the solution clustering, new individual generation, and selection [16].

**Algorithm 1** The Basic Procedure of the Original Brain Storm Optimization Algorithm

Initialization: Randomly generate $n$ individuals (potential solutions), and evaluate the $n$ individuals;
**while** *not find "good enough" solution or not reach the pre-determined maximum number of iterations* **do**
  **Clustering** operation: Group $n$ individuals into $m$ clusters by a clustering algorithm;
  **New individual generation** operation: Select one or two cluster(s) randomly to generate new individual (solution);
  **Selection** operation: Compare the newly generated individual (solution) and the existing individual (solution) with the same individual index; the better one is kept and recorded as the new individual;
  Evaluate the $n$ individuals (solutions);

**Algorithm 2** The Basic Procedure of the BSO in Objective Space Algorithm

**Initialization**: Generate $n$ individuals (potential solutions) randomly, and evaluate them;
**while** *have not found "good enough" solution or not reached the pre-determined maximum number of iterations* **do**
  **Elitist strategy**: Classify all solutions into two categories: the solutions with better fitness values as elitists and the others as normals;
  **New individual generation** operation: randomly select one or two individuals from elitists or normal to generate new individual;
  **Solution disruption** operation: re-initialize one dimension of a randomly selected individual and update its fitness value accordingly;
  **Selection** operation: The newly generated individual is compared with the existing individual with the same individual index; the better one is kept and recorded as the new individual;
  Evaluate all individuals;

### C. BRAIN STORM OPTIMIZATION IN OBJECTIVE SPACE

In the original BSO algorithm, the computational resources are spending a lot on the clustering strategy at each iteration. To reduce the computational burden, the brain storm optimization in objective space (BSO-OS) algorithm was proposed, and the clustering strategy was replaced by a simple elitist strategy based on the fitness values [7]. The procedure of the BSO in objective space algorithm is given in Algorithm 2.

#### 1) NEW INDIVIDUAL GENERATION OPERATION

The new individual generation strategy is the chief difference between the original BSO and the BSO-OS algorithm. Individuals are clustered into several groups in the original BSO algorithm. While for the BSO-OS algorithm, individuals are classified into two categories, the elitists and the normals, according to their fitness values. The procedure of new individual generation strategy is given in Algorithm 3. Two parameters, probability $p_{elitist}$ and probability $p_{one}$, are used in this strategy.

**Algorithm 3** The Basic Procedure of **New Individual Generation** Operation

**New individual generation**: Select one or two individual(s) randomly to generate new individual;
**if** *random value rand* $<$ *a probability* $p_{elitist}$ **then**
  `/* generate a new individual based on elitists */`
  **if** *random value rand* $<$ *a pre-determined probability* $p_{one}$ **then**
    generate a new individual based on one randomly selected elitist;
  **else**
    two individuals from elitists are randomly selected to generate new individual;
**else** `/* generate a new individual based on normal */`
  **if** *random value rand* $<$ *a pre-determined probability* $p_{one}$ **then**
    generate a new individual based on one randomly selected normal individual;
  **else**
    two individuals from normal are randomly selected to generate new individual;
The newly generated individual is compared with the existing individual with the same individual index, the better one is kept and recorded as the new individual;

The new individuals are generated according to the functions (1) and (2).

$$x^i_{\text{new}} = x^i_{\text{old}} + \xi(t) \times N(\mu, \sigma^2) \tag{1}$$

$$\xi(t) = \text{logsig}(\frac{0.5 \times T - t}{k}) \times \text{rand}() \tag{2}$$

where $x^i_{\text{new}}$ and $x^i_{\text{old}}$ are the $i$th dimension of $\mathbf{x}_{\text{new}}$ and $\mathbf{x}_{\text{old}}$; rand() is a random function to generate uniformly distributed random numbers in the range $[0, 1)$; and the value $\mathbf{x}_{\text{old}}$ is a copy of one individual or the combination of two individuals. The $N(\mu, \sigma^2)$ is a random value that generated with a Gaussian distribution. The parameter $T$ is the maximum number of iterations, $t$ is the current iteration number, $k$ is a coefficient to change logsig() function's slope of the step size function $\xi(t)$, which can be utilized to balance the convergence speed of the algorithm.

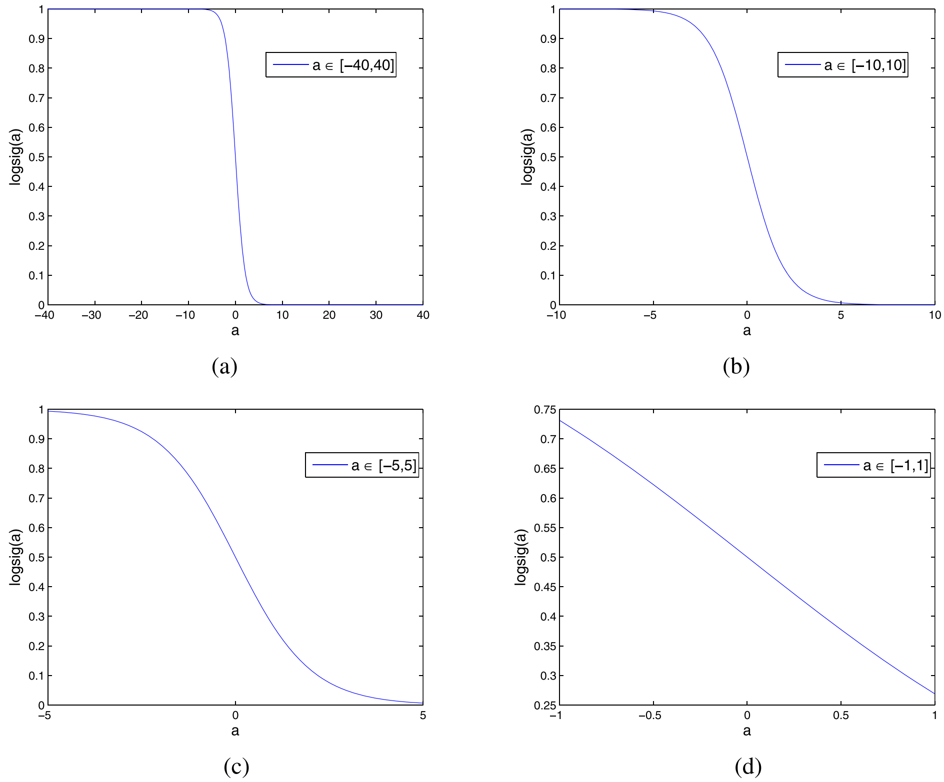



**FIGURE 1.** The function logsig($-a$) with different variable ranges. (a) a ∈ [−40, 40]. (b) a ∈ [−10, 10]. (c) a ∈ [−5, 5]. (d) a ∈ [−1, 1].

