IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A Brain Storm Optimization with Multi-information Interactions for Global Optimization Problems

CHUNQUAN LI[1,2], ZHENSHOU SONG[1], JINGHUI FAN[1], QIANGQIANG CHENG[3], and PETER X. LIU[1,2], Senior, IEEE

[1]School of Information Engineering, Nanchang University, Nanchang 330029, CHINA
[2]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, CANADA
[3]School of Measuring and Optical Engineering, Hangkong University, Nanchang 330036, CHINA

Corresponding author: Jinghui Fan (e-mail: jinghuiFan@ncu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grants 61503177 and 81660299, by the China Scholarship Council under the State Scholarship Fund (CSC No. 201606825041), by the Science and Technology Department of Jiangxi Province of China under Grants 20161ACB21007, 20171BBE50071, and 20171BAB202033, and by the Education Department of Jiangxi province of China under Grants GJJ14228 and GJJ150197.

ABSTRACT The original BSO fails to consider some potential information interactions in its individual update pattern, causing the premature convergence for complex problems. To address this problem, we propose a BSO algorithm with multi-information interactions (MIIBSO). First, a multi-information interaction (MII) strategy is developed, thoroughly considering various information interactions among individuals. Specially, this strategy contains three new MII patterns. The first two patterns aim to reinforce information interaction capability between individuals. The third pattern provides interactions between the corresponding dimensions of different individuals. The collaboration of the above three patterns is established by an individual stagnation feedback (ISF) mechanism, contributing to preserve the diversity of the population and enhance the global search capability for MIIBSO. Second, a random grouping (RG) strategy is introduced to replace both the K-means algorithm and cluster center disruption of the original BSO algorithm, further enhancing the information interaction capability and reducing the computational cost of MIIBSO. Finally, a dynamic difference step-size (DDS), which can offer individual feedback information and improve search range, is designed to achieve an effective balance between global and local search capability for MIIBSO. By combining the MII strategy, RG, and DDS, MIIBSO achieves the effective improvement in the global search ability, convergence speed, and computational cost. MIIBSO is compared with 11 BSO algorithms and five other algorithms on the CEC2013 test suit. The results confirm that MIIBSO obtains the best global search capability and convergence speed amongst the 17 algorithms.

INDEX TERMS Brain storm optimization (BSO), Multi-information Interaction, Swarm intelligence, Global optimization.

I. INTRODUCTION

In the last few decades, various swarm intelligence techniques, such as particle swarm optimization (PSO) [1], [2], difference evolution (DE) [3], ant colony optimization (ACO) [4], artificial bee colony (ABC) [5], and fruit fly optimization (FOA) [6] have been developed to handle a variety of optimization issues in many fields. Compared with the canonical algorithms depending on the gradient computation, these swarm intelligence algorithms use the simple strategy that mimics cooperating and interacting behaviors in birds flocking and fish schooling. This enables the individuals of these swarm intelligence algorithms to move toward better positions and search globally optimal solutions. Owing to such a strategy, these swarm intelligence algorithms can offer the excellent performance in handling some complicated optimal issues, especially nonlinear and discontinuous optimal issues.

Inspired by the human brainstorming process, Shi proposed a novel swarm intelligence technique, termed brain storm optimization (BSO), in 2011 [7], [8]. Unlike the aforesaid swarm intelligent algorithms that mimic the behaviors of animal, the BSO algorithm simulates the human brainstorming process, where a group of people gathers together to spur some new ideas for solving the thorny issues.

For the BSO algorithm, each solution from the search space can be treated as an idea of the brainstorming process. For each generation of the iterative evolution process, these ideas are all gathered into different groups using the clustering operation. The best idea on each group serves as the corresponding cluster center. Besides, updating each idea can be implemented by combining the Gaussian factor or merging with the ideas of other clusters. In brief, the BSO algorithm includes five main processes: individual initialization, individual clustering, cluster center disruption, individual update, and individual selection.

Intuitively, the BSO algorithm should be superior to other

swarm intelligence algorithms because it simulates the creative behaviors from the human beings, the most intelligent behaviors in the world, rather than the simple swarm behaviors of animals such as birds flocking and fish schooling behaviors [7], [8]. As a promising intelligent optimization technique, BSO has been successfully used to solve a variety of scientific and engineering problems [9-17].

However, as a young algorithm, BSO still needs to be further improved and optimized in many aspects such as the algorithm computational efficiency, convergence speed, and global search capability.

In the original BSO algorithm, the K-means clustering method is used as the individual clustering strategy. However, such a strategy has the high time complexity. Consequently, to improve the computational efficiency, Zhan et al. [18] proposed a modified brain storm optimization (MBSO) algorithm, which uses a simple grouping method (SGM) to replace the K-means method. Subsequently, Qiu and Duan [19] used the fitness values of individuals rather than distances between individuals as a clustering standard. Cao et al. [20] presented a random grouping BSO (RGBSO) algorithm, where the random grouping strategy replaces the k-means clustering method. This reduces the computational burden and has more opportunities to discover promising solutions. Zhu and Shi [21] adopted k-medians clustering algorithm to replace the K-means clustering algorithm, which can avoid the cluster centers being effortlessly influenced by extreme values and improve the clustering efficiency. Chen et al. [22] presented a modified affinity propagation (AP) clustering strategy in place of the K-means clustering method, which can adaptively vary the number of clusters during the iterative search process. Cao et al. [23] also adopted a dynamic K-means clustering method. Its fundamental principle is that the K-means clustering method is periodically carried out to improve the exploration capability and reduce the computational burden.

Furthermore, some significant efforts were made to perfect the individual update strategy. Zhou et al. [24] presented a modified step-size with a new individual update based on a batch-mode to escape from the local optima and accelerate the convergence speed. Li and Duan [25] invented a simple brain storm optimization (SBSO) algorithm by simplifying four individual-generating operators into one operator to obtain an improvement in the convergence speed. Owing to the difference strategy containing some feedback information from updating individual, it can play an important role in the improvement of the step-size function in the original BSO algorithm. Consequently, Zhan et al. [18] developed a new idea difference strategy (IDS) as the new step-size function, avoiding the individuals to become trapped in local optima. Similarly, Sun and Duan [26] designed three closed-loop brain storm optimization (CLBSO) algorithms, which utilize difference evolution strategies as new step-size functions to update new individuals and improve the convergence performance. Cao et al. [27] adopted the differential evolution strategy with a new step-size technique to update new individuals and achieve an effective balance between exploration and exploitation.

Moreover, other techniques were also introduced to the original BSO algorithm to improve individual update and enhance the search efficiency. In [19], both the chaotic search technique and the probability update strategy were used to improve the individual update of the original BSO algorithm and avoid the local optimum. Similarly, Yang and Shi [28] introduced the chaotic search technique to the original BSO algorithm. Although both references [19] and [28] adopt the chaotic search technique, the differences between them are that the former uses the chaotic search technique for the best individual in the entire swarm, however, the latter applies this technique to each individual. In [29], the teaching-learning-based algorithm was incorporated into the BSO algorithm to obtain a self-evolving feature in the entire iterative process. Duan et al. [30] developed a predator-prey BSO (PPBSO), where the predator-prey strategy was employed to update new individuals and avoid local optima. Duan and Li [31] proposed a quantum-behaved BSO (QBSO) algorithm, where the quantum behavior is used to update new individuals and improve the diversity of the population. Song et al. [32] proposed a simple BSO algorithm with a periodic quantum learning strategy (SBSO-PQLS), including three new strategies developed to improve the global search and decrease the computation cost. Yang et al. [33] presented an advanced discussion mechanism-based brain storm optimization (ADMBSO), which adopts inter-cluster and intra-cluster discussing strategies to improve the individual update strategy and achieve the beneficial balance between the exploration and exploitation. Jia et al. [34] incorporated the simulated annealing (SA) technique into the original BSO algorithm to refine the search space and avoid the local optima. Mohammed El-Abd [35] proposed a Global-best BSO (GBSO) algorithm by combining a global-best with dimension updates and fitness-based cluster.

In addition, Cheng et al. [36] adopted two kinds of partial reinitializing strategies to improve the population diversity and enhance the global search capability of the algorithm. Mohammed El-Abd [37] presented an improved RGBSO (IRGBSO) algorithm, where re-initialized individuals combined with adaptive step size are used to enhance the RGBSO algorithm performance. Zhou et al. [38] investigated the convergence performance of the original BSO algorithm by using the Markov model. A survey of various BSO variants, including their historical developments and the state-of-the-art was detailed in [39].

From the BSO variants mentioned above, various investigations have been done for improving the individual clustering and update. However, for most of the BSO variants, their update strategies failed to consider thoroughly various information interaction patterns between individuals during the process of new individuals generated. For instance, the individual update strategy of the original BSO algorithm did not consider information interaction patterns between individuals from one cluster, and between the corresponding dimensions of individuals. Although the individual update strategy of ADMBSO has considered the information interaction between individuals from one cluster and two clusters, an interaction pattern between the corresponding dimensions of individuals was also ignored. GBSO utilized a dimension information strategy to carry out individual interactions [35]. However, it failed to consider the individual interactions like that of the original BSO or ADMBSO algorithm. The interaction patterns ignored may lead to the absence of some beneficial solution information. For this reason,

the BSO algorithms may become trapped in local optima while dealing with complex optimization issues [40].

To address the aforesaid issue, this paper proposes a novel BSO variant, termed BSO algorithm with multi-information interaction (MIIBSO). For the individual update strategy in MIIBSO, we develop a new multi-information (MII) strategy including three new MII patterns that focus on: MII pattern between individuals from one cluster (MII pattern I), MII pattern between individuals from two clusters (MII pattern II), and MII pattern between the corresponding dimensions of individuals (MII pattern III). The first two patterns are improved based on the information interaction patterns of the original BSO algorithm to enrich the diversity of the population and simplify the redundant information interaction for MIIBSO. Inspired by the dimensional update pattern of CLPSO [41], we propose the MII pattern III to further enrich the diversity of the population and improve the global search capability. Furthermore, from the point of view of computational cost and convergence performance, we establish an individual stagnation feedback (ISF) mechanism to effectively couple three MII patterns, which can improve the global and local search capability. In addition, for the individual clustering strategy, we introduce a random grouping (RG) strategy [20] to replace both the K-means clustering algorithm and the cluster center disruption of the original BSO algorithm, which is conducive to further enrich the information interactions and decrease the computational cost of MIIBSO. Finally, we propose a dynamic difference step-size (DDS) function that can provide individual feedback information and increase search range to implement an effective balance between global and local search capability. Therefore, the proposed MIIBSO algorithm can effectively enhance the global search capability, accelerate the convergence speed, and decrease computational cost.

The remainder of this paper is organized as follows. Section II introduces the original BSO algorithm. Section III develops the MIIBSO algorithm in detail. Section IV executes extensive experiments to evaluate the proposed MIIBSO algorithm. Section V provides the discussion with future works on MIIBSO. Finally, conclusions on the proposed algorithm are given in Section VI.

II. ORIGINAL BSO ALGORITHM

Since brainstorming was introduced by Osborn in 1939 [42], it has been extensively used to encourage creative thinking. Enlightened by the brainstorming process, Shi proposed a new population-based optimization algorithm, imitating such a process that a set of people with as various backgrounds as possible congregate together and spontaneously contribute their best ideas for solving a specific issue. The original BSO algorithm chiefly involves five processes that focus on individual initialization, individual clustering, cluster center disruption, individual update, and individual selection, described separately as follows.

A. INDIVIDUAL INITIALIZATION

In the original BSO algorithm, we assume that there are N individuals. Everyone, called an idea of the brainstorming process, characterizes a candidate solution to a specific issue, expressed as a vector $X_i = [x_{i1}, x_{i2}, \cdots, x_{iD}]$, $i \in \{1, 2, \cdots, N\}$. Here, D means the dimension of a solution space. The jth dimension of the idea X_i is defined as scalar x_{ij}, $j \in \{1, 2, \cdots, D\}$ in the boundary range $[x_{j\_min}, x_{j\_max}]$, where x_{j_min} and x_{j_max} are described as the minimum and maximum boundaries of the jth dimension of the candidate solution, respectively. As a result, x_{ij} can be randomly initialized as

$$x_{ij} = x_{j\_min} + r_j\left(x_{j\_max} - x_{j\_min}\right), \quad (1)$$

where r_j is a random number uniformly distributed in the range $[0, 1]$. Subsequently, the fitness value of each idea X_i, $i \in \{1, 2, \cdots, N\}$ can be computed using a fitness function $f[X_i(k)]$.

B. INDIVIDUAL INITIALIZATION CLUSTERING STRATEGY

In each iteration, we separate all the N ideas of the entire swarm into M different clusters via the K-means clustering strategy [7]. For each cluster $m \in \{1, 2, \cdots, M\}$, an idea with the best fitness value is designated as the cluster center $G_m = [g_{m1}, g_{m2}, \cdots, g_{mD}]$. All M cluster centers can be written as $\{G_1, G_2, \cdots, G_M\}$. The individual clustering strategy aims at converging ideas into several small different search regions and refines these candidate solutions.

C. CLUSTER-CENTER DISRUPTION

In each iterative process, we first randomly select a cluster center $G_\tau = [g_{\tau 1}, g_{\tau 2}, \cdots, g_{\tau D}]$, $\tau \in \{1, 2, \cdots, M\}$ from M cluster centers $\{G_2, G_2, \cdots, G_M\}$. Then, we create a new idea $Q = [q_1, q_2, \cdots, q_D]$ where each q_d, $d \in \{1, 2, \cdots, D\}$ can be generated according to (1). Finally, the cluster center G_τ is replaced by the new random idea Q when $r_0 < p_{r0}$ is true; otherwise, it is not. Here, r_0 is a uniformly distributed random number in the range $[0, 1]$, and p_{r0} is the pre-determined probability in the original BSO algorithm. The cluster center disruption is intended to diverge the cluster centers to search more potential solution regions and improve the global search capability.

D. INDIVIDUAL UPDATE STRATEGY

In each iterative process, the original BSO algorithm has two individual update patterns including individual update patterns I and II. In the former, an individual is updated, or a new idea is created by one "old" idea from one cluster; the latter updates an individual or creates a new idea by two "old" idea from two clusters. Their implementations are illustrated as follows.