#### 2) DISTRIBUTION FUNCTION

The individual is generated by adding a Gaussian random value in Eq. (1). The distribution of this random number could be changed to Cauchy distribution. The BSO-OS algorithm with the Gaussian random values is termed as the BSO-OS-Gaussian, which the BSO-OS algorithm with the Cauchy random values is termed as the BSO-OS-Cauchy algorithm. The new individual generation equation for BSO-OS-Cauchy algorithm is in Eq. (3).

$$x^i_{\text{new}} = x^i_{\text{old}} + \xi(t) \times C(\mu, \sigma^2) \tag{3}$$

#### 3) TRANSFER FUNCTION

A transfer function logsig($a$), which is given in Eq. (4), has been deployed in step size Eq. (2). The parameter $k$ is used to adjust the function's slope. In the original BSO-OS algorithm, the maximum number of iterations $T$ is set as 2000, and the slope $k$ is 25 [7]. Thus, the value $\frac{0.5\times T-t}{k}$ in Eq. (2) is linearly decreased from $\frac{1000-0}{25}$ to $\frac{1000-2000}{25}$, *i.e.*, it is mapped into a range $[-40, 40]$. Figure 1 gives the function logsig() with different variable ranges. It shows that, for function logsig() with variables in range $[-40, 40]$, nearly the half values are close to one, and the other half values are close to zero. In this paper, to have a smooth curve of step size, the value $\frac{0.5\times T-t}{k}$ is mapped into the range $[-10, 10]$.

$$\text{logsig}(a) = \frac{1}{1+exp(-a)} \tag{4}$$

#### 4) BOUNDARY CONSTRAINT

The original BSO algorithm and BSO-OS algorithm lack of strategies to handle the boundary constraints. A new individual may be generated out of the search space. For the conventional boundary handling methods, the solutions were kept inside the feasible search space. The fitness value is only calculated when solutions created in the search space. If a solution exceeds the boundary limit in one dimension at one iteration, that search information will be abandoned. The different boundary constraint handling strategies for particle swarm optimization algorithm have been investigated [17]. Resetting the individual in that dimension is an effective choice for boundary constraint handling. The classic strategy is to set the solution at the boundary when it exceeds the boundary. The equation of this strategy is as follows:

$$x_j = \begin{cases} x_{\max,j} & \text{if } x_j > x_{\max,j} \\ x_{\min,j} & \text{if } x_j < x_{\min,j} \\ x_j & \text{otherwise} \end{cases} \tag{5}$$

This strategy resets solutions in a particular point, *i.e.*, the search boundary, which constrains solutions to explore in the search space limited by a boundary.

## III. MULTIMODAL OPTIMIZATION

### A. MULTIMODAL OPTIMIZATION PROBLEM

Many optimization algorithms are designed for locating a single global solution. Nevertheless, many real-world

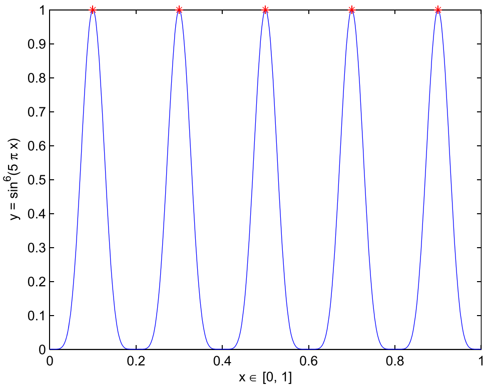



**FIGURE 2.** **An example of a function with equal maxima.**

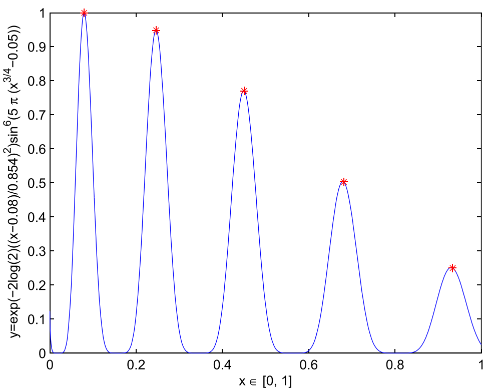



**FIGURE 3.** **An example of function with uneven decreasing maxima.**

problems may have multiple satisfactory solutions exist. The multimodal optimization problem is a function with multiple global/local optimal values.

The equal maxima function, given in Eq. (6), is an example of the multimodal optimization problem,

$$f(x) = \sin^6(5 \times \pi \times x) \tag{6}$$

where $x \in [0, 1]$. From Fig. 2, it can be seen that there are five equal optima for Eq. (6).

The uneven decreasing maxima function in Eq. (7) is another example of the multimodal optimization problem. From Fig. 3, it can be seen that there are one global optimum and four local optima for Eq. (7).

$$f(x) = \exp(-2\log(2)(\frac{x-0.08}{0.854})^2)\sin^6(5\pi(x^{\frac{3}{4}}-0.05)) \tag{7}$$

where $x \in [0, 1]$.

### *B. MULTIMODAL OPTIMIZATION*

For multimodal optimization, the objective is to locate multiple peaks/optima in a single run [18], [19], and to keep these found optima until the end of a run [11]–[13]. An algorithm on solving multimodal optimization problems should have two kinds of abilities: find global/local optima as many as possible and preserve these found solutions until the end of the search.