(1) Individual Update Pattern I

First, a new idea $X_i = [x_{i1}, x_{i2}, \cdots, x_{iD}]$, $i \in \{1, 2, \cdots, N\}$ is created by one "old" idea $U_i = [u_{i1}, u_{i2}, \cdots, u_{iD}]$ from one cluster of M clusters if $r_1 < p_{r1}$ is true as follows:

$$x_{ij} = u_{ij} + \eta_j(\mu, \sigma)\xi_j(\mathrm{k}), \; r_1 < p_{r1} \quad (2)$$

where x_{ij} and u_{ij} are the jth dimension of the new idea X_i and the "old" idea U_i, respectively; η_j is a Gaussian random number, μ and σ are its mean and variance, respectively, and ξ_j represents the jth dimension of the step size function for balancing the global search and local search capability; r_1 and p_{r1} represents a uniform distribution random number and pre-determined probability, respectively.

If $r_{11} < p_{r11}$ is true, the "old" idea U_i is equivalent to a cluster center G_m, $m \in \{1, 2, \cdots, M\}$ that is randomly selected from M cluster centers $\{G_2, G_2, \cdots, G_M\}$; otherwise, the "old"

idea U_i is equivalent to an idea $\Psi_i = [\psi_{m1}, \psi_{m2}, \cdots, \psi_{mD}]$ that is randomly selected from cluster m. Here, r_{11} and p_{r11} are a uniform distribution random number and a predetermined probability, respectively. Therefore, U_i is formulated as

$$u_{ij} = \begin{cases} g_{mj}, & r_{11} < p_{r11} \\ \psi_{mj}, & r_{11} \geq p_{r11}, \end{cases} \quad (3)$$

where g_{mj} and ψ_{mj} are the jth dimension of the center G_m and the idea Ψ_i, respectively. In addition, a probability p_C is employed to select the cluster center G_m via roulette wheel selection [7] as

$$p_C = f[G_m] / \sum_{i=1}^{M} f[G_i], \quad (4)$$

where $f[G_m]$ and $\sum_{i=1}^{M} f[G_i]$ are the fitness value of the center G_m and the sum of the fitness values of the M cluster centers $\{G_2, G_2, \cdots, G_M\}$, respectively.

(2) Individual Update Pattern II

Subsequently, a new idea $X_i = [x_{i1}, x_{i2}, \cdots, x_{iD}]$, $i \in \{1, 2, \cdots, N\}$ is also created by two "old" ideas $U_i = [u_{i1}, u_{i2}, \cdots, u_{iD}]$ and $V_i = [v_{i1}, v_{i2}, \cdots, v_{iD}]$ from two different cluster of M clusters if $r_1 \geq p_{r1}$ is true as follows:

$$x_{ij} = r_j u_{ij} + (1 - r_j) v_{ij} + \eta_j(\mu, \sigma)\xi_j(\mathrm{k}), \; r_1 \geq p_{r1} \quad (5)$$

where x_{ij}, u_{ij}, and v_{ij} are the jth dimension of the new idea X_i, the "old" idea U_i and the "old" idea V_i, respectively; r_j is a random number uniformly distributed in the range $[0, 1]$.

The "old" ideas U_i and V_i are equivalent to different cluster centers G_m and G_n, $m \neq n, \; m, n \in \{1, 2, \cdots, M\}$, respectively, when $r_{12} < p_{r12}$ is true; otherwise, the "old" ideas U_i and V_i are equivalent to two ideas $\Psi_i = [\psi_{m1}, \psi_{m2}, \cdots, \psi_{mD}]$ and $\Phi_i = [\phi_{n1}, \phi_{n2}, \cdots, \phi_{nD}]$, respectively. Here, G_m and G_n are randomly selected from M cluster centers $\{G_1, G_2, \cdots, G_M\}$, Ψ_i and Φ_i are randomly selected from two different clusters m and n, respectively, and r_{12} and p_{r12} are a uniform distribution random number and a predetermined probability, respectively. Consequently, U_i and V_i are formulated as

$$u_{ij} = \begin{cases} g_{mj}, & r_{12} < p_{r12} \\ \psi_{mj}, & r_{12} \geq p_{r12} \end{cases} \quad (6)$$

and

$$v_{ij} = \begin{cases} g_{nj}, & r_{12} < p_{r12} \\ \phi_{nj}, & r_{12} \geq p_{r12}, \end{cases} \quad (7)$$

respectively. Here, g_{mj}, ψ_{mj}, g_{nj}, and ϕ_{nj} are the jth dimension of G_m, Ψ_i, G_n, and Φ_i, respectively.

E. STEP SIZE FUNCTION

Finally, the step size function is defined as

$$\xi_j(k) = r_j \mathrm{logsig}[(0.5 \times K - k)/\mathrm{c}], \quad (8)$$

where K is the maximum iterative number, k is the current iterative number, and c is a parameter factor for switching the slope of the step size function and improving the global and local search capability.

F. INDIVIDUAL SELECTION STRATEGY

In each iteration, an individual selection strategy is employed to determine all competitive solutions in the entire swarm. More specially, the fitness value of each idea $X_i(k+1)$, $i \in \{1, 2, \cdots, N\}$ in the (k+1)th iteration is compared with that of each idea $X_i(k)$, $i \in \{1, 2, \cdots, N\}$ in the kth iteration. The idea with the better fitness value between $X_i(k+1)$ and $X_i(k)$, $i \in \{1, 2, \cdots, N\}$ is selected as the new idea for next iterative update.

Here, without loss of generality, we assume that the considered fitness value is for minimization. Consequently, the individual selection procedure is formulated as follows:

$$X_i(k+1) = \begin{cases} X_i(k+1), & f[X_i(k+1)] < f[X_i(k)] \\ X_i(k), & f[X_i(k+1)] \geq f[X_i(k)] \end{cases}, \quad (9)$$

where $f[X_i(k)]$ and $f[X_i(k+1)]$ are the fitness value of the idea $X_i(k)$ an $X_i(k+1)$, respectively. The individual selection aims to achieve the essence of everyone in each iteration.

After the individual selection for all N new ideas have been completed in each iteration, the termination conditions of the BSO algorithm needs to be checked. If the termination condition is matched, the iterative process will be terminated and the corresponding results are obtained. Otherwise, the BSO algorithm will continue to run until its termination conditions are met.

III. PROPOSED BSO ALGORITHM

In the original BSO algorithm, its individual update strategy, however, ignored some potential information interaction patterns such as information interactions between individuals from one cluster, and interactions between the corresponding dimensions of individuals. Although most of variants BSO algorithm are proposed, their improvements chiefly focused on the clustering strategy or the step-size function of the individual update strategy rather than the individual update patterns of the individual update strategy. Recently, the ADMBSO algorithm adopted a discussing strategy to improve the individual interaction patterns [33]. However, this algorithm also ignored the interaction between the corresponding dimensions of individuals. More recently, the GBSO algorithm adopted a dimension information interaction pattern [35]. However, it failed to involve the individual interactions like that of the original BSO or ADMBSO algorithm. Crucially, the neglect of these information interactions may trigger the loss of some beneficial solution information, which causes the original BSO algorithms to get trapped to in local optima, especially for tackling complex issues [40].

Furthermore, the K-means clustering method resulted in a high computational cost of the original BSO. In addition, the step-size function of the original BSO algorithm has two disadvantages. One is that the step size function fails to provide the feedback information for the individual update; the other is that it fails to acquire sufficient search ranges.

To overcome the disadvantages over the original BSO algorithm, a new MIIBSO algorithm is proposed. In MIIBSO, we first develop the MII strategy to enhance the information interaction capability and avoid the premature convergence. Subsequently, we adopt the RG strategy [20] to replace both the K-means strategy and the cluster center disruption to further enhance the information interaction capability and reduce the computational cost and redundancy. Finally, the DDS function is designed to improve the step size function of the original BSO algorithm.

A. MULTI-INFORMATION INTERACTION STRATEGY

In MIIBSO, the MII strategy consists of three different patterns including the MII pattern I, II, and III, providing the information interaction between individuals from one cluster, between individuals from two clusters, and between the corresponding dimensions of individuals, respectively, illustrated in detail as follows.

(1) MII Pattern I

According to (2) and (3) in the original BSO algorithm, a new idea created from one cluster only exploited a cluster center or an idea selected randomly, but did not consider the information interaction between the cluster center and the idea selected randomly. Thus, this may result in the decrease of the diversity of information interactions. To tackle this disadvantage, the MII pattern I uses the information interaction between two "old" ideas from one cluster to create a new idea as follows.

In the MII pattern I, a cluster center $G_m = [g_{m1}, g_{m2}, \cdots, g_{mD}]$, $m \in \{1, 2, \cdots, M\}$ is first randomly selected from M cluster centers $\{G_1, G_2, \cdots, G_M\}$. Then, an idea $\Psi = [\psi_1, \psi_2, \cdots, \psi_D]$ is randomly selected from this selected cluster m. Ultimately, a new idea $X_i = [x_{i1}, x_{i2}, \cdots, x_{iD}]$, $i \in \{1, 2, \cdots, N\}$ is created by using the information interaction between ideas Ψ and G_m as follows:

$$x_{ij} = r_j g_{mj} + (1 - r_j)\psi_j + \xi_j(\mathrm{k}) \quad (10)$$

where x_{ij}, ψ_j, and g_{mj} are the jth dimension of X_i, Ψ, and G_m, respectively; r_j is a random number uniformly distributed in the range $[0, 1]$; $\xi_j(\mathrm{k})$ represents the jth dimension of the step size function for balancing the global search and local search capability.

Note that an information interaction between an idea Ψ and the cluster center G_m is expressed as the item $r_j g_{mj} + (1 - r_j)\psi_j$ in (10). Such an item can possess more potential solution information compared with u_{ij} in (3). More specially, $r_j g_{mj} + (1 - r_j)\psi_j$ is to be equivalent to g_{mj}, ψ_j, or any combination between g_{mj} and ψ_j when r_j is equivalent to 1, 0, or any value between 0 and 1. In contrast, u_{ij} is only equivalent to g_{mj} or ψ_j based on (4) in the individual update pattern I of the original BSO. Therefore, the MII pattern I can provide more potential solution information compared with the individual update pattern I of the original BSO.

(2) MII Pattern II

The original BSO algorithm involved some redundant interaction information for creating a new idea from two clusters even though it considered the information interaction between individuals from two clusters. In particular, $r_j u_{ij} + (1 - r_j)v_{ij}$ in (5) has two possible outcomes, $r_j g_{mj} + (1 - r_j)g_{nj}$ and $r_j \psi_{mj} + (1 - r_j)\psi_{nj}$ based on (5), (6), and (7) in the original BSO algorithm. Note that ideas Ψ_i and Φ_i are not only representative of the cluster centers G_m and G_n, respectively, but also represent two different ideas from two clusters m and n other than the cluster centers G_m and G_n. From the aforesaid analysis, $r_j \psi_{mj} + (1 - r_j)\psi_{nj}$ also contains $r_j g_{mj} + (1 - r_j)g_{nj}$.

To eliminate redundant information and decrease the complexity, the MII pattern II adopts the information interaction between two "old" ideas from two clusters to create a new idea. Two different clusters m and n are first randomly selected from M cluster centers. Subsequently, two ideas $\Psi = [\psi_1, \psi_2, \cdots, \psi_D]$ and $\Phi = [\phi_1, \phi_2, \cdots, \phi_D]$ are randomly selected from the selected clusters m and n, respectively. Eventually, by applying the information interaction between ideas Ψ and Φ, a new idea $X_i = [x_{i1}, x_{i2}, \cdots, x_{iD}]$, $i \in \{1, 2, \cdots, N\}$ is created as follows:

$$x_{ij} = r_j \psi_j + (1 - r_j)\phi_j + \xi_j(\mathrm{k}) \quad (11)$$

where x_{ij}, ψ_j, and ϕ_j are the jth dimension of X_i, Ψ, and Φ, respectively.

(3) MII Pattern III

In the CLPSO algorithm [41], the individual update used a dimensional update pattern to improve the diversity of the population and promote the global search capability. Inspired by this dimensional update of CLPSO, the MII pattern III adopts the information interaction pattern between the corresponding dimensions of two ideas to create a new idea as follows.

For the j th dimension x_{ij}, $j \in \{1, 2, \cdots, D\}$ of the individual X_i, Δ ideas $\{\Psi_1, \Psi_2, \cdots, \Psi_\Delta | j\}$, $2 \le \Delta \le N$ are first randomly selected from N ideas. We then compare the fitness values of the Δ ideas, identifying the better ($\Delta = 2$) or best fitness value ($2 < \Delta \le N$). Here, without loss of generality, we assume that the better or best fitness value amongst the fitness values $\{f[\Psi_1], f[\Psi_2], \cdots, f[\Psi_\Delta] | j\}$ is for minimization. Next, the idea that corresponds to the minimum fitness value can be determined as

$$\{\Psi | j\} = \arg(\min\{f[\Psi_1], f[\Psi_2], \cdots, f[\Psi_\Delta] | j\}). \quad (12)$$

Therefore, the jth dimension x_{ij} of the new idea X_i can be updated as

$$x_{ij} = \{\psi_j | j\} + \xi_j(\mathrm{k}), \quad (13)$$

where x_{ij} and $\{\psi_j | j\}$ are the jth dimension of the new idea X_i and $\{\Psi | j\}$, respectively.

Note that each dimension $\{\psi_j | j\}$, $j \in \{1, 2, \cdots, D\}$ derives from the jth dimension of one of Δ ideas $\{\Psi_1, \Psi_2, \cdots, \Psi_\Delta | j\}$ selected randomly so that each $x_{ij}(k)$, $j \in \{1, 2, \cdots, D\}$ in the MII pattern III also has more opportunities to achieves many new potential solutions.

In summary, the three MII patterns can provide various potential information interactions compared with the original BSO algorithm.

(4) Individual Stagnation Feedback Mechanism

Note that the MII pattern III needs to update each dimension of an idea. Consequently, the MII pattern III can provide promising information interactions of the individual to enhance the global search capability of MIIBSO. However, such a pattern is to cause high computational costs of MIIBSO if it is performed for each idea in each iteration. In addition, the MII pattern I focuses on local search capability. Conversely, the MII pattern II emphasizes on global search capability. To provide an efficient collaboration for the three patterns, we develop the ISF mechanism, illustrated as follows.

1) In each iteration, a new idea X_i, $i \in \{1, 2, \cdots, N\}$ can be created according to (10) in the MII pattern I when $r_1 < p_{r1}$ is true; otherwise, it can be created according to (11) in the MII pattern II. Here, r_1 and p_{r1} represent a uniform distribution random number and pre-determined probability p_{r1}, respectively.

2) Once the fitness value $f[X_i]$ of the new idea X_i, $i \in \{1, 2, \cdots, N\}$ has not been improved for successive ∇_i iterations, the new idea X_i is to be created by the pattern III to discourage the premature convergence and improve the global search capability. Here, ∇_i represents the stagnation number of the idea X_i, and is used as a trigger condition of the MII pattern III.

In the ISF mechanism, the probability selection mechanism is employed to select either the MII pattern I or the MII pattern II to provide a suitable compromise between the local and global search capability. In addition, the ISF mechanism can use rationally the MII pattern III via the stagnation number to achieve the effective balance between the global search capability and the computational cost. By utilizing the ISF mechanism, we can effectively take advantage of three MII patterns to preserve the diversity of the population, enhance the global search capability, and decrease the computational costs for MIIBSO.

B. RANDOM GROUPING STRATEGY

Furthermore, we introduce the RG strategy [20] into the MIIBSO algorithm for two purposes: first, to further enhance information interaction diversity for the MII strategy, and second to decrease the computational cost for the MIIBSO algorithm. The RG strategy is demonstrated as follows.

First, we randomly split N ideas into m clusters, and each cluster $m \in \{1,2,\cdots,M\}$ has the same number (N/m) of ideas. Subsequently, for every idea in each cluster $m \in \{1,2,\cdots,M\}$, its fitness value is compared with that of other ideas in the same cluster. The idea with the best fitness value is selected as the cluster center G_m, $m \in \{1,2,\cdots,M\}$. Afterwards, instead of precise clustering, the RG strategy randomly allocates the same number of ideas to each cluster. Therefore, various solution information can be randomly supplied for each cluster, which effectively increases the diversity of information interactions in the MII strategy. Moreover, the RG strategy can provide such an effect similar to the combination of the K-means method and the cluster center disruption. In addition, the RG strategy does not need to calculate the distances between all the ideas. From these reasons, we use the RG strategy to replace the K-means method and the cluster center disruption in the original BSO to enhance the performance of MIIBSO.

C. DYNAMIC DIFFERENCE STEP SIZE FUNCTION

In the original BSO algorithm, the step size function could neither provide feedback information nor obtain sufficient search ranges. To overcome the aforesaid imperfections, the DDS function is developed by introducing a dynamically adjustable parameter into the difference step-size function to achieve an effective balance between the local search and global search capability. The DDS function is formulated as

$$\xi_j(k) = \delta(k)\left(x_{mj} - x_{nj}\right), \tag{14}$$

where x_{mj} and x_{nj} are the jth dimension of different ideas X_m and X_n that are randomly selected from the entire swarm. The dynamically adjustable parameter $\delta(k)$ is given as

$$\delta(k) = 0.5 \times 2^{\exp[1-K/(K+1-k)]} \tag{15}$$

where *K* and *k* are the maximum iterative number and the current iterative number, respectively.

Specially, in the early iterative search, the item $\left(x_{mj} - x_{nj}\right)$ from (14) reflects a large difference between x_{mj} and x_{nj}, contributing to enhance the global search capability. In contrast, this item reflects a small difference in the convergence process, contributing to refine a local search. Therefore, this item plays an information feedback role in the update of ideas, and causes an effective balance between the local and global search for the MIIBSO algorithm. Furthermore, in (15), $\delta(k)$ provides an effective balance between global and local search for MIIBSO. Specially, in the early stages of iterative search with the small iterative number *k*, $\delta(k)$ is approximately equivalent to 1, and can provide a large search scope for the DDS function to enhance the global search capability of MIIBSO. On the other hand, in the late periods of iterative search with the large iterative number *k*, $\delta(k)$ is gradually reduced to 0.5. In this case, $\delta(k)$ provides a small search scope for the DDS function to improve the local search capability of MIIBSO and accelerate the convergence speed. Therefore, $\delta(k)$ further improves the global and local search capability.

In addition, owing to x_{mj} and x_{nj} that are situated in$\left[x_{j_min}, x_{j_max}\right]$, $\xi_j(k)$ can provide enough search ranges for MIIBSO.

D. TIME COMPLEXITY ANALYSIS OF MIIBSO

We assume that $\mathcal{F}$ is the maximum number of the fitness evaluations (FEs). The computational cost of the original BSO algorithm mainly concerns the individual initialization (T_{ini}), the individual clustering (T_{inc}), the individual update (T_{inu}), and individual selection (T_{ins}) for all individuals. Then, the time complexity of the original BSO algorithm can be evaluated as

$$\begin{aligned} T(D) &= T_{\text{ini}} + T_{\text{inc}} + T_{\text{inu}} + T_{\text{ins}} \\ &= D + \mathcal{F}DNYM + \mathcal{F}DN + \mathcal{F}DN \\ &= D(1 + \mathcal{F}NYM + 2\mathcal{F}N) \end{aligned} \tag{16}$$

where D, N, Y and M are the dimension of solution search space, the number of all the individuals, the maximum iteration of the K-means clustering method, and the number of all the clusters, respectively. Therefore, the time complexity of the original BSO is assessed as $O(D\mathcal{F}NYM)$.

Unlike the original BSO algorithm, the MIIBSO algorithm adopts the RG strategy so that its T_{inc} can be accessed as $\mathcal{F}M$. Furthermore, considering the worst scenarios that the MIII pattern III with the corresponding selection strategy is performed in each iteration, we can evaluate both T_{inu} and T_{ins} as $2\mathcal{F}DN$. In this case, the time complexity of the MIIBSO algorithm can be expressed as

$$\begin{aligned} T(D) &= T_{\text{ini}} + T_{\text{inc}} + T_{\text{inu}} + T_{\text{ins}} \\ &= D + \mathcal{F}(M + 2DN + 2DN) \\ &= D + \mathcal{F}M + 4\mathcal{F}DN\,. \end{aligned} \tag{17}$$

Hence, the time complexity of the MIIBSO algorithm is assessed as $O(\mathcal{F}DN)$ in the worst case.

From the above analysis, the proposed MIIBSO algorithm has lower time complexity compared with the original BSO algorithm. The essential reason is that the proposed BSO algorithm adopts the RG strategy rather than the K-means clustering method.

E. PROCEDURE OF MIIBSO

For each iteration of MIIBSO, we first randomly initialize all the

ideas in the entire swarm, and their fitness values are evaluated. Second, the RG strategy is executed. Third, the MII strategy combining with DDS function is applied to create new ideas and conduct individuals update. Fourth, the individual selection is employed to pick out the ideas with best fitness value. Finally, MIIBSO ceases running when termination criteria are true; or else it continues to the next iteration. The pseudo code of MIIBSO is illustrated in Algorithm 1, and it's the corresponding flowchart is shown in Fig. 1.

Additionally, in MIIBSO, the search range of the dimension x_{ij} $(i \in \{1,2,\cdots,N\})$ of the idea X_i $(i \in \{1,2,\cdots,N\})$ is constrained to $\min\left\{x_{j\_max}, \max\left\{x_{j\_min}, x_{ij}\right\}\right\}$.

Algorithm 1 MIIBSO Algorithm

/***Initialization***/
Randomly initialize each idea $X_i, i \in \{1,2,\cdots,N\}$ using (1);
Evaluate the fitness value $f[X_i]$, $, i \in \{1,2,\cdots,N\}$;
while *(termination criteria is not true)* **do**
  k=k+1;
  /***RG Strategy***/
  Randomly divide N ideas $\{X_1(k), X_2(k), \cdots, X_N(k)\}$ into M clusters, and each cluster $m \in \{1,2,\cdots,M\}$ owns N/M ideas;
  For each cluster $m \in \{1,2,\cdots,M\}$, the idea with the best fitness value is selected as the cluster center G_m;
  /***MII Strategy***/
  for i=1 **to** N **do**
    if $r_1 < p_{r1}$
      Randomly select one cluster from M cluster;
      for j=1 **to** D **do**
        Update x_{ij} by (10) and (14); **/*DDS Function*/**
      end for
    else
      Randomly select two cluster from M cluster;
      for j=1 **to** D **do**
        Update x_{ij} by (11) and (14); **/*DDS Function*/**
      end for
    end if
    if $f[X_i(k+1)] < f[X_i(k)]$
      $X_i(k+1)$=$X_i(k+1)$;
      sg_i=0;
    else
      $X_i(k+1)$=$X_i(k)$;
      sg_i=$\text{sg}_i + 1$;
    end if
    if sg_i==∇
      for j=1 **to** D **do**
        Randomly select $\{\Psi_1(k), \Psi_2(k), \cdots, \Psi_\Delta(k)|j\}$ from N ideas;
        Select the idea with the best fitness value by (12);
        Update x_{ij} by (13) and (14); **/*DDS Function*/**
      end for
      if $f[X_i(k+1)] < f[X_i(k)]$
        $X_i(k+1)$=$X_i(k+1)$;
      else
        $X_i(k+1)$=$X_i(k)$;
      end if
    end if
  end for
end while

IV. PERFORMANCE EVALUSTION

A. BENCHMARK FUNCTIONS AND COMPARED ALGORITHMS

We assessed the performance of MIIBSO via a popular test suite, called CEC2013 benchmark functions [40], shown in Table I. It consists of 28 benchmark functions involving shifted functions and shifted rotated functions for real parameter optimization in extremely complex circumstances. In terms of their characteristics, the 28 functions are categorized into three categories: unimodal functions *f*1-*f*5, multimodal functions *f*6-*f*20, and composition functions *f*21-*f*28. Their dimensions are initialized in [-100, 100]. Likewise, the search range for each dimension is assigned to [-100, 100]. Note that $f[X^*]$ from Table I represents the global optimum of each function.

To verify the performance of MIIBSO, we first compared it with the original BSO algorithms [7] and the other nine BSO variants. Note that the nine BSO variants have provided good performances in the literature, involving MBSO [18], the closed-loop BSO with random selection (CLBSO-RS) [26], PPBSO [30], the BSO with chaotic operation (BSOCO) [28], SBSO [25], the BSO algorithm with a differential evolution (BSODE) [27], ADMBSO [33], RGBSO [20], the BSO with dynamic clustering strategy (BSO-DCS) [23], and GBSO [35]. In the BSOCO algorithm, the logistic map of chaotic operation is executed 200 times iteraation in each 5 iterations.

TABLE I
CEC2013 BENCHMARK FUNCTIONS WITH INITIALIZATION AND SEARCH RANGE: $[-100, 100]^D$

Types	No.	Functions	$f[X^*]$
Unimodal	f1	Shifted Sphere Function	-1400
	f2	Shifted Rotated High Conditioned Elliptic Function	-1300
	f3	Shifted Rotated Bent Cigar Function	-1200
	f4	Shifted Rotated Discus Function	-1100
	f5	Shifted Different Powers Function	-1000
Multimodal	f6	Shifted Rotated Rosenbrock's Function	-900
	f7	Shifted Rotated Schaffers F7 Function	-800
	f8	Shifted Rotated Ackley's Function	-700
	F9	Shifted Rotated Weierstrass Function	-600
	f10	Shifted Rotated Griewank's Function	-500
	f11	Shifted Rastrigin's Function	-400
	f12	Shifted Rotated Rastrigin's Function	-300
	f13	Shifted Non-Continuous Rotated Rastrigin's Function	-200
	f14	Shifted Schwefel's Function	-100
	f15	Shifted Rotated Schwefel's Function	100
	f16	Shifted Rotated Katsuura Function	200
	f17	Shifted Lunacek Bi_Rastrigin Function	300
	f18	Shifted Rotated Lunacek Bi_Rastrigin Function	400
	f19	Shifted Rotated Expanded Groewank's plus Rosenbrock Function	500
	f20	Shifted Rotated Expanded Scaffer's F6 Function	600
Composition	f21	Composition Function 1 (n=5, Rotated)	700
	f22	Composition Function 2 (n=3, Unrotated)	800
	f23	Composition Function 3 (n=3, Rotated)	900
	f24	Composition Function 4 (n=3, Rotated)	1000
	f25	Composition Function 5 (n=3, Rotated	1100
	f26	Composition Function 6 (n=5, Rotated)	1200
	f27	Composition Function 7 (n=5, Rotated)	1300
	f28	Composition Function 8 (n=5, Rotated)	1400

Besides, we further compared MIIBSO with PSO [1], DE[3], ABC [44], and the covariance matrix adaptation evolution strategy (CMA-ES) [43] to confirm if it could outperform them broadly applied. The PSO and DE adopt the global version and the DE/rand/1/bin version, respectively. Since the dimensional update pattern of CLPSO [41] is incorporated into MIIBSO, a comparison was also performed between them. In Table II, parameter configurations of all the algorithms follow the original literature, excluding the population size, the solution dimensional size, and the maximum number of FEs. All the algorithms were programmed in MATLAB R2014b, and then performed on a PC with an Intel Core (TM) CPU i5-3230M CPU @ 2.60 GHz with 4 GB RAM.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2821118, IEEE Access