**Algorithm 4** The Basic Procedure for Determining if All Global Optima Are Found

**Input**: $S_{individuals}$: a set of individuals (candidate solutions) in the population; $\epsilon$: accuracy level; $fit(\mathbf{g}^*)$: the fitness of global optima;
**Output** $S_{solutions}$: a set of best-fit individuals identified as unique solutions; *count*: the number of identified global optima found in the end of a run;
Initialization: $S_{solutions} = \emptyset$, $count = 0$;
**for** *each individual* $\mathbf{x}_i$ *in the candidate solutions set* $S_{individuals}$ **do**
  **if** $|fit(\mathbf{g}^*) - fit(\mathbf{x}_i)| \leq \epsilon$ **then**
    **if** $count == 0$ **then**
      $count += 1$; $S_{solutions} \leftarrow \mathbf{x}_i$;
    **else if** $\mathbf{x}_i \notin S_{solutions}$ **then**
      $count += 1$;
      $S_{solutions} \leftarrow \mathbf{x}_i$;

Different kinds of swarm intelligence and evolutionary computation algorithms have been used to solve multimodal optimization problems, such as species conserving genetic algorithm [20], niching particle swarm optimization with local search [18], adaptive elitist-population based genetic algorithm [21], differential evolution algorithm with neighborhood mutation [19], dynamic fitness sharing mechanism [22], hybrid niching PSO enhanced with recombination-replacement crowding strategy [23], collective animal behavior algorithm [24], sequential niching memetic algorithm [25], the multiobjective optimization techniques [26], [27], and multistart hillclimbing strategy [28], just to name a few. Result visualization is an important issue in multimodal optimization. Similar to the multiobjective optimization, a set of candidate solutions are found for the multimodal optimization. A visualization method for multimodal optimization was proposed to give the distribution information and convergence information in a coordinate plane [29].

### *C. PERFORMANCE CRITERIA*

Two criteria are used to measure the performance of an algorithm. One is the *NPF*, which denotes the total number of global optima found in all runs. The other indicator is the peak ratio (*PR*), which measures the average percentage of all known global optima found over multiple runs [30]. The equations of *PR* calculation are given in Eq. (8).

$$PR = \frac{\sum_{run=1}^{NR} NPF_i}{NKP \times NR} = \frac{NPF}{NKP \times NR} \tag{8}$$

where $NPF_i$ denotes the number of global optima found in the end of the $i$-th run, *NKP* the number of known global optima [30]. The process for determining whether all global optima are found is given in Algorithm 4.

**TABLE 1.** The benchmark problems used in experimental study, where $D$ is the dimension of each problem.

| Func. | Function Name | $D$ | Optima (global/local) |
|---|---|---|---|
| $f_1$ | Five-Uneven-Peak Trap | 1 | 2 / 3 |
| $f_2$ | Equal Maxima | 1 | 5 / 0 |
| $f_3$ | Uneven Decreasing Maxima | 1 | 1 / 4 |
| $f_4$ | Himmelblau | 2 | 4 / 0 |
| $f_5$ | Six-Hump Camel Back | 2 | 2 / 4 |
| $f_6$ | Shubert | 2 / 3 | $D \cdot 3^D$ / many |
| $f_7$ | Vincent | 2 / 3 | $6^D$ / 0 |
| $f_8$ | Modified Rastrigin - All Global Optima | 2 | $\prod_{i=1}^{D} k_i$ / 0 |

**TABLE 2.** The settings of benchmark problems.

| Function | $r$ | Maximum | No. of Global Optima |
|---|---|---|---|
| $f_1$ (1D) | 0.01 | 200.0 | 2 |
| $f_2$ (1D) | 0.01 | 1.0 | 5 |
| $f_3$ (1D) | 0.01 | 1.0 | 1 |
| $f_4$ (2D) | 0.01 | 200.0 | 4 |
| $f_5$ (2D) | 0.5 | 4.126513 | 2 |
| $f_6$ (2D) | 0.5 | 186.73090 | 18 |
| $f_6$ (3D) | 0.5 | 2709.09350 | 81 |
| $f_7$ (2D) | 0.2 | 1.0 | 36 |
| $f_7$ (3D) | 0.2 | 1.0 | 216 |
| $f_8$ (2D) | 0.01 | -2.0 | 12 |
| $f_8$ (16D) | 0.01 | -16.0 | 48 |

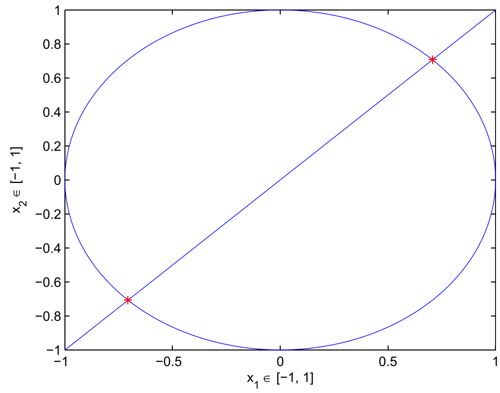



**FIGURE 4.** An example of nonlinear equation systems (nonlinear function $n_1$).

## IV. EXPERIMENTAL STUDY

### A. BENCHMARK FUNCTIONS AND PARAMETERS SETTING

The eight benchmark functions are given in Table 1, and the settings of each function are given in Table 2 [30]. Table 3 gives the parameters of population size and the number of iterations. Two kinds of accuracy levels $\epsilon$, $1.0E-02$ and $1.0E-04$ respectively, are used to reveal the properties of the algorithms' search process. All other parameters of fireworks algorithm are taken from [3]. The parameters of PSO are taken from [31] and [32]. The detailed parameter settings are as follows:

- Original BSO algorithm: $p_{\text{clustering}} = 0.2$, $p_{\text{generation}} = 0.6$, $p_{\text{oneCluster}} = 0.4$ and $p_{\text{twoCluster}} = 0.5$. The parameter $k$ in $k$-means algorithm is 25.
- BSO-OS algorithm: $p_{elitist} = 0.1$, $p_{one} = 0.8$, slope $k = 500$.
- Fireworks algorithm: $a = 0.04$, $b = 0.8$, $\hat{m} = 5$.
- PSO algorithms: $w = 0.72984$, $c_1 = c_2 = 1.496172$.