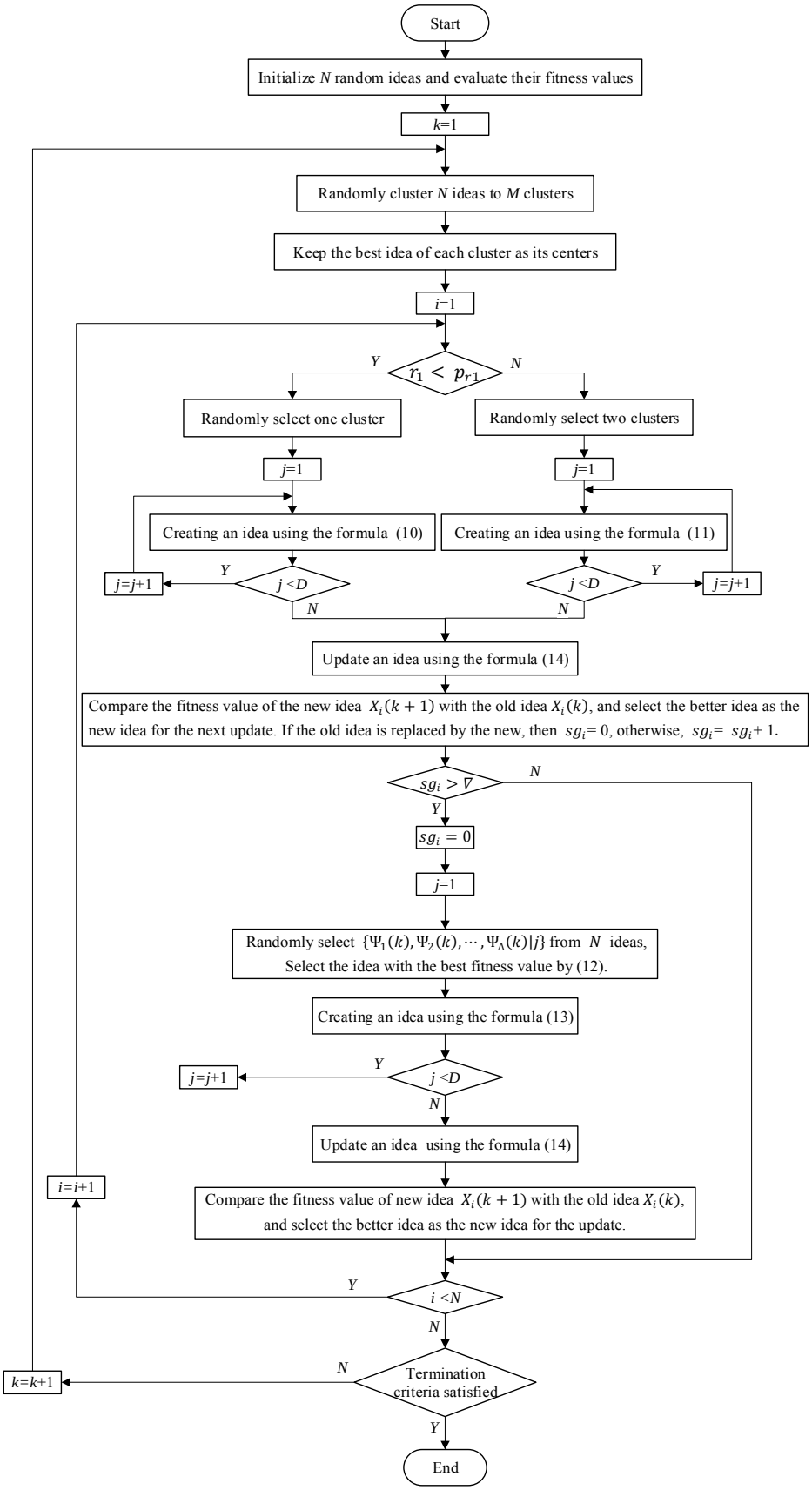

FIGURE 1. Flowchart of MIIBSO

TABLE II
ALGORITHM PARAMETER CONFIGURATIONS

Algorithm	Year	Parameter Settings	Reference
DE	1997	CR=0.5, F=0.5	[3]
PSO	1999	ω: 0.9-0.4, c1=2, c2=2	[1]
CMA-ES	2001	$\lambda = 4 + \lfloor 3 \ln n \rfloor$, $\mu = \lfloor 0.5\,\lambda \rfloor$	[43]
CLPSO	2006	ω: 0.9-0.4, c=1.49445, sg=5, p_c= 0.05-0.5	[41]
ABC	2007	α=1, limit=100	[44]
BSO	2011	p_{r0}=0.2, p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, c=25, M=5	[7]
MBSO	2012	p_{r0}=0.2, p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, c=25, M=5, p_r= 0.005	[18]
CLBSO-RS	2013	p_{r0}=0.2, p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, M=5	[26]
PPBSO	2013	p_{r0}=0.2, p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, M=5, p_{prey}=0.1, $W_{predator}$=0.05	[30]
BSOCO	2015	p_{r0}=0.2, p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, c=25, M=5, r=4	[28]
SBSO	2015	p_{r0}=0.2, p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, c=25, M=5	[25]
BSODE	2015	p_{r0}=0.2, p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, M=5, CR=0.5, F=0.5	[27]
ADMBSO	2015	p_{cen}=0.7, p_{ind}=0.2, p_{rnd}=0.1, p_{cens}=0.7, p_{low}=0.2, p_{high}=0.2, M=5	[33]
RGBSO	2015	p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, M=5	[20]
BSO-DCS	2016	p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, M=5, $p_{dynamic}$=0.5	[23]
GBSO	2017	p_{r1}=0.8, p_{r11}=0.4, p_{r21}=0.5, c=25, M=5, ∇=10, F=0.5, C_{min}=0.2, C_{max}=0.8	[35]
MIIBSO		p_{r1}=0.8, M=5, Δ=20, ∇=30	

B. PARAMETERS REGULATING

As for the MIIBSO algorithm, two significant parameters from the MII strategy need to be regulated to enhance the information interaction capability and obtain the effective balance between global and local search capability. One is the number of ideas (Δ) that are randomly selected from the entire swarm to update each dimension of the new idea in the MII pattern III. The other is the update stagnation number of the idea (∇) in the ISF mechanism. They are regulated by performing examinations on the 28 CEC2013 benchmark functions. Each function runs 30 times independently, its dimensions being set to D=30. The population size, the maximum number of the Fes, and the maximum iterative number are set to 50, 400000, and 8000, respectively. Experimental results of each parameter are ranked via the error mean and standard deviation values. The final rank of each parameter is determined by the average ranks of the 28 functions.

To ascertain the best appropriate value of Δ, we evaluate a set of setting values, Δ=5, Δ=10, Δ=15, Δ=20, Δ=25, Δ=35, and Δ=45 with the stagnation number ∇=30 on the 28 CEC2013 benchmark functions. Because of the space limitation, the detailed results are given in Section S-I of the supplementary file, which shows the error mean values, standard deviation values, the average ranks, and final ranks of the setting values on the 28 functions. Table III only lists average and final ranks for the different setting values of Δ. The best result on overall performance among all the setting values is highlighted in bold. As shown in Table III, Δ=20 owns the best final rank among all the setting values. This implies that Δ=20 provides the best overall performance for MIIBSO.

TABLE III
COMPARISONS AMONG SETTING VALUES OF PARAMETER Δ WITH ∇=30

Evaluation Criteria	Δ=5	Δ=10	Δ=15	Δ=20	Δ=25	Δ=35	Δ=45
Average Rank	4.39	4.36	3.54	**3.11**	4.11	3.75	3.68
Final Rank	7	6	2	**1**	5	4	3

Furthermore, different setting values of ∇ with Δ=20 are tested on the 28 functions so as to identify the suitable setting value. The detailed information of the parameter ∇ is also given in Section S-I of the supplementary file. Only the average and final ranks of the setting values of the parameter ∇ is listed in Table IV, where the results indicate that ∇=30 supplies the best performance consistently for MIIBSO.

TABLE IV
COMPARISONS AMONG SETTING VALUES OF PARAMETER ∇ WITH Δ=20

Evaluation Criteria	∇=5	∇=10	∇=15	∇=25	∇=30	∇=35	∇=40
Average Rank	4.32	4.29	3.71	3.93	**3.04**	3.57	4.00
Final Rank	7	6	3	4	**1**	2	5

Hence, we use Δ=20 and ∇=30 to further assess the proposed MIIBSO algorithm.

C. COMPARISON WITH BSO ALGORITHMS

We first compare MIIBSO with ten different BSO algorithms on the 28 CEC2013 benchmark functions with 30 dimensions. The population size, the maximum number of the Fes, and the maximum iterative number are assigned to 50, 400000, and 8000, respectively. Each algorithm is evaluated on each function in 30 independent operations. Table V lists the error mean and standard deviation values of all the functions for each algorithm with the best values highlighted in bold. In accordance with the error mean and standard deviation values, we ascertain the average and final rank of each algorithm.

In Table V, we observe that MBSO provides the best results on functions *f*20, *f*21, and *f*23. SBSO obtains the best on functions *f*1 and f15. CLBSO and RGBSO obtain the best on functions *f*19 and f16, respectively. ADMBSO yields the best on functions *f*2, *f*13, *f*17, *f*25, and *f*26. BSO-DCS offers the best on functions *f*8 and *f*10. MIIBSO algorithm wins the best on functions *f*1, *f*3-*f*7, *f*9, *f*11, *f*12, *f*14, *f*18, *f*22, *f*24, *f*27, and *f*28. An interesting thing is that MIIBSO provides the first rank on 15 out of the 28 functions, which is the largest number of the first ranks among all the BSO algorithms.

Furthermore, none of the MIIBSO, BSODE, and BSOCO algorithms yields the worst results on the 28 functions. However, neither of BSODE and BSOCO achieves the best results on all the functions. Particularly, the other eight BSO algorithms yield the worst on some of the 28 functions.

From the analyses above, MIIBSO is unable to obtain the best results on all the functions, but it provides relatively better results compared with other examined algorithms except for functions *f*2, *f*15, *f*16, and *f*26. To fairly compare their overall performance, we provide the average and final rank on the 28 functions for each algorithm. The results show that MIIBSO achieves the best overall performance on the 28 functions among all the algorithms.

In addition, we also provide specialized comparisons for all the algorithms on unimodal functions, multimodal functions, and composition functions as follows.

For unimodal functions *f*1-*f*5, SBSO and ADMBSO achieve the best results on functions *f*1 and *f*2, respectively. MIIBSO provides the best on functions *f*1 and *f*3-*f*5. The final rank of each algorithm on unimodal functions *f*1-*f*5 is listed in table VI, where MIIBSO achieves the best overall performance on unimodal functions *f*1-*f*5 among all the algorithms.

For multimodal functions *f*6-*f*20, SBSO, RGBSO, CLBSO, and MBSO provide the best results on functions f15, f16, *f*19, and *f*20, respectively. ADMBSO yields the best on functions *f*13 and

TABLE V
COMPARISONS OF MIIBSO AND TEN BSO ALGORITHMS ON 28 CEC2013 FUNCTIONS WITH DIMENSION 30

Functions	Evaluation Criteria	BSO	MBSO	CLBSO	PPBSO	SBSO	BSODE	RGBSO	ADMBSO	BSO-DCS	BSOCO	MIIBSO
*f*1	Mean	1.36E-13	3.41E-13	2.43E-13	4.98E-02	**0.00E+00**	1.67E-13	1.52E-13	2.65E-13	6.82E-14	1.82E-13	**0.00E+00**
	Std	1.11E-13	2.01E-13	1.43E-13	2.33E-02	0.00E+00	1.01E-13	1.07E-13	1.19E-13	1.04E-13	9.09E-14	0.00E+00
	rank	4	10	8	11	**1**	6	5	9	3	7	**1**
*f*2	Mean	9.79E+05	1.22E+05	1.25E+05	2.22E+06	1.78E+06	2.89E+05	2.14E+05	**1.02E+05**	2.70E+05	9.90E+05	3.06E+05
	Std	3.34E+05	7.50E+04	7.25E+04	7.05E+05	2.97E+05	1.12E+05	1.22E+05	5.15E+04	1.33E+05	2.44E+05	5.48E+05
	rank	8	2	3	11	10	6	4	**1**	5	9	7
f 3	Mean	6.40E+07	1.24E+07	1.15E+07	1.15E+08	5.75E+07	2.00E+07	4.12E+07	1.55E+07	3.14E+07	7.63E+07	**4.92E+05**
	Std	6.02E+07	2.36E+07	2.04E+07	1.35E+08	1.21E+08	2.71E+07	6.16E+07	4.96E+07	4.90E+07	9.81E+07	1.29E+06
	rank	9	3	2	11	8	5	7	4	6	10	**1**
*f*4	Mean	1.35E+04	1.82E+02	2.42E+03	6.10E+04	3.14E+04	2.82E+00	6.57E+03	5.48E+01	1.28E+04	1.29E+04	**6.31E-02**
	Std	6.32E+03	3.52E+02	4.70E+03	1.64E+04	6.15E+03	5.41E+00	2.61E+03	3.75E+01	3.93E+03	5.36E+03	1.37E-01
	rank	9	4	5	11	10	2	6	3	7	8	**1**
*f*5	Mean	1.08E-02	3.83E-13	2.43E-13	5.58E-02	6.08E-03	1.18E-03	1.44E-03	4.21E-13	1.25E-03	1.08E-02	**1.14E-13**
	Std	2.67E-03	1.62E-13	1.05E-13	1.42E-02	1.70E-03	1.87E-04	2.69E-04	2.22E-13	2.62E-04	2.96E-03	2.94E-14
	rank	9	3	2	11	8	5	7	4	6	10	**1**
*f*6	Mean	3.43E+01	1.95E+01	1.33E+01	5.80E+01	5.46E+01	2.59E+01	2.76E+01	2.09E+01	2.35E+01	4.27E+01	**9.59E+00**
	Std	2.79E+01	2.21E+01	7.25E+00	2.50E+01	2.52E+01	2.24E+01	2.31E+01	2.18E+01	2.14E+01	2.99E+01	7.68E+00
	rank	8	3	2	11	10	6	7	4	5	9	**1**
*f*7	Mean	1.19E+02	3.32E+01	3.06E+01	1.29E+02	9.88E+01	1.80E+02	1.85E+02	2.78E+01	1.39E+02	1.33E+02	**2.92E+00**
	Std	3.07E+01	2.56E+01	1.86E+01	2.90E+01	2.94E+01	5.53E+01	1.04E+02	1.76E+01	6.07E+01	6.22E+01	3.27E+00
	rank	6	4	3	7	5	10	11	2	9	8	**1**
f 8	Mean	20.8857	20.9434	20.9234	20.9185	20.9140	20.8818	20.8673	20.9321	**20.8534**	20.8919	20.9137
	Std	8.75E-02	4.25E-02	5.33E-02	5.65E-02	5.07E-02	5.78E-02	5.92E-02	4.43E-02	6.98E-02	5.94E-02	5.93E-02
	rank	4	11	9	8	7	3	2	10	**1**	5	6
*f*9	Mean	3.23E+01	2.04E+01	2.34E+01	3.24E+01	2.70E+01	3.04E+01	3.20E+01	2.67E+01	3.26E+01	3.21E+01	**1.49E+01**
	Std	2.90E+00	4.43E+00	4.05E+00	2.82E+00	3.72E+00	3.06E+00	3.55E+00	4.78E+00	3.09E+00	2.78E+00	3.43E+00
	rank	9	2	3	10	5	6	7	4	11	8	**1**
*f*10	Mean	3.10E-01	2.03E-01	1.48E-01	1.08E+00	3.40E-01	7.72E-03	3.86E-03	2.51E-01	**3.20E-03**	2.79E-01	1.17E-02
	Std	2.70E-01	8.94E-02	8.01E-02	3.84E-02	2.46E-01	6.57E-03	5.57E-03	1.22E-01	5.13E-03	2.76E-01	7.56E-03
	rank	9	6	5	11	10	3	2	7	**1**	8	4
*f*11	Mean	3.95E+02	5.89E+01	5.57E+01	1.33E+02	3.36E+02	2.99E+02	4.94E+02	4.44E+01	4.35E+02	4.06E+02	**2.22E+01**
	Std	8.44E+01	1.85E+01	1.77E+01	3.10E+01	6.89E+01	6.79E+01	9.28E+01	1.31E+01	9.73E+01	6.77E+01	6.68E+00
	rank	8	4	3	5	7	6	11	2	10	9	**1**
*f*12	Mean	3.97E+02	6.45E+01	5.45E+01	3.74E+02	3.19E+02	3.54E+02	4.71E+02	4.75E+01	4.60E+02	4.27E+02	**3.69E+01**
	Std	8.61E+01	2.27E+01	1.40E+01	8.22E+01	6.02E+01	7.67E+01	8.83E+01	1.49E+01	1.02E+02	9.18E+01	1.50E+01
	rank	8	4	3	7	5	6	11	2	10	9	**1**
*f*13	Mean	5.14E+02	1.24E+02	1.18E+02	4.20E+02	4.26E+02	4.16E+02	5.02E+02	**1.07E+02**	5.21E+02	4.82E+02	1.13E+02
	Std	8.38E+01	2.83E+01	3.57E+01	7.52E+01	6.05E+01	7.87E+01	8.34E+01	2.89E+01	7.40E+01	7.46E+01	4.35E+01
	rank	10	4	3	6	7	5	9	**1**	11	8	2
f 14	Mean	4.18E+03	2.74E+03	3.09E+03	2.70E+03	3.88E+03	3.32E+03	4.41E+03	3.46E+03	4.39E+03	4.06E+03	**2.34E+03**
	Std	6.33E+02	8.72E+02	8.55E+02	6.41E+02	7.45E+02	5.19E+02	5.49E+02	6.93E+02	7.12E+02	5.49E+02	9.06E+02
	rank	9	3	4	2	7	5	11	6	10	8	**1**
*f*15	Mean	4.23E+03	4.95E+03	5.30E+03	4.28E+03	**4.19E+03**	4.34E+03	4.42E+03	6.32E+03	4.38E+03	4.36E+03	5.40E+03
	Std	6.97E+02	1.59E+03	1.45E+03	4.59E+02	6.77E+02	4.84E+02	5.99E+02	1.42E+03	5.75E+02	5.06E+02	1.55E+03
	rank	2	8	9	3	**1**	4	7	11	6	5	10
*f*16	Mean	1.90E-01	2.39E+00	2.42E+00	3.95E-01	5.89E-01	3.59E-02	**1.42E-02**	2.38E+00	1.48E-02	1.88E-01	2.27E+00
	Std	6.97E-02	2.67E-01	2.64E-01	1.29E-01	3.67E-01	2.18E-02	7.59E-03	3.00E-01	5.86E-03	7.05E-02	3.73E-01
	rank	5	10	11	6	7	3	**1**	9	2	4	8
*f*17	Mean	4.28E+02	8.90E+01	8.23E+01	2.55E+02	3.45E+02	4.75E+02	5.50E+02	**7.41E+01**	5.36E+02	3.85E+02	1.58E+02
	Std	9.41E+01	1.51E+01	1.27E+01	4.66E+01	6.43E+01	1.02E+02	1.12E+02	1.36E+01	1.07E+02	6.52E+01	5.40E+01
	rank	8	3	2	5	6	9	11	**1**	10	7	4
*f*18	Mean	3.82E+02	1.87E+02	1.87E+02	3.26E+02	4.03E+02	3.49E+02	5.12E+02	1.94E+02	4.89E+02	3.88E+02	**1.72E+02**
	Std	6.39E+01	4.00E+01	3.52E+01	6.01E+01	7.03E+01	9.23E+01	7.91E+01	2.17E+01	1.05E+02	7.48E+01	4.30E+01
	rank	7	3	2	5	9	6	11	4	10	8	**1**
*f*19	Mean	8.40E+00	7.48E+00	**5.28E+00**	9.20E+00	1.41E+01	1.04E+01	7.73E+00	5.78E+00	7.88E+00	7.37E+00	6.84E+00
	Std	2.00E+00	3.26E+00	2.04E+00	2.00E+00	2.68E+00	1.88E+00	1.23E+00	2.02E+00	1.40E+00	1.54E+00	4.94E+00
	rank	8	5	**1**	9	11	10	6	2	7	4	3
*f*20	Mean	1.45E+01	**1.13E+01**	1.15E+01	1.44E+01	1.33E+01	1.39E+01	1.45E+01	1.14E+01	1.45E+01	1.43E+01	1.17E+01
	Std	7.12E-03	6.08E-01	9.90E-01	3.42E-01	8.68E-01	8.38E-01	1.46E-01	9.67E-01	3.30E-03	5.51E-01	5.43E-01
	rank	11	**1**	3	8	5	6	9	2	10	7	4
*f*21	Mean	3.38E+02	**2.76E+02**	3.11E+02	3.30E+02	3.22E+02	2.95E+02	3.18E+02	2.97E+02	3.13E+02	2.91E+02	2.80E+02
	Std	8.02E+01	8.06E+01	6.82E+01	1.07E+02	7.05E+01	8.62E+01	8.46E+01	7.81E+01	1.02E+02	8.17E+01	4.00E+01
	rank	11	**1**	6	10	9	4	8	5	7	3	2
*f*22	Mean	5.70E+03	3.13E+03	3.22E+03	2.83E+03	4.46E+03	4.47E+03	5.51E+03	3.15E+03	5.68E+03	5.39E+03	**2.45E+03**
	Std	9.62E+02	5.81E+02	6.53E+02	7.14E+02	7.27E+02	8.73E+02	9.97E+02	9.72E+02	1.04E+03	7.73E+02	1.11E+03
	rank	11	3	5	2	6	7	9	4	10	8	**1**
*f*23	Mean	5.79E+03	**4.50E+03**	5.16E+03	5.32E+03	5.61E+03	5.33E+03	5.92E+03	4.93E+03	5.37E+03	5.17E+03	4.94E+03
	Std	7.12E+02	1.10E+03	1.09E+03	7.01E+02	7.81E+02	8.92E+02	8.79E+02	1.46E+03	8.55E+02	8.85E+02	1.31E+03
	rank	10	**1**	4	6	9	7	11	2	8	5	3
*f*24	Mean	3.16E+02	2.56E+02	2.60E+02	2.92E+02	3.06E+02	2.95E+02	3.24E+02	2.47E+02	3.23E+02	3.06E+02	**2.39E+02**
	Std	1.69E+01	9.88E+00	9.21E+00	1.26E+01	1.72E+01	1.18E+01	1.63E+01	1.06E+01	1.61E+01	1.50E+01	8.19E+00
	rank	9	3	4	5	7	6	11	2	10	8	**1**
*f*25	Mean	3.54E+02	2.84E+02	2.81E+02	3.14E+02	3.42E+02	3.18E+02	3.59E+02	**2.71E+02**	3.63E+02	3.30E+02	2.82E+02
	Std	1.63E+01	1.16E+01	1.37E+01	1.15E+01	1.34E+01	1.31E+01	1.54E+01	1.00E+01	1.82E+01	1.36E+01	2.18E+01
	rank	9	4	2	5	8	6	10	**1**	11	7	3
*f*26	Mean	2.61E+02	2.30E+02	2.35E+02	3.34E+02	2.87E+02	2.06E+02	2.62E+02	**2.00E+02**	2.32E+02	2.11E+02	2.56E+02
	Std	7.76E+01	6.10E+01	6.41E+01	7.46E+01	8.14E+01	2.86E+01	8.35E+01	7.67E-03	6.69E+01	3.27E+01	6.88E+01
	rank	8	4	6	11	10	2	9	**1**	5	3	7
*f*27	Mean	1.15E+03	8.36E+02	8.36E+02	1.12E+03	1.03E+03	1.11E+03	1.22E+03	7.76E+02	1.24E+03	1.16E+03	**6.60E+02**
	Std	1.21E+02	9.07E+01	9.60E+01	1.06E+02	8.25E+01	1.05E+02	9.18E+01	1.13E+02	1.14E+02	1.09E+02	7.68E+01
	rank	8	3	4	7	5	6	10	2	11	9	**1**
*f*28	Mean	4.17E+03	3.41E+02	3.00E+02	3.85E+03	3.49E+03	3.37E+03	4.11E+03	4.16E+02	4.04E+03	4.10E+03	**3.00E+02**
	Std	5.73E+02	2.22E+02	2.54E-13	4.14E+02	4.12E+02	6.98E+02	6.75E+02	3.50E+02	5.37E+02	6.29E+02	2.29E-13
	rank	11	3	2	7	6	5	10	4	8	9	**1**
Overall	**Average Rank**	8.14	4.11	4.14	7.54	7.11	5.54	7.96	3.89	7.50	7.25	**2.79**
Overall	**Final Rank**	11	3	4	9	6	5	10	2	8	7	**1**

TABLE VI
COMPARISONS OF MIIBSO AND TEN BSO ALGORITHMS ON UNIMODAL, MULTIMODAL, AND COMPOSITION FUNCTIONS

Functions	Evaluation Criteria	BSO	MBSO	CLBSO	PPBSO	SBSO	BSODE	RGBSO	ADMBSO	BSO-DCS	BSOCO	MIIBSO
Unimodal	Average Rank	7.8	4.4	4	11	7.4	4.8	5.8	4.2	5.4	8.8	**2.2**
	Final Rank	9	4	2	11	8	5	7	3	6	10	**1**
Multimodal	Average Rank	7.47	4.73	4.2	6.87	6.80	5.87	7.73	4.47	7.53	7.13	**3.2**
	Final Rank	9	4	2	7	6	5	11	3	10	8	**1**
Composition	Average Rank	9.625	2.75	4.125	6.625	7.5	5.375	9.75	2.625	8.75	6.5	**2.375**
	Final Rank	10	3	4	7	8	5	11	2	9	6	**1**

*f*17. BSO-DCS offers the best on functions *f*8 and *f*10. Unfortunately, the aforesaid six algorithms all generate the worst results on some of the 15 multimodal functions. However, MIIBSO wins the best results on functions *f*6, *f*7, *f*9, *f*11, *f*12, *f*14, and *f*18, without the worst results on the 15 multimodal functions. Table V and Table VI show that MIIBSO is consistently doing well and wins the best overall results when dealing with multimodal problems.

For composite functions *f*21- *f*28, MBSO provides the best results on functions *f*21 and *f*23. ADMBSO yields the best on functions *f*25 and *f*26. MIIBSO algorithm wins the best on functions *f*22, *f*24, *f*27, and *f*28. Similarly, Table VI shows that MIIBSO wins the best overall results on all the composite functions.

Therefore, MIIBSO achieves the best overall performance on the unimodal, multimodal, and composite functions.

D. COMPARISON WITH DE, PSO, AND CLPSO

We further compare MIIBSO with DE, PSO, and CLPSO on the 28 CEC2013 benchmark functions with 30 dimensions. Likewise, the population size, the maximum number of the Fes, and the maximum iterative number are also assigned to 50, 400000, and 8000, respectively. Each algorithm is evaluated on each function with 30 independent operations. Table VII lists the error mean and standard deviation values of all the functions for each algorithm with the best values marked in bold. Experimental results show that the four algorithms obtain the first and the second rank on seven and five, two and ten, nine and three, and 11 and nine out of the 28 functions, respectively. On the other hand, they obtain the worst results on 14, nine, four, and one, respectively.

In terms of the error mean and standard deviation values, the average and final rank of each algorithm are also listed in Table VII, where MIIBSO also achieves the best overall performance on the 28 functions among the four algorithms.