### B. EXPERIMENTAL RESULTS AND ANALYSIS

The percentages of global optima found by different algorithms on these functions are listed in Table 4 and Table 5. The Table 4 is the results with accuracy level $\epsilon = 1.0E-02$, while the Table 5 is the results with accuracy level $\epsilon = 1.0E-04$. For both accuracy level, the parameter settings and number of iterations are the same, and the difference only occurs on the performance criteria. Results in Table 4 is better than results in Table 5 because some solutions may reach the accuracy level $\epsilon = 1.0E-02$ but not reached an accuracy level $\epsilon = 1.0E-04$.

In general, the BSO-OS-Gaussian algorithm performs best than the BSO-OS-Cauchy algorithm. The BSO-OS-Gaussian algorithm and PSO-vonNeumann algorithm were outperformed than other algorithms among all the test variants. PSO with star structure will converge to one optimum at the end of a run. The PSO with star structure and fireworks algorithm could not find multiple solutions on problems with multiple global optima, and it only finds all solutions on function $f_3$. This is because that function $f_3$ has one global optimum. The BSO-OS algorithm performs better than PSO-Star algorithm but worse than PSO-vonNeumann algorithm for problems with high dimensions. For computational efficiency, the original BSO algorithm has spent the longest running time because the solution clustering strategy has used a large number of computing resources. In two BSO-OS algorithms, the search efficiency has been enhanced by the elitist strategy.

From the results of the experimental study, it could be concluded that: the exploitation ability of BSO-OS variants and original BSO algorithm should be improved. The solutions found by the BSO-OS algorithm are close to the real optima, but the solution accuracy needs to be enhanced. Combining the BSO-OS algorithm with some local search strategies,

**TABLE 3. The population size and number of iterations for BSO, fireworks, and PSO algorithms. "Popu." is the population size, "Iter." is the number of iterations, and the "*MaxFEs*" is the maximum number of fitness evaluation.**

| Function | BSO Algo. Popu. | BSO Algo. Iter. | Fireworks Algo. Popu. | Fireworks Algo. Iter. | PSO Algo. Popu. | PSO Algo. Iter. | $MaxFEs$ |
|---|---|---|---|---|---|---|---|
| $f_1$ (1D) | 500 | 100 | 100 | 500 | 100 | 500 | 5.0E+04 |
| $f_2$ (1D) | 500 | 100 | 100 | 500 | 100 | 500 | 5.0E+04 |
| $f_3$ (1D) | 500 | 100 | 100 | 500 | 100 | 500 | 5.0E+04 |
| $f_4$ (2D) | 500 | 100 | 100 | 500 | 100 | 500 | 5.0E+04 |
| $f_5$ (2D) | 500 | 100 | 100 | 500 | 100 | 500 | 5.0E+04 |
| $f_6$ (2D) | 500 | 400 | 200 | 1000 | 400 | 500 | 2.0E+05 |
| $f_6$ (3D) | 500 | 800 | 400 | 1000 | 800 | 500 | 4.0E+05 |
| $f_7$ (2D) | 500 | 400 | 200 | 1000 | 400 | 500 | 2.0E+05 |
| $f_7$ (3D) | 500 | 800 | 400 | 1000 | 800 | 500 | 4.0E+05 |
| $f_8$ (2D) | 500 | 400 | 200 | 1000 | 400 | 500 | 2.0E+05 |
| $f_8$ (16D) | 500 | 800 | 400 | 1000 | 400 | 1000 | 4.0E+05 |

**TABLE 4. Peak Ratio (PR) of seven algorithms (with accuracy level $\epsilon = 1.0E-02$).**

| Function | $NKP \times NR$ | Original BSO NPF | Original BSO PR | BSO-OS-Gaussian NPF | BSO-OS-Gaussian PR | BSO-OS-Cauchy NPF | BSO-OS-Cauchy PR | Fireworks NPF | Fireworks PR |
|---|---|---|---|---|---|---|---|---|---|
| $f_1$ (1D) | 100 | 87 | 0.87 | **100** | 1.0 | 50 | 0.5 | 50 | 0.5 |
| $f_2$ (1D) | 250 | 243 | 0.972 | **250** | 1.0 | **250** | 1.0 | 57 | 0. 228 |
| $f_3$ (1D) | 50 | **50** | 1.0 | **50** | 1.0 | **50** | 1.0 | **50** | 1.0 |
| $f_4$ (2D) | 200 | 124 | 0.62 | **200** | 1.0 | 190 | 0.95 | 53 | 0.265 |
| $f_5$ (2D) | 100 | 99 | 0.99 | **100** | 1.0 | 91 | 0.91 | 50 | 0.5 |
| $f_6$ (2D) | 900 | 143 | 0.1589 | 666 | 0.74 | 46 | 0.051 | 40 | 0.0444 |
| $f_6$ (3D) | 4050 | 79 | 0.0195 | 23 | 0.0056 | 0 | 0.0 | 1 | 0.0002 |
| $f_7$ (2D) | 1800 | 59 | 0.0328 | 192 | 0.1067 | 108 | 0.06 | 64 | 0.0355 |
| $f_7$ (3D) | 10800 | 55 | 0.0051 | 155 | 0.0144 | 99 | 0.0091 | 53 | 0.0049 |
| $f_8$ (2D) | 600 | 137 | 0.2283 | 597 | 0.995 | 514 | 0.8567 | 50 | 0.0833 |
| $f_8$ (16D) | 2400 | 50 | 0.0208 | 4 | 0.0017 | 0 | 0.0 | 50 | 0.0208 |