TABLE VII
COMPARISON OF MIIBSO, DE, PSO, AND CLPSO ON 28 CEC2013 FUNCTIONS WITH DIMENSION 30

Functions	Evaluation Criteria	DE	PSO	CLPSO	MIIBSO
*f*1	Mean	**0.00E+00**	4.40E-13	1.89E-13	**0.00E+00**
	Std	0.00E+00	1.68E-13	8.47E-14	0.00E+00
	rank	**1**	4	3	**1**
*f*2	Mean	5.99E+07	1.03E+07	1.56E+07	**3.06E+05**
	Std	1.37E+07	5.98E+06	3.09E+06	5.48E+05
	rank	4	2	3	**1**
f 3	Mean	**6.18E+01**	6.73E+07	1.43E+08	4.92E+05
	Std	1.38E+02	8.43E+07	9.52E+07	1.29E+06
	rank	**1**	3	4	2
*f*4	Mean	1.89E+04	2.72E+03	1.65E+04	**6.31E-02**
	Std	3.35E+03	8.23E+02	3.05E+03	1.37E-01
	rank	4	2	3	**1**
*f*5	Mean	**1.06E-13**	3.68E-13	1.21E-13	1.14E-13
	Std	2.84E-14	9.29E-14	2.84E-14	2.94E-14
	rank	**1**	4	3	2
*f*6	Mean	1.28E+01	7.24E+01	2.33E+01	**9.59E+00**
	Std	4.29E-01	3.36E+01	5.89E+00	7.68E+00
	rank	2	4	3	**1**
*f*7	Mean	**7.84E-02**	3.35E+01	6.39E+01	2.92E+00
	Std	2.76E-01	1.14E+01	8.96E+00	3.27E+00
	rank	**1**	3	4	2
f 8	Mean	20.9280	20.9089	20.9232	20.9137
	Std	5.30E-02	5.28E-02	4.86E-02	5.93E-02
	rank	4	**1**	3	2
*f*9	Mean	3.84E+01	2.13E+01	2.62E+01	**1.49E+01**
	Std	1.43E+00	2.76E+00	1.69E+00	3.43E+00
	rank	4	2	3	**1**
*f*10	Mean	**6.36E-03**	1.64E-01	1.83E+00	1.17E-02
	Std	6.56E-03	1.16E-01	3.52E-01	7.56E-03
	rank	**1**	3	4	2
*f*11	Mean	8.24E+01	2.07E+01	**5.49E-14**	2.22E+01
	Std	7.98E+00	5.99E+00	1.02E-14	6.68E+00
	rank	4	2	**1**	3
*f*12	Mean	1.80E+02	7.38E+01	9.76E+01	**3.69E+01**
	Std	1.09E+01	2.80E+01	1.16E+01	1.50E+01
	rank	4	2	3	**1**
*f*13	Mean	1.77E+02	1.46E+02	1.40E+02	**1.13E+02**
	Std	1.09E+01	3.94E+01	1.81E+01	4.35E+01
	rank	4	3	2	**1**
f 14	Mean	3.98E+03	7.80E+02	**1.51E+00**	2.34E+03
	Std	2.06E+02	2.83E+02	9.28E-01	9.06E+02
	rank	4	2	**1**	3
*f*15	Mean	7.15E+03	6.34E+03	**4.19E+03**	5.40E+03
	Std	2.31E+02	1.08E+03	3.58E+02	1.55E+03
	rank	4	3	**1**	2
*f*16	Mean	2.35E+00	2.15E+00	**1.74E+00**	2.27E+00
	Std	2.70E-01	3.10E-01	2.23E-01	3.73E-01
	rank	4	2	**1**	3
*f*17	Mean	1.15E+02	4.82E+01	**3.10E+01**	1.58E+02
	Std	6.45E+00	1.79E+01	1.44E-01	5.40E+01
	rank	3	2	**1**	4
*f*18	Mean	2.08E+02	2.21E+02	1.86E+02	**1.72E+02**
	Std	9.67E+00	2.98E+01	1.29E+01	4.30E+01
	rank	3	4	2	**1**
*f*19	Mean	1.16E+01	3.11E+00	**1.18E+00**	6.84E+00
	Std	9.21E-01	9.40E-01	2.76E-01	4.94E+00
	rank	4	2	**1**	3
*f*20	Mean	1.23E+01	1.46E+01	1.34E+01	**1.17E+01**
	Std	2.05E-01	1.05E+00	4.25E-01	5.43E-01
	rank	2	4	3	**1**
*f*21	Mean	2.75E+02	2.89E+02	**2.72E+02**	2.80E+02
	Std	5.52E+01	7.63E+01	3.80E+01	4.00E+01
	rank	2	4	**1**	3
*f*22	Mean	4.35E+03	8.57E+02	**1.09E+02**	2.45E+03
	Std	3.28E+02	3.24E+02	1.91E+01	1.11E+03
	rank	4	2	**1**	3
*f*23	Mean	7.44E+03	6.20E+03	5.06E+03	**4.94E+03**
	Std	2.99E+02	1.09E+03	3.93E+02	1.31E+03
	rank	4	3	2	**1**
*f*24	Mean	**2.00E+02**	2.63E+02	2.69E+02	2.39E+02
	Std	5.04E-04	7.70E+00	6.95E+00	8.19E+00
	rank	**1**	3	4	2
*f*25	Mean	3.02E+02	**2.78E+02**	2.90E+02	**2.82E+02**
	Std	1.68E+01	8.41E+00	5.08E+00	2.18E+01
	rank	4	**1**	3	**2**
*f*26	Mean	2.04E+02	2.92E+02	**2.01E+02**	2.56E+02
	Std	7.87E-01	7.66E+01	3.70E-01	6.88E+01
	rank	2	4	**1**	3
*f*27	Mean	**3.01E+02**	8.59E+02	7.18E+02	6.60E+02
	Std	3.91E+00	8.47E+01	2.87E+02	7.68E+01
	rank	**1**	4	3	2
*f*28	Mean	3.00E+02	3.81E+02	3.00E+02	**3.00E+02**
	Std	2.61E-13	3.07E+02	7.20E-11	**2.29E-13**
	rank	2	4	3	**1**
Overall	**Average Rank**	2.82	2.82	2.39	**1.93**
Overall	**Final Rank**	3	3	2	**1**

In addition, we also adopt separately unimodal, multimodal, and composite functions to evaluate MIIBSO DE, PSO, and CLPSO. Table VIII provides their overall performance. This

indicates that MIIBSO wins the best overall performance on unimodal, multimodal, and composite functions.

TABLE VIII
COMPARISON OF MIIBSO, DE, PSO, AND CLPSO ON UNIMODAL, MULTIMODAL, AND COMPOSITION FUNCTIONS

Functions	Evaluation Criteria	DE	PSO	CLPSO	MIIBSO
Unimodal	Average Rank	2.2	3	3.2	**1.4**
	Final Rank	2	3	4	**1**
Multimodal	Average Rank	3	2.73	2.27	**2**
	Final Rank	4	3	2	**1**
Composition	Average Rank	2.5	3.13	2.25	**2.13**
	Final Rank	3	4	2	**1**

E. STATISTICAL ANALYSIS

To fully assess the performance of MIIBSO, we conducted the Wilcoxon signed-rank test [45] with a significance level of 0.05, which is a popular non-parametric statistical hypothesis test, equal to the dependent t-test, and employed to compare two correlated samples to evaluate the difference. In Table X, we can observe the statistical results between MIIBSO and the 13 algorithms on the 28 functions. Symbol "1" denotes that MIIBSO beats the compared algorithm on the same benchmark function; symbol "0" means that there is no significant difference in the same benchmark function between MIIBSO and the compared algorithm; symbol "-1" means that MIIBSO is significantly worse than the compared algorithm. As an example, we can observe that the results of the comparison between MIIBSO and ADMBSO include 15 symbols "1", nine symbols "0", and four symbols "-1" on the 28 functions. Fifteen symbols "1" confirm that MIIBSO has significantly better solutions than ADMBSO on 15 out of the 28 benchmark functions; nine symbols "0" mean that MIIBSO provides nine statistical equivalent results to ADMBSO on nine out of the 28 functions; four symbols "-1" indicates that MIIBSO has significantly worse results than ADMBSO on four out of the 28 benchmark functions.

As shown in Table X, MIIBSO conducts significantly better than the 13 algorithms on most benchmark functions according to the Wilcoxon signed-rank test. This further confirms that MIIBSO has better overall performance compared with the 13 algorithms.

TABLE X
COMPARISONS BETWEEN MIIBSO AND EACH OF 13 ALGORITHMS ON 28 CEC2013 FUNCTIONS WITH DIMENSION 30

	Pairwise Comparison: MIIBSO **Versus**												
Functions	**DE**	**PSO**	**CLPSO**	**BSO**	**MBSO**	**CLBSO**	**PPBSO**	**SBSO**	**BSODE**	**RGBSO**	**ADMBSO**	**BSO-DCS**	**BSOCO**
*f*1	0	+1	+1	+1	+1	+1	+1	0	+1	+1	+1	+1	+1
*f*2	+1	+1	+1	+1	-1	-1	+1	+1	+1	0	-1	0	+1
*f*3	-1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*4	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*5	0	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*6	0	+1	+1	+1	0	+1	+1	+1	+1	+1	+1	+1	+1
*f*7	-1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*8	0	0	0	0	+1	0	0	0	0	-1	0	-1	0
*f*9	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*10	-1	+1	+1	+1	+1	+1	+1	+1	0	-1	+1	-1	+1
*f*11	+1	0	-1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*12	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*13	+1	+1	+1	+1	0	0	+1	+1	+1	+1	0	+1	+1
*f*14	+1	-1	-1	+1	0	+1	+1	+1	+1	+1	+1	+1	+1
*f*15	+1	+1	-1	-1	0	0	-1	-1	-1	-1	0	-1	-1
*f*16	0	0	-1	-1	0	0	-1	-1	-1	-1	0	-1	-1
*f*17	-1	-1	-1	+1	-1	-1	+1	+1	+1	+1	-1	+1	+1
*f*18	+1	+1	0	+1	0	+1	+1	+1	+1	+1	+1	+1	+1
*f*19	+1	-1	-1	0	0	0	0	+1	+1	0	0	0	0
*f*20	+1	+1	+1	+1	-1	0	+1	+1	+1	+1	0	+1	+1
*f*21	0	0	0	+1	0	0	+1	0	+1	+1	0	+1	0
*f*22	+1	-1	-1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*23	+1	+1	0	+1	0	0	0	+1	0	+1	0	+1	0
*f*24	-1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*25	+1	0	0	+1	0	0	+1	+1	+1	+1	-1	+1	+1
*f*26	0	+1	0	0	0	0	+1	+1	0	0	-1	0	0
*f*27	-1	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
*f*28	0	+1	+1	+1	0	0	+1	+1	+1	+1	0	+1	+1
"+1"/ "0" /"-1"	**14/8/6**	**19/5/4**	**13/8/7**	**23/3/2**	**13/12/3**	**15/11/2**	**23/3/2**	**23/3/2**	**21/4/3**	**21/3/4**	**15/9/4**	**21/3/4**	**21/5/2**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2821118, IEEE Access

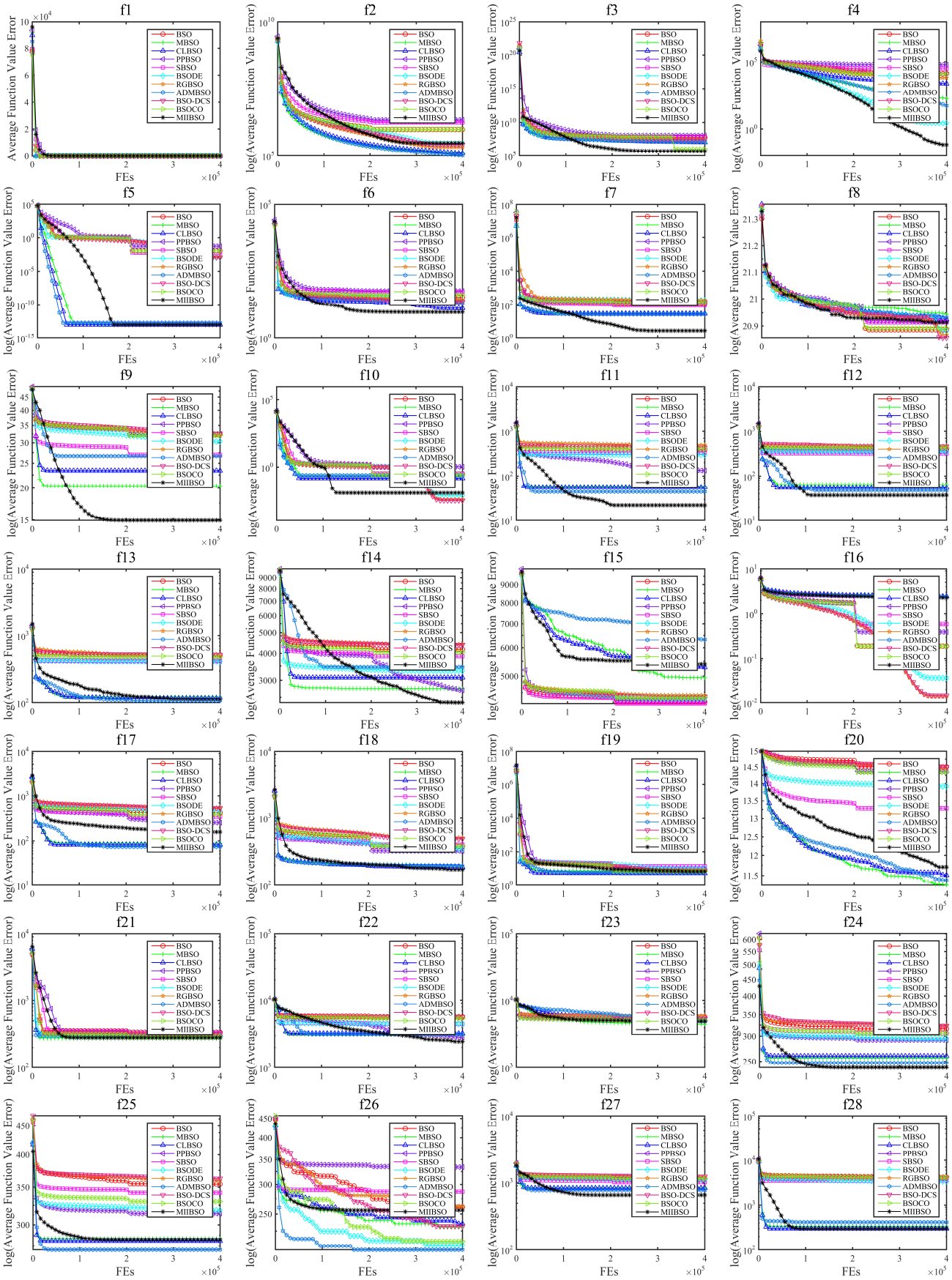

FIGURE 2. Convergence Curves of 11 algorithms on 28 CEC2013 functions in 30 dimensions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2821118, IEEE Access

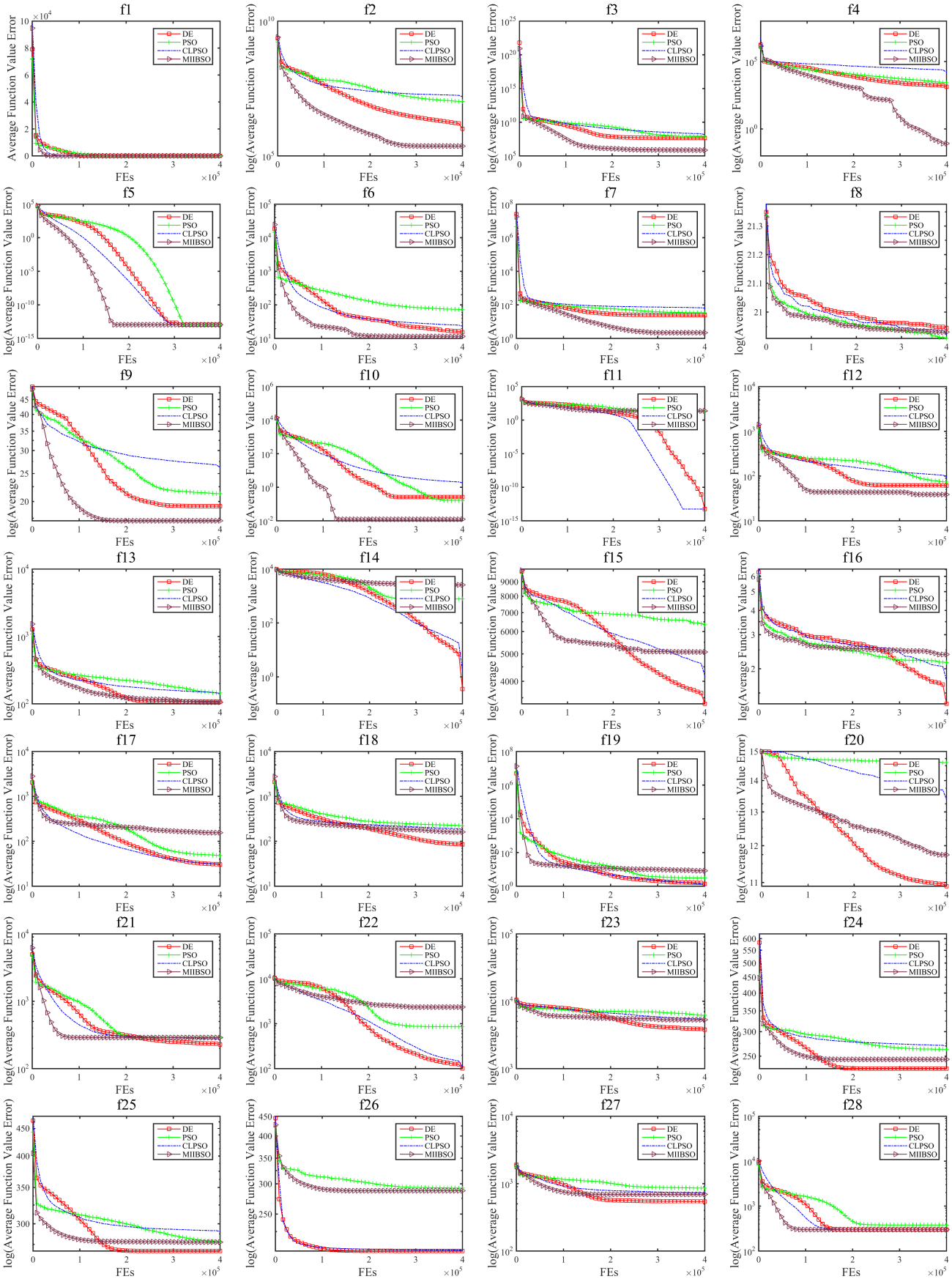

FIGURE 3. Convergence Curves of DE, PSO, CLPSO, and MIIBSO on 28 CEC2013 functions in 30 dimensions.

F. COMPARISON ON CONVERGENCE

We first compare the convergence speed of MIIBSO with these of the 10 BSO algorithms on the 28 functions. The convergence curves of all the algorithms are given in Fig. 2, where the horizontal axis denotes the maximum number of FEs, and the vertical axis is the base 2 logarithm of mean error values in 30 independent runs.

Fig. 2 show MBSO owns the best convergence performance on functions *f*20, *f*21, and *f*23. SBSO possesses the best on functions *f*1 and f15. CLBSO and RGBSO obtain the best on functions *f*19 and f16, respectively. ADMBSO yields the best on functions *f*2, *f*13, *f*17, *f*25, and *f*26. BSO-DCS offers the best on functions *f*8 and *f*10. MIIBSO algorithm provides the best convergence performance on functions *f*1, *f*3-*f*7, *f*9, *f*11, *f*12, *f*14, *f*18, *f*22, *f*24, *f*27, and *f* 28.

Interestingly, MIIBSO obtains the best on 13 out of the 28 functions. ADMBSO does the best on five out of the 28 functions. Each of the other nine BSO algorithms does the best on no more than three out of the 28 functions. Another Interesting thing is that although MIIBSO is not the best convergence speed on functions *f*8, *f*10, *f*13, *f*17, *f*19, *f*20, *f*21, *f*23, and *f*25, it renders relatively better convergence speed on them than other BSO algorithms.

Furthermore, we also compare the convergence speed of MIIBSO with that of DE, PSO, and CLPSO. The results are given in Fig. 3, where DE supplies the best convergence speed on functions *f*1, *f*3, *f*5, *f*7, *f*10, *f*24, and *f*27, PSO achieve the best on functions *f*8 and *f*25, CLPSO obtains the best on functions *f*11, *f*14-*f*17, *f*19, *f*21, *f*22, and *f*26. However, MIIBSO wins the best on 13 out of the 28 functions: *f*1, *f*2, *f*4, *f*6, f8, *f*9, *f*12, *f*13, *f*18, f20, *f*23, and *f*28 among four algorithms. Additionally, it also displays relatively better convergence performance on functions *f*3, *f*5, *f*7, *f*10, *f*15, *f*24, *f*25, and *f*27 compared with three other algorithms.

Briefly, the convergence analysis shows that MIIBSO provides the best convergence speed among the 14 algorithms.

G. SCALABILITY ANALYSIS

To verify whether the performance of the proposed MIIBSO algorithm degrades significantly with its increasing dimensions of the CEC2013 functions, we perform scalability analysis by comparing the proposed algorithm with eight algorithms on 30-D and 50-D functions. Eight algorithms include GBSO, ADMBSO, MBSO, PSO, CLPSO, ABC, DE, CMA-ES. Note that GBSO was a recently proposed BSO variant, which also exploits the dimension update scheme between individuals so that it has an outstanding global search capability. Both ADMBSO and MBSO also have excellent global search capability among most BSO variants from Table V. In addition, some representative algorithms (i.e., PSO, CLPSO, DE, ABC, and CMA-ES) are also applied to perform scalability analysis on 30-D and 50-D functions. The parameter settings of all the above nine algorithms are described in Table II. The population size is assigned to 50 except for CMA-ES. This is because the population size of CMA-ES relies on its problem dimension. Here, the population size of CMA-ES on 30-D and 50-D functions are set to $4 + \lfloor 3 \ln 30 \rfloor$ and $4 + \lfloor 3 \ln 50 \rfloor$, respectively. The maximum number of the Fes on 30-D and 50-D are set to 400000 and 600000, respectively. Likewise, each of the above nine algorithms is evaluated on each function of 28 CEC2013 functions with 30 independent operations.

Due to space limitation, the error mean and standard deviation values of 28 CEC2013 functions on 30-D and 50-D are shown in Section S-III of the supplementary file. Table XI only lists average and final ranks for the different algorithms on 30-D and 50-D problems. The best values are highlighted in bold. From Table XI, we observe that MIIBSO always achieves the best ranks among all nine algorithms on 30-D and 50-D problems.

To further validate the scalability performance of MIIBSO, the Wilcoxon signed-rank test with a significance level of 0.05 is conducted. Similarly, the statistical results between MIIBSO and the above eight algorithms on the 28 CEC2013 functions with both 30-D and 50-D problems are shown in Section S-III of the supplementary file because of space limitation. Here, Table XII merely shows the number of significantly better performance (symbol "1"), no significant difference (symbol "0"), and significantly worse performance (symbol "-1") between MIIBSO and each of the above eight algorithms on 30-D and 50-D CEC2013 functions. From Table XII, we observe that compared with each of these algorithms, MIIBSO constantly achieve the better performance on 30-D and 50-D problems. For instance, MIIBSO has better performance than GBSO on 30-D and 50-D problems.

TABLE XI
AVERAGE RANK AND FINAL RANK OF MIIBSO AND EIGHT ALGORITHMS ON 28 CEC2013 FUNCTIONS WITH DIMENSIONS 30 AND 50

Dimension	Evaluation Criteria	PSO	CLPSO	DE	ABC	CMA-ES	ADMBSO	MBSO	GBSO	MIIBSO
30-D	Average Rank	5.86	4.54	5.39	5.04	6.75	4.79	4.89	4.57	**3.18**
30-D	Final Rank	8	2	7	6	9	4	5	3	**1**
50-D	Average Rank	5.39	4.46	5.61	5.07	6.18	4.64	5.29	4.32	**3.39**
50-D	Final Rank	7	3	8	5	9	4	6	2	**1**

TABLE XII
NUMBER OF SIGNIFICANTLY BETTER PERFORMANCE (SYMBOL "1"), NO SIGNIFICANT DIFFERENCE (SYMBOL "0"), AND SIGNIFICANTLY WORSE PERFORMANCE (SYMBOL "-1") BETWEEN MIIBSO AND EACH OF EIGHT ALGORITHMS ON CEC2013 FUNCTIONS WITH ON DIMENSIONS 30 (30-D) AND 50 (50-D)

	Pairwise Comparison: MIIBSO **Versus**								
	Dimension	**PSO**	**CLPSO**	**DE**	**ABC**	**CMA-ES**	**ADMBSO**	**MBSO**	**GBSO**
"+1"/ "0" /"-1"	30-D	19/5/4	13/8/7	14/8/6	15/3/10	19/5/4	15/9/4	13/12/3	15/6/7
"+1"/ "0" /"-1"	50-D	20/2/6	15/4/9	15/4/9	16/3/9	20/2/6	16/9/3	20/4/4	12/10/6

H. IMPACT OF INDIVIDUAL COMPONENTS OF PROPOSED ALGORITHM

To explicitly observe the impact of the MII strategy, the RG strategy, and the DDS function on MIIBSO, we develop three variants of MIIBSO: MIIBSO-01, MIIBSO-02, and MIIBSO-03. MIIBSO-01 takes advantage of the K-means approach to replace the RG strategy of MIIBSO; MIIBSO-02 employs the individual update pattern of the original BSO to replace the MII strategy; MIIBSO-03 substitutes the DDS strategy of MIIBSO with the RDS strategy because both ADMBSO and CLBSO adopt the random difference step (RDS) to achieve the second-best and third-best performance amongst all the BSO algorithms, respectively. By comparing MIIBSO algorithm with the above MIIBSO variants and the original BSO on the 28 CEC2013 functions, we can clearly distinguish the effects of MII, RG, and DDS on MIIBSO. The five algorithms run 30 times independently on each function with dimension 30. The population size, the maximum number of Fes, and the maximum iterative number are still designated to 50, 400000, and 8000, respectively. Each algorithm is ranked via error mean and standard deviation value of each benchmark function. The final rank of each algorithm can be ascertained according to the average rank on the 28 functions. The average running time of each algorithm is defined as the average amount of time spent by the algorithm in running 30 times independently on the 28 functions. Owing to space limitation, the detailed results are shown in the Section-II of the supplementary file including error mean values, standard deviation values, average ranks, final ranks, running time of each algorithm on each function.

Table XIII distinctly lists similarities and differences among five algorithms with their average ranks, final ranks and average running time. An interesting observation is that MIIBSO achieves the best overall performance on the 28 functions among five algorithms. This indicates that the combination of MII, RG, and DDS is the best among all the algorithms, greatly improving the overall performance of MIIBSO. On the contrary, BSO achieves the worst overall performance on the 28 functions because it does not use such a combination.

TABLE XIII
EVALUATION OF MII STRATEGY, RG STRATEGY, AND DDS FUNCTION

Algorithm	Individual clustering	Individual Update	Step Size	Average Rank	Final Rank	Average Running Time
BSO	K-means	Individual Update Pattern of BSO	Step Size of BSO	4.25	5	76.8
MIIBSO01	K-means	MII	DDS	2.18	2	59.0
MIIBSO02	RG	Individual Update Pattern of BSO	DDS	3.96	4	**26.0**
MIIBSO03	RG	MII	RDS	2.5	3	29.0
MIIBSO	RG	MII	DDS	**2.11**	**1**	27.1

We also separately compare different MIIBSO variants with MIIBSO. Table XIII shows that although only one distinction between them is that MIIBSO uses the RG strategy rather than the K-means method which the MIIBSO01 does, MIIBSO owns the better overall performance than MIIBSO01 on the 28 functions. This implies that RG provides a better overall performance for MIIBSO compared with K-means. Afterwards, we also compared MIIBSO with MIIBSO02 on the 28 functions. The results imply that compared with the individual update pattern of BSO, the MII strategy offers a better overall performance for MIIBSO. Eventually, a comparison between MIIBSO and MIIBSO03 confirms that DDS contribute to a better improvement in the performance of MIIBSO compared with RDS.

A further observation shows the average ranks of MIIBSO, MIIBSO01, MIIBSO03, and MIIBSO02 are 2.11, 2.18, 2.5, and 3.96, respectively. The first three algorithms provide the similar ranks. However, MIIBSO02 achieves the worst average rank among the four algorithms. A vital factor in this situation is that MIIBSO, MIIBSO01, and MIIBSO03 all use the MII strategy, but MIIBSO02 does not. Note that MIIBSO02 also uses RG and DDS. This implies that compared with RG and DDS, the MII strategy plays a vital role in the improvement of MIIBSO.

In addition, each algorithm offers the corresponding average running time in seconds. Only these algorithms that utilize the RG strategy provide low computational costs. Therefore, the RG strategy can greatly improve the computational cost of MIIBSO.

In summary, the collaboration of MII, RG, and DDS contributes to the effective improvement in the overall performance of MIIBSO.

V. DISCUSSION

The original BSO algorithm failed to use some potential information interaction patterns between individuals for the individual update, thereby suffering from the premature convergence for complex optimization problems, such as the CEC 2013 test suit [40]. To address this problem, we have proposed a novel MIIBSO algorithm with the MII strategy, the RG strategy, and the DDS function. Experimental results show that MIIBSO obtains the best overall performance among all the 14 algorithms shown in Tables V and VII on the unimodal, multimodal, and composite functions based on the mean, standard deviation, statistical analysis, and convergence results.

Furthermore, we conduct scalability analysis to validate whether or not the performance of MIIBSO degenerates notably along with its increasing dimensions of the CEC2013 functions, by comparing it with eight algorithms shown in Table XI or XII on 30-D and 50-D CEC2013 functions. Experimental results confirm that MIIBSO has the best scalability performance among them.

In addition, we also evaluate the impacts of the MII strategy, the RG strategy, and the DDS function on MIIBSO. Experimental results show that the MII strategy plays an essential part in improving the global search capability of MIIBSO; RG strategy greatly improve the computational cost and enhance the global search capability for MIIBSO; DDS can further improve the balance between global and local search capability for MIIBSO.

In summary, MIIBSO achieves good global search capability and convergence speed, as well as decreases computational cost due to the combination of MII, RG, and DDS.

In contrast to most of the existing BSO variants, MIIBSO possesses the advantages and characteristics as follows:

Firstly, the MII strategy includes three new MII patterns with the ISF mechanism. Such patterns not only supply various information interactions between individuals, but also provide information interactions between the corresponding dimensions of individuals. This indicates that more potential solution information can be obtained for new individuals. Furthermore, the ISF mechanism is allowed to provide an efficient collaboration for the three patterns. Since the MII pattern I and the MII pattern II focus on local and global search, respectively, the ISF mechanism rationally uses a probability to select one of them, causing the suitable balance between the local and global search capability. The MII III can provide abundant information interactions to enhance the global search capability of MIIBSO. On the other hand, if the MII patterns III is conducted for each idea in each iteration, it has high computational costs due to using the dimensional update pattern. To achieve a compromise between the global search capability and the computational cost, the ISF mechanism utilizes rationally the MII pattern III via the stagnation number. To be more specific, once both the MII pattern I and II are unable to improve the new idea for successive 20 iterations, the ISF mechanism is to allocate the MII pattern III to improve the premature convergence. Consequently, the MII strategy can contribute to the significant improvement in the global search capability and convergence speed of MIIBSO.