| Function | $NKP \times NR$ | PSO-Star NPF | PSO-Star PR | PSO-FourClusters NPF | PSO-FourClusters PR | PSO-vonNeumann NPF | PSO-vonNeumann PR |
|---|---|---|---|---|---|---|---|
| $f_1$ (1D) | 100 | 47 | 0.47 | 84 | 0.84 | 92 | 0.92 |
| $f_2$ (1D) | 250 | 59 | 0.236 | 153 | 0.612 | 236 | 0.944 |
| $f_3$ (1D) | 50 | 49 | 0.98 | 48 | 0.96 | 47 | 0.94 |
| $f_4$ (2D) | 200 | 53 | 0.265 | 121 | 0.605 | 169 | 0.845 |
| $f_5$ (2D) | 100 | 50 | 0.5 | 95 | 0.95 | 100 | 1.0 |
| $f_6$ (2D) | 900 | 50 | 0.0556 | 187 | 0.2078 | 701 | 0.7789 |
| $f_6$ (3D) | 4050 | 50 | 0.0123 | 191 | 0.0472 | 1778 | 0.4390 |
| $f_7$ (2D) | 1800 | 56 | 0.0311 | 163 | 0.0906 | 374 | 0.2078 |
| $f_7$ (3D) | 10800 | 70 | 0.0065 | 194 | 0.0179 | 885 | 0.0819 |
| $f_8$ (2D) | 600 | 53 | 0.0883 | 176 | 0.2933 | 546 | 0.91 |
| $f_8$ (16D) | 2400 | 50 | 0.0208 | 181 | 0.0754 | 535 | 0.2229 |

such as the variable neighborhood search algorithm, could be a good approach to improve the performance of BSO-OS variants for multimodal optimization problems.

## V. NONLINEAR EQUATION SYSTEM

The definition of a nonlinear equation system (NES) could be stated as

$$\begin{cases} e_1(\mathbf{x}) = 0 \\ e_2(\mathbf{x}) = 0 \\ \cdots \\ e_M(\mathbf{x}) = 0 \end{cases} \tag{9}$$

where $\mathbf{x} = [x_1, \ldots, x_D] \in \mathcal{S}$ is the decision vector consisting of $D$ decision variables, $\mathcal{S}$ is the search space. $e_i(\mathbf{x})(i \in \{1, \ldots, M\})$ is the $i$th equation, and $M$ is the number of equations. Generally, A NES contains at least one nonlinear equation [26]. For example, as showed in Fig. 4, there are two solutions, i.e., $[-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}]$ and $[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$ for Eq. (10).

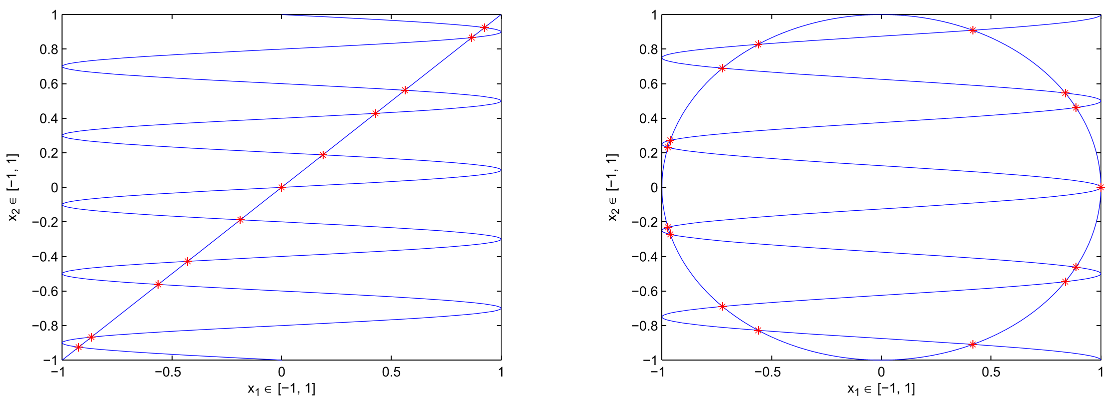
$$\begin{cases} x_1^2 + x_2^2 - 1 = 0 \\ x_1 - x_2 = 0 \end{cases} \tag{10}$$

The solving of nonlinear equation system could be transferred to locate multiple optimal solutions at the same time, i.e. the multimodal optimization. The Eq. (10) could be converted to an optimization function as follows:

**TABLE 5. Peak Ratio (PR) of seven algorithms (with accuracy level $\epsilon = 1.0E-04$).**

| **Function** | $NKP \times NR$ | Original BSO NPF | Original BSO PR | BSO-OS-Gaussian NPF | BSO-OS-Gaussian PR | BSO-OS-Cauchy NPF | BSO-OS-Cauchy PR | Fireworks NPF | Fireworks PR |
|---|---|---|---|---|---|---|---|---|---|
| $f_1$ (1D) | 100 | 86 | 0.86 | **100** | 1.0 | 50 | 0.5 | 50 | 0.5 |
| $f_2$ (1D) | 250 | 242 | 0.968 | **250** | 1.0 | 250 | 1.0 | 50 | 0.2 |
| $f_3$ (1D) | 50 | **50** | 1.0 | **50** | 1.0 | 50 | 1.0 | **50** | 1.0 |
| $f_4$ (2D) | 200 | 119 | 0.595 | 118 | 0.59 | 45 | 0.225 | 22 | 0.11 |
| $f_5$ (2D) | 100 | 95 | 0.95 | **100** | 1.0 | 66 | 0.66 | 50 | 0.5 |
| $f_6$ (2D) | 900 | 102 | 0.1133 | 45 | 0.05 | 11 | 0.1222 | 0 | 0.0 |
| $f_6$ (3D) | 4050 | 67 | 0.0165 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 |
| $f_7$ (2D) | 1800 | 59 | 0.0328 | 192 | 0.1067 | 117 | 0.065 | 51 | 0.0283 |
| $f_7$ (3D) | 10800 | 53 | 0.0049 | 189 | 0.0175 | 72 | 0.0067 | 52 | 0.0048 |
| $f_8$ (2D) | 600 | 114 | 0.19 | 88 | 0.1467 | 65 | 0.1083 | 50 | 0.0833 |
| $f_8$ (16D) | 2400 | 45 | 0.0188 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 |

| **Function** | $NKP \times NR$ | PSO-Star NPF | PSO-Star PR | PSO-FourClusters NPF | PSO-FourClusters PR | PSO-vonNeumann NPF | PSO-vonNeumann PR |
|---|---|---|---|---|---|---|---|
| $f_1$ (1D) | 100 | 48 | 0.48 | 82 | 0.82 | 99 | 0.99 |
| $f_2$ (1D) | 250 | 50 | 0.2 | 144 | 0.576 | 232 | 0.928 |
| $f_3$ (1D) | 50 | 47 | 0.94 | 47 | 0.94 | 50 | 1.0 |
| $f_4$ (2D) | 200 | 50 | 0.25 | 127 | 0.635 | 161 | 0.805 |
| $f_5$ (2D) | 100 | 50 | 0.5 | 91 | 0.91 | 100 | 1.0 |
| $f_6$ (2D) | 900 | 50 | 0.0555 | 181 | 0.2011 | 693 | 0.77 |
| $f_6$ (3D) | 4050 | 50 | 0.0123 | 192 | 0.0474 | 1707 | 0.4215 |
| $f_7$ (2D) | 1800 | 51 | 0.0283 | 161 | 0.0894 | 369 | 0.205 |
| $f_7$ (3D) | 10800 | 51 | 0.0047 | 180 | 0.0167 | 848 | 0.0785 |
| $f_8$ (2D) | 600 | 50 | 0.0833 | 179 | 0.2983 | 544 | 0.9067 |
| $f_8$ (16D) | 2400 | 50 | 0.0208 | 177 | 0.0738 | 502 | 0.2092 |




(a) (b)

**FIGURE 5. Two examples of nonlinear equation systems.**

Seven test instances of NES, which listed in Table 6, are used in the experimental study to test the effectiveness of different algorithms on solving nonlinear equation systems. Three functions ($n_1$, $n_3$, $n_4$) have 2 dimensions in the search space. Fig. 4 gives an illustration of nonlinear function $n_1$, and Fig. 5 gives illustrations of nonlinear function $n_3$ and $n_4$, respectively. The distribution of optima is not the same for different functions. Function $n_4$ have two optima pairs that an optimum is very close to another. This may increase the difficulty of search that solutions are easily stuck in one optimum. Functions $n_2$ and $n_7$ have 20 variables in the search space, i.e. 20 dimensions. For other functions, $n_5$ has 3 dimensions and $n_6$ has 6 dimensions in the search space. The peak ratio measure is not applicable for function $n_5$, $n_6$, and $n_7$ due to these functions have an infinite number of optima.

The population size and number of iterations are given in Table 7. Two accuracy levels $\epsilon$ are $1.0E-02$ and $1.0E-04$ respectively. The results of algorithms on solving seven NES functions are given in Table 8 and Table 9, respectively. The Table 8 is the results with accuracy level $\epsilon = 1.0E-02$, while the Table 9 is the results with accuracy level $\epsilon = 1.0E-04$.

In general, BSO-OS-Gaussian and PSO with von Neumann structure perform better than other algorithms. From the results, these two algorithms have found all solutions

**TABLE 6.** Seven nonlinear equation systems used in the experimental study, where $D$ is the dimension of each problem, "Lin." is number of linear equations, "Non." is number of nonlinear equations, and "Optima" is number of the optimal solutions.

| | Function | $D$ | Search Space | Lin./ Non. | Optima |
|---|---|---|---|---|---|
| $n_1$ | $\begin{cases} x_1^2 + x_2^2 - 1 = 0 \\ x_1 - x_2 = 0 \end{cases}$ | 2 | $[-1,1]^D$ | 1/1 | 2 |
| $n_2$ | $\begin{cases} \sum_{i=1}^{D} x_i^2 - 1 = 0 \\ \lvert x_1 - x_2 \rvert + \sum_{i=3}^{D} x_i^2 = 0 \end{cases}$ | 20 | $[-1,1]^D$ | 0/2 | 2 |
| $n_3$ | $\begin{cases} x_1 - \sin(5 \times \pi \times x_2) = 0 \\ x_1 - x_2 = 0 \end{cases}$ | 2 | $[-1,1]^D$ | 1/1 | 11 |
| $n_4$ | $\begin{cases} x_1 - \cos(4 \times \pi \times x_2) = 0 \\ x_1^2 + x_2^2 = 1 \end{cases}$ | 2 | $[-1,1]^D$ | 0/2 | 15 |
| $n_5$ | $\begin{cases} x_1 + x_2 + x_3 - 1 = 0 \\ x_1 - x_2^3 = 0 \end{cases}$ | 3 | $[-1,1]^D$ | 1/1 | infinite |
| $n_6$ | $\begin{cases} x_1^2 + x_3^2 = 1 \\ x_2^2 + x_4^2 = 1 \\ x_1^3 x_5 + x_2^3 x_6 = 0 \\ x_3^3 x_5 + x_4^3 x_6 = 0 \\ x_1 x_3^2 x_5 + x_2 x_4^2 x_6 = 0 \\ x_1^2 x_3 x_5 + x_2^2 x_4 x_6 = 0 \end{cases}$ | 6 | $[-1,1]^D$ | 0/6 | infinite |
| $n_7$ | $\begin{cases} (x_k + \sum_{i=1}^{D-k-1} x_i x_{i+k}) x_D - c_k = 0 \\ 1 \leq k \leq D-1 \\ \sum_{i=1}^{D-1} x_i + 1 = 0 \end{cases}$ $c_k = 0, \forall k \in \{1, \ldots, D-1\}$ | 20 | $[-1,1]^D$ | 1/19 | infinite |