Secondly, the RG strategy is introduced to replace the combination of the K-means method and the cluster center disruption in the original BSO algorithm because the individuals are grouped randomly rather than accurately based on the distances between individuals. Hence, the RG strategy combined with the MII strategy can further enrich the diversity of the information interactions and decrease the computational cost for MIIBSO.

Thirdly, the DDS function uses the dynamic difference step-size to adjust search ranges and provide feedback information for updating new individuals, which can further improve the global and local search capability for MIIBSO.

In general, the proposed MIIBSO algorithm effectively enhances the global search capability, accelerates the convergence speed, and decreases computational cost by combining the MII strategy with the RG strategy and DDS function.

Notably, although both MIIBSO and GBSO adopt some similar strategies including the individual stagnation mechanism, dynamic step size, and individual dimension update, they are fundamentally different as follows: (1) MIIBSO provides the information interaction pattern between the corresponding dimensions of two individuals for improving the individual stagnation when an individual has not been improved for successive iterations; however, GBSO adopts the difference update or re-initialization strategy by a probability when an individual generates successive stagnations. (2) MIIBSO uses (14) as dynamic difference step size function, where $(x_{mj} - x_{nj})$ plays key roles in supplying feedback information and improving sufficient search ranges. Specially, in the early iterative search of MIIBSO, $(x_{mj} - x_{nj})$ reflects a large difference between x_{mj} and x_{nj}, contributing to enhance the global search capability. Conversely, as x_{mj} and x_{nj} gradually converge towards the global optimal value in the late iterative process of MIIBSO, $(x_{mj} - x_{nj})$ offers a gradually smaller difference, contributing to refine a local search. In GBSO, the dynamic step size is expressed as $\delta(k)N(\mu,0)$, where $N(\mu,0)$ represents a Gaussian distribution and cannot provide sufficient feedback information between x_{mj} and x_{nj}. As an example, in the late iteration process of GBSO, $\delta(k)$ can provide a small value, however, $N(\mu,0)$ may generate a sufficiently large random number. This may lead to the very large value of $\delta(k)N(\mu,0)$ so that the individuals may not achieve good enough convergence performance; (3) In MIIBSO, the dimension update scheme is inspired by that of the CLPSO algorithm. For each dimension update of an individual X, some individuals are first randomly selected from the entire swarm, then the individual with the best fitness value is selected, and each dimension of the individual X is replaced by the corresponding dimension of the individual with the best fitness value. However, the GBSO algorithm conducts the dimension update scheme by using the probability selection mechanism instead of the fitness value comparison used in MIIBSO. For each dimension update of an individual X in GBSO, a new individual Y is first generated from a cluster or one combination of two individuals from two cluster by using the probability selection. Subsequently, each dimension of the individual X is replaced by the corresponding dimension of the individual Y; (4) Unlike GBSO that merely adopts the dimension update scheme, MIIBSO also includes the information interaction between individuals from one cluster, between individuals from two clusters; (5) Different from GBSO that uses a clustering strategy based on the fitness values, MIIBSO uses the RG strategy as its clustering strategy.

In addition, it should be noted that although MIIBSO provides the best overall performance on CEC2013 functions among all the 14 algorithms, MIIBSO is unable to win the best result on each of all the functions. For instance, MIIBSO achieves the better performance than CLPSO on the 28 functions, but it does not provide relatively competitive results on functions $f11$ and $f14$ compared with CLPSO. One possible reason is that the CLPSO algorithm uses an exemplary learning strategy [41] with the dimension update mechanism to provide historical search experiences and escape the local optimum of the functions $f11$ and $f14$.

In the future, we will consider how to combine various exemplary learning strategies [41], [46]-[48] with the MIIBSO algorithm to further enhance both the global search capability and the convergence speed. Moreover, the proposed MIIBSO algorithm will be employed to tackle multi-objective optimization problems in Big Data [49]. Moreover, the MII strategy can also be combined with other population-based swarm intelligent algorithms such as DMSPSO [50], improved ABC [51], and HB [52] for optimizing haptic devices or soft tissue modeling in virtual surgery.

VI. CONCLUSION

In this paper, we propose a MIIBSO algorithm, which uses the combination of the MII strategy, the RG strategy, and DDS function to enhance the information interactions capability, improve the global search capability, and decrease the computational cost. The MII strategy provides multiple information interaction patterns so that it greatly enhances the global search capability and convergence performance of MIIBSO. Furthermore, since the RG strategy is used, it can greatly improve the computational cost of MIIBSO, as well as enrich the diversity of the information interactions. In addition, the DDS function dynamically regulates the search range and provide feedback information, and thus further improves the balance between the global and local search capability. By combining the MII strategy, the RG strategy, and the DDS function, MIIBSO achieves the good performance.

REFERENCES

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, Nagoya, Japan, 1995, pp. 39–43.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, 1995, pp. 1942–1948.

[3] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, 1997.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. SMC*, vol. 26, no. 1, pp. 29–41, 1996.

[5] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report-TR06-Erciyes University, October, 2005.

[6] W. T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example", *Knowl.-Based Syst*. vol. 26, no. 2, pp. 69–74, 2012.

[7] Y. Shi, "Brain storm optimization algorithm," in *Proc. 2nd Int. Conf. Swarm Intell.*, Chongqing, China, Jun. 12–15, 2011, pp. 303–309.

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

[9] Y. Sun, "A hybrid approach by integrating brain storm optimization algorithm with grey neural network for stock index forecasting," *Abstract and Applied Analysis*, pp.1-10, 2014.

[10] Y. Shi, J. Xue, and Y. Wu, "Multi-objective optimization based on brain storm optimization algorithm," *International Journal of Swarm Intelligence Research*, vol. 4, no. 3, pp. 1–21, 2013.

[11] K. Lenin, B. R. Reddy, and M. SuryaKalavathi, "Brain storm optimization algorithm for solving optimal reactive power dispatch problem," *Int J Res Electron Commun Technol* vol. 1, no. 3, pp. 25–30, 2014

[12] M. Arsuaga-Ríos and MA. Vega-Rodríguez. "Multi-objective energy optimization in grid systems from a brain storming strategy," *Soft comput.*, vol. 19, no. 11, pp. 3159–3172, 2015.

[13] A. R. Jordehi, "Brainstorm optimization algorithm (BSOA): An efficient algorithm for finding optimal location and setting of FACTS devices in electric power systems," *Int. J. Elec. Power*, vol. 69, pp. 48-57, 2015.

[14] X. Guo, Y. Wu, L. Xie, S. Cheng, and J. Xin, "An adaptive brain storm optimization algorithm for multi-objective optimization problems". International Conference in Swarm Intelligence, Springer International Publishing pp.365-372, 2015.

[15] S. Cheng, Q. Qin, J. Chen, G. G. Wang, and Y. Shi, "Brain storm optimization in objective space algorithm for multimodal optimization problems". Advances in Swarm Intelligence. Springer International Publishing, 2016.

[16] R. Roy and J. Anuradha, "A modified brainstorm optimization for clustering using hard c-means". *IEEE International Conference on Research in Computational Intelligence and Communication Networks.*, pp. 202-207, 2016.

[17] A. Cervantes-Castillo and E. Mezura-Montes, "A study of constraint-handling techniques in brain storm optimization". *Evol. Comput*, pp.3740-3746, 2016.

[18] Z. Zhan, J. Zhang, Y. H. Shi, and H. L. Liu, "A modified brain storm optimization," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, Australia, Jun. 2012, pp. 1–8.

[19] H. Qiu and H. Duan, "Receding horizon control for multiple UAV formation flight based on modified brain storm optimization," *Nonlinear Dyn.,* vol.78, no.3, pp.1973–1988, 2014.

[20] 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," *Int. Conf. Adv. in Swarm Intell.*, 2015, pp. 357–364.

[21] H. Zhu and Y. Shi, "Brain storm optimization algorithms with k-medians clustering algorithms". Seventh International Conference on Advanced Computational Intelligence, pp. 107-110, 2015

[22] J. Chen, S. Cheng, Y. Chen, Y. Xie, and Y. Shi, "Enhanced Brain Storm Optimization Algorithm for Wireless Sensor Networks Deployment," in *Proc. 6th Int. Conf. Adv. in Swarm & Comput. Intell.,* Springer-Verlag New York, Vol. 9140, pp. 373-381, 2016.

[23] Z. Cao, X. Rong, and Z. Du, "An improved brain storm optimization with dynamic clustering strategy," *2016 the 3th Int. Conf. on Mec. & Mech. Eng.*, 2016, pp. 1–6.

[24] D. Zhou, Y. Shi, and S. Cheng, "Brain storm optimization algorithm with modified step-size and individual generation," *Int. Conf. Adv. in Swarm Intell.,* Springer-Verlag, 2012, pp.243-252.

[25] J. Li and H. Duan, "Simplified brain storm optimization approach to control parameter optimization in F/A-18 automatic carrier landing system," *Aerosp. Sci. Technol*., vol. 42, pp. 187-195, 2015.

[26] C. Sun, H. Duan, and Y. Shi, "Optimal satellite formation reconfiguration based on closed-loop brain storm optimization," *IEEE Comput. Intell. M.,* vol. 8, no. 4, pp. 39–51, 2013.

[27] Z. Cao, L. Wang, X. Hei, Y. Shi, and X. Rong, "An improved brain storm optimization with differential evolution strategy for applications of ANNs," *Math Problems Eng.*, pp. 1–18, 2015.

[28] Z. Yang and Y. Shi, "Brain storm optimization with chaotic operation". Seventh International Conference on Advanced Computational Intelligence, pp.111-115,2015.

[29] K. R. Krishnanand, S. M. F. Hasani, B. K. Panigrahi, and S. K. Panda, "Optimal Power Flow Solution Using Self–Evolving Brain–Storming Inclusive Teaching–Learning–Based Algorithm". Advances in Swarm Intelligence. Springer Berlin Heidelberg, pp. 338-345, 2013.

[30] H. Duan, S. Li, and Y. Shi, "Predator-prey based brain storm optimization for DC brushless motor," *IEEE Trans. Magn.*, vol. 49, no. 10, pp. 5336–5340, 2013.

[31] H. Duan and C. Li, "Quantum–behaved brain storm optimization approach to solving loney's solenoid problem," *IEEE Trans. Magn*., vol. 51, no.1, pp.1–7, 2015.

[32] Z. Song, J. Peng, C. Li, and P. X. Liu, "A Simple Brain Storm Optimization Algorithm with a Periodic Quantum Learning Strategy," in IEEE Access, vol. PP, no. 99, pp. 1-1, 2017. doi: 10.1109/ACCESS.2017.2776958

[33] Y. Yang, Y. Shi, and S. Xia, "Advanced discussion mechanism-based brain storm optimization algorithm," *Soft. Computing*, vol.19, no.10, pp. 2997–3007, 2015.

[34] Z. Jia, H. Duan, and Y. Shi, "Hybrid brain storm optimization and simulated annealing algorithm for continuous optimization problems," *Int. J. Bio–Inspired Comput*., vol.8, no.2, pp.109–121, 2016.

[35] M. El-Abd, "Global-best brain storm optimization algorithm," *Swarm Evol Comput*, vol. 37, pp. 27-44, 2017.

[36] S. Cheng, Y. Shi, Q. Qin, Q. Zhang, and R. Bai, "Population diversity maintenance in brain storm optimization algorithm," *J. Artif. Intell. Soft. Comput Res*., vol.4, no. 2, pp. 83–97, 2014.

[37] M. El-Abd, "Brain storm optimization algorithm with re-initialized ideas and adaptive step size," *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, 2016, pp. 2682-2686.

[38] Z. Zhou, H. Duan, and Y. Shi, "Convergence analysis of brain storm optimization algorithm". *Evol. Comput.* pp. 3747-3752, 2016.

[39] S. Cheng, Q. Qin, J. Chen J, and Y. Shi, "Brain storm optimization algorithm: a review", *Artificial Intelligence Review*, vol. 46, no. 4, pp. 445-458, 2016.

[40] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," *Comput. Intell. Lab.*, Zhengzhou Univ., Zhengzhou, China, Tech. Rep. Dec. 2012, Jan. 2013.

[41] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particles swarm optimization for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[42] A. F. Osborn, "*Applied Imagination: Principles and Procedures of Creative Problem Solving,"* New York, NY, USA, 1963.
[43] N.Hansen, A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies", *Evol. Comput.*, vol.9, no.2, pp.159-195, 2001
[44] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *J. Glob. Optim.,* vol. 39, no. 3, pp.459-471, Nov. 2007.
[45] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.,* vol. 1, no. 1, pp. 3 – 18, Mar. 2011.
[46] Z. H. Zhan, J. Zhang, Y. Lin, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.* vol.15, no. 6, pp. 832–847, Dec. 2011.
[47] Z. Ren, A. Zhang, C. Wen, and Z. Feng, "A scatter learning particle swarm optimization algorithm for multimodal problems," *IEEE Trans. Cybern.* Vol. 44, no. 7, pp. 1127–1140, Jul. 2014.
[48] Y. J. Gong, J. J. Li, Y. C Zhou, Y. Li, H. S. H. Chung, Y. H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.,* vol. 46, no. 10, pp. 2277–2290, Oct. 2016.
[49] J. Wu, S. Guo, J. Li and D. Zeng, "Big Data Meet Green Challenges: Big Data Toward Green Applications," *IEEE Sys J*, vol. 10, no. 3, pp. 888-900, Sept. 2016.
[50] S. Z. Zhao, P. N. Suganthan and S. Das, "Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search," *IEEE Congress on Evolutionary Computation*, Barcelona, 2010, pp. 1-8.
[51] S. Das, S. Biswas, S. Kundu, "Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization," *Appl Soft Comput*, vol. 13, no. 12, pp. 4676-4694, 2013.
[52] A. Rajasekhar, N. Lynn, S. Das, P.N. Suganthan, "Computing with the collective intelligence of honey bees – A survey," *Swarm Evol Comput*, vol 32, pp. 25-48, 2017.