**TABLE 7.** The population size and number of iterations for BSO, fireworks, and PSO algorithms.

| Function | BSO Algo. Popu. | BSO Algo. Iter. | Fireworks Algo. Popu. | Fireworks Algo. Iter. | PSO Algo. Popu. | PSO Algo. Iter. | $MaxFEs$ |
|---|---|---|---|---|---|---|---|
| $n_1$ (2D) | 250 | 200 | 100 | 500 | 200 | 250 | 5.0E+04 |
| $n_2$ (20D) | 500 | 800 | 400 | 1000 | 800 | 500 | 4.0E+05 |
| $n_3$ (2D) | 250 | 200 | 100 | 500 | 200 | 250 | 5.0E+04 |
| $n_4$ (2D) | 250 | 200 | 100 | 500 | 200 | 250 | 5.0E+04 |
| $n_5$ (3D) | 500 | 800 | 400 | 1000 | 800 | 500 | 4.0E+05 |
| $n_6$ (6D) | 500 | 800 | 400 | 1000 | 800 | 500 | 4.0E+05 |
| $n_7$ (20D) | 500 | 800 | 400 | 1000 | 800 | 500 | 4.0E+05 |

**TABLE 8.** Results of the seven algorithms on solving seven nonlinear equation systems (with accuracy level $\epsilon = 1.0E-02$).

| Function | | Original BSO | BSO-OS -Gaussian | BSO-OS -Cauchy | Fireworks | PSO -Star | PSO -FourClusters | PSO -vonNeumann |
|---|---|---|---|---|---|---|---|---|
| $n_1$ (2D) | $NPF$ | 0 | 99 | 69 | 47 | 50 | 95 | **100** |
| | $PR$ | 0.0 | 0.99 | 0.69 | 0.47 | 0.5 | 0.95 | 1.0 |
| $n_2$ (20D) | $NPF$ | 0 | 44 | 0 | 50 | 50 | 95 | **100** |
| | $PR$ | 0.0 | 0.44 | 0 | 0.5 | 0.5 | 0.95 | 1.0 |
| $n_3$ (2D) | $NPF$ | 100 | **445** | 174 | 48 | 51 | 181 | 421 |
| | $PR$ | 0.1819 | 0.8091 | 0.3164 | 0.0873 | 0.0927 | 0.3291 | 0.7655 |
| $n_4$ (2D) | $NPF$ | 1 | 96 | 50 | 36 | 50 | 162 | **471** |
| | $PR$ | 0.0013 | 0.128 | 0.0667 | 0.048 | 0.0667 | 0.216 | 0.628 |
| $n_5$ (3D) | $NPF$ | 1 | 1107 | 565 | 53 | 189 | 366 | **2852** |
| $n_6$ (6D) | $NPF$ | 0 | **11961** | 324 | 53 | 270 | 503 | 3655 |
| $n_7$ (20D) | $NPF$ | 0 | **11024** | 782 | 50 | 4028 | 4483 | 8364 |

for function $n_1$ and $n_2$, most solutions for function $n_3$ and $n_4$, and many solutions for function $n_5$, $n_6$, and $n_7$. It can be concluded that the BSO-OS-Gaussian and PSO with von Neumann structure are two good search algorithms for solving NES functions. The original BSO algorithm performs worst among all algorithms. The original BSO algorithm has

**TABLE 9. Results of the seven algorithms on solving seven nonlinear equation systems (with accuracy level $\epsilon = 1.0E-04$).**

| Function | | Original BSO | BSO-OS -Gaussian | BSO-OS -Cauchy | Fireworks | PSO -Star | PSO -FourClusters | PSO -vonNeumann |
|---|---|---|---|---|---|---|---|---|
| $n_1$ (2D) | $NPF$ | 0 | 13 | 3 | 2 | 50 | 93 | **100** |
| | $PR$ | 0.0 | 0.13 | 0.03 | 0.02 | 0.5 | 0.93 | 1.0 |
| $n_2$ (20D) | $NPF$ | 0 | 0 | 0 | 0 | 50 | 91 | **100** |
| | $PR$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 0.91 | 1.0 |
| $n_3$ (2D) | $NPF$ | 0 | 10 | 1 | 27 | 50 | 175 | 400 |
| | $PR$ | 0.0 | 0.0182 | 0.0018 | 0.0491 | 0.0909 | 0.3182 | 0.7273 |
| $n_4$ (2D) | $NPF$ | 0 | 50 | 49 | 0 | 50 | 162 | 457 |
| | $PR$ | 0.0 | 0.0667 | 0.0653 | 0.0 | 0.0667 | 0.216 | 0.6093 |
| $n_5$ (3D) | $NPF$ | 0 | 87 | 4 | 22 | 50 | 179 | 2546 |
| $n_6$ (6D) | $NPF$ | 0 | 0 | 0 | 17 | 52 | 188 | 2184 |
| $n_7$ (20D) | $NPF$ | 0 | 10 | 0 | 10 | 62 | 188 | 2161 |

a slow convergence, thus, all solutions could not reach the certain accuracy level after all iterations. Results of BSO-OS and fireworks algorithm with accuracy level $\epsilon = 1.0E-02$ are significantly better than results with accuracy level $\epsilon = 1.0E-04$. This indicates that algorithms have found the areas that may contain optima, but the convergence speed is not fast enough to locate the optima with certain accuracy level. In other words, algorithms should enhance its exploitation ability during the search.

## VI. CONCLUSIONS

On solving multimodal optimization problems, the aim is to locate multiple optima/peaks in a single run and to maintain these found optima until the end of a run. In this paper, three variants of brain storm optimization algorithms have been utilized to solve multimodal optimization problems. The performance and effectiveness of different algorithms on solving multimodal optimization problems have been validated. The experimental tests are conducted on eight benchmark functions and seven nonlinear equation system (NES) problems. Based on the experimental results, the conclusions could be made that the BSO-OS algorithm performs better than FWA and PSO-Star algorithm but worse than PSO-vonNeumann algorithm for multimodal optimization problems in several problems.

The exploitation ability of BSO-OS algorithm should be enhanced on solving multimodal optimization problems. Combining the BSO-OS algorithm with some local search strategies, such as the variable neighborhood search algorithm, may be a good approach to improve the search performance of different variants of BSO-OS algorithm. In addition, to obtain good performances on multimodal optimization problems, a swarm intelligence algorithm needs to balance its global search ability and solutions maintenance ability simultaneously.

## REFERENCES


[1] Y.-S. Ong, M. H. Lim, and X. Chen, "Memetic computation—Past, present & future [research Frontier]," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, May 2010.

[2] Y. Shi, "Developmental swarm intelligence: Developmental learning perspective of swarm intelligence algorithms," *Int. J. Swarm Intell. Res.*, vol. 5, no. 1, pp. 36–54, Jan./Mar. 2014.

[3] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence* (Lecture Notes in Computer Science), Y. Tan, Y. Shi, and K. C. Tan, Eds. Berlin, Germany: Springer-Verlag, 2010, pp. 355–364.

[4] Y. Tan, *Fireworks Algorithm: A Novel Swarm Intelligence Optimization Method*. Berlin, Germany: Springer, 2015.

[5] Y. Shi, "Brain storm optimization algorithm," in *Advances in Swarm Intelligence* (Lecture Notes in Computer Science), Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds. Berlin, Germany: Springer, 2011, pp. 303–309.

[6] Y. Shi, "An optimization algorithm based on brainstorming process," *Int. J. Swarm Intell. Res.*, vol. 2, no. 4, pp. 35–62, Oct./Dec. 2011.

[7] Y. Shi, "Brain storm optimization algorithm in objective space," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 1227–1234.

[8] 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.

[9] S. Cheng *et al.*, "A comprehensive survey of brain storm optimization algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1637–1644.

[10] Z. Song, J. Peng, C. Li, and P. X. Liu, "A simple brain storm optimization algorithm with a periodic quantum learning strategy," *IEEE Access*, to be published. [Online]. Available: http://ieeexplore.ieee.org/document/8119509/, doi: 10.1109/ACCESS.2017.2776958.

[11] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 440–458, Aug. 2006.

[12] J. Rönkkönen, "Continuous multimodal global optimization with differential evolution-based methods," Ph.D. dissertation, Dept. Inf. Technol., Lappeenranta Univ. Technol., Lappeenranta, Finland, Dec. 2009.

[13] X. Li, "Niching without Niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.

[14] C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, and S. Yang, "An adaptive multipopulation framework for locating and tracking multiple optima," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 590–605, Aug. 2016.

[15] S. Cheng, J. Chen, Q. Qin, and Y. Shi, "Population diversity of particle swarm optimisation algorithms for solving multimodal optimisation problems," *Int. J. Comput. Sci. Eng.*, 2016. [Online]. Available: http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijcse

[16] S. Cheng, Y. Shi, Q. Qin, and S. Gao, "Solution clustering analysis in brain storm optimization algorithm," in *Proc. IEEE Symp. Swarm Intell. (SIS)*, Singapore, Apr. 2013, pp. 111–118.

[17] S. Cheng, Y. Shi, and Q. Qin, "Experimental study on boundary constraints handling in particle swarm optimization: From population diversity perspective," *Int. J. Swarm Intell. Res.*, vol. 2, no. 3, pp. 43–69, 2011.

[18] B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Inf. Sci.*, vol. 197, pp. 131–143, Aug. 2012.

[19] B.-Y. Qu, P. N. Suganthan, and J.-J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

[20] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, 2002.

[21] K.-S. Leung and Y. Liang, "Adaptive elitist-population based genetic algorithm for multimodal function optimization," in *Proc. Conf. Companion Genet. Evol. Comput. Companion*, 2003, pp. 1160–1171.

[22] A. D. Cioppa, C. D. Stefano, and A. Marcelli, "Where are the niches? Dynamic fitness sharing," *IEEE Trans. Evol. Comput.*, vol. 11, no. 4, pp. 453–465, Aug. 2007.

[23] M. Li, D. Lin, and J. Kou, "A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization," *Appl. Soft Comput.*, vol. 12, no. 3, pp. 975–987, 2012.

[24] E. Cuevas and M. Gonzalez, "An optimization algorithm for multimodal functions inspired by collective animal behavior," *Soft Comput.*, vol. 17, no. 3, pp. 489–502, 2013.

[25] J. E. Vitela and O. Castaños, "A sequential niching memetic algorithm for continuous multimodal function optimization," *Appl. Math. Comput.*, vol. 218, no. 17, pp. 8242–8259, 2012.

[26] W. Song, Y. Wang, H.-X. Li, and Z. Cai, "Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 414–431, Jun. 2015.

[27] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.

[28] F. G. Lobo and M. Bazargani, "When hillclimbers beat genetic algorithms in multimodal optimization," in *Proc. Conf. Companion Genetic Evol. Comput. Companion*, 2015, pp. 1421–1422.

[29] R. Cheng, M. Li, and X. Yao, "Parallel peaks: A visualization method for benchmark studies of multimodal optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 263–270.

[30] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," School Comput. Sci. IT, RMIT Univ., Melbourne, VIC, Australia, Tech. Rep., 2013.

[31] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2000, pp. 84–88.

[32] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2007, pp. 120–127.

**SHI CHENG** (M'14) received the bachelor's degree in mechanical and electrical engineering from Xiamen University, Xiamen, in 2005, the master's degree in software engineering from Beihang University, Beijing, China, in 2008, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K., in 2013. He is currently a Lecturer with the School of Computer Science, Shaanxi Normal University, China. His current research interests include swarm intelligence, multiobjective optimization, and data mining techniques and their applications.

**JUNFENG CHEN** received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2011. She is currently an Associate Professor with the College of IOT Engineering, Hohai University, Changzhou, China. Her research interests include swarm intelligence, artificial intelligence with uncertainty, and big data analytics.

**XIUJUAN LEI** received the Ph.D. degree from Northwestern Polytechnical University, Xi'an, China, in 2005. She is currently a Professor with the School of Computer Science, Shaanxi Normal University, Xi'an, China. Her research interests include swarm intelligence, bioinformatics, and big data analytics.

**YUHUI SHI** (F'16) received the Ph.D. degree in electronic engineering from Southeast University, Nanjing, China, in 1992. He is currently a Chair Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Before joining the Southern University of Science and Technology, he was with Electronic Data Systems Corporation, Indianapolis, IN, USA. His main research interests include the areas of computational intelligence techniques (including swarm intelligence) and their applications. He is an Editor-in-Chief of the *International Journal of Swarm Intelligence Research*.