IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 8, 2018, accepted November 30, 2018, date of publication December 4, 2018, date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2884862

A Vector Grouping Learning Brain Storm Optimization Algorithm for Global Optimization Problems

CHUNQUAN LI[1,2], DUJUAN HU[1], ZHENSHOU SONG[1], FENG YANG[1], ZU LUO[1], JINGHUI FAN[1], AND PETER X. LIU[1,2], (Senior Member, 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

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

This work was supported in part by the National Natural Science Foundation of China under Grant 61503177, Grant 81660299, and Grant 61863028, in part by the China Scholarship Council through the State Scholarship Fund under Grant 201606825041, in part by the Science and Technology Department of Jiangxi Province of China under Grant 20161ACB21007, Grant 20171BBE50071, and Grant 20171BAB202033, and in part by the Education Department of Jiangxi Province of China under Grant GJJ14228 and Grant GJJ150197.

ABSTRACT The original brain storm optimization (BSO) method does not rationally compromise global exploration and local exploitation capability, which results in the premature convergence when solving complicated optimization problems such as the shifted or shifted rotated functions. To address this problem, this paper develops a vector grouping learning BSO (VGLBSO) method. In VGLBSO, the individuals' creation based on a VGL scheme is first developed to improve the population diversity and compromise the global exploration and local exploitation capability. Moreover, a hybrid individuals' update scheme is established by reasonably combing two different individuals' update schemes, which further compromises the global exploration and local exploitation capability. Finally, the random grouping scheme, instead of K-means grouping, is allowed to shrink the computational cost and maintain the diversity of the information exchange between different individuals. Twenty-eight popular benchmark functions are used to compare VGLBSO with 12 BSO and nine swarm intelligence methods. Experimental results present that VGLBSO achieves the best overall performance, including the global search ability, convergence speed, and scalability among all the compared algorithms.

INDEX TERMS Brain storm optimization (BSO), vector grouping learning (VGL), swarm intelligence (SI), global optimization.

I. INTRODUCTION

Global optimization is of great significance in many scientific and engineering problems. However, there are still some complex landscapes of many global problems such as non-convex, non-differentiable, and multimodal optimization problems. Traditional gradient-based algorithms [1] do not work for most of these problems such as non-differentiable problems. Hence, scholars try to develop swarm intelligence algorithms to solve such problems. The swarm intelligence algorithms are derived from the simulation of nature-inspired behaviors such as biological foraging or biological evolution, as well as can provide good performance in solving complex problems such as non-convex, non-differentiable, and multimodal optimization problems. Owing to such advantages, many swarm intelligence algorithms have been presented such as ant colony optimization (ACO) [2], particle swarm optimization (PSO) [3], artificial bee colony (ABC) [4], firefly algorithm (FA) [5], bat algorithm (BA) [6], fruit fly optimization algorithm (FOA) [7], the genetic algorithm (GA) [8], culture algorithm (CA) [9], differential evolution (DE) [10], covariance matrix adaptation evolution strategy (CMA-ES) [11], and fireworks algorithm (FWA) [12].

The brain storm optimization (BSO) algorithm [13] is a newly developed swarm intelligence technique motivated by the human brainstorming process where a crowd of people comes together to promote the generation of new ideas for dealing with the thorny issues. Specifically, in the BSO, each individual of the entire swarm is regarded as an idea of the brainstorming process. During each iterative search process, all individuals are classified by grouping operation such as K-means grouping method; the best individual in each group serves as the center of the corresponding group. After that, each new individual is created via the information exchange between the individuals from the same group or two different groups. Finally, the new individuals are updated by the logarithmic sigmoid function with a Gaussian random number. In short, the BSO algorithm has three major processes: individuals' grouping, individuals' creation, and individuals' update. Furthermore, as a competitive swarm intelligence technique, BSO has been widely used in many scientific issues and engineering applications such as wireless sensor networks [14], multiple satellites with impulse control [15], energy consumption [16], locating multiple optima [17], and multi-objective optimization [18], the optimization of a grey neural network [19], power dispatch problem [20], energy optimization in grid systems [16], electric power systems [21], and grouping problem [22].

Actually, as a recently proposed swarm intelligence technique, the BSO algorithm should simultaneously offer promising global exploration and local exploitation capability to locate the global optimal solution [23]. In particular, the global exploration capability signifies that the BSO is capable of exploring many prospective solution domains; however, the local exploitation capability implies that it is capable of exploiting the optimal solution to the prospective solution domains explored and improving the search accuracy of solutions. Overemphasizing the global exploration capability might waste too much time in seeking inferior solution areas, resulting in the decline of the convergence performance; on the other hand, overemphasizing the local exploitation capability might trigger the loss of population diversity during the early stages of the entire iterative process, thereby leading the population to sink into local optima. Consequently, as for the BSO, how to implement a rational compromise between the global exploration and local exploitation capability is a challenging issue. Besides, since K-means grouping [13] needs to compute distances between different individuals, it makes the BSO consume a very high computational cost on grouping operation.

To address those problems, many BSO variants have recently been proposed to enhance the performance of the original BSO. Most investigations in the performance improvement of the original BSO mainly focus on the one or more of the following three respects: the individuals' creation (IC), the individuals' update (IU), and the individuals' grouping (IG), concisely reviewed as follows.

Recently, some efforts have been made in the improvement of IC so as to enhance global exploration and local exploitation capability. Reference [24] presented a simple BSO (SBSO) algorithm, where the IC modes are simplified to an operational mode to increase the local exploitation capability and convergence speed. Reference [25] adopted the mutation and crossover strategy of the differential evolution technique to improve the IC scheme and offer the balance between exploration and exploitation. Reference [26] developed an advanced discussion mechanism-based BSO (ADMBSO), by introducing inter-group and intra-group discussing schemes to optimize the IC scheme, enrich the population diversity, and highlight both the global exploration and local exploitation. Recently, [27] put forward a global BSO (GBSO) algorithm by utilizing the individual dimension information interaction scheme between individuals to enhance the population diversity and improve the global exploration capability. Most recently, [28] developed a BSO with multi-information interaction (MIIBSO) method, covering individual information and individual dimension information interaction scheme to enhance information interaction ability and improve the global exploration capability.

Moreover, various individuals' update schemes have been developed to compromise global exploration and local exploitation capability. In [29], a modified BSO (MBSO) was proposed by using the idea differential strategy (IDS) to compromise the local and global search ability. Reference [30] developed a new individuals' update scheme called modified step-size based on a batch-mode to update new individuals, contributing to compromise global exploration and local exploitation. Reference [15] introduced three differential evolution scheme as step-size functions in closed-loop brain storm optimization (CLBSO) algorithms, which avoids the individuals to sink into local optima and accelerates the convergence speed. Reference [31] proposed a new individuals' update scheme based on a predator-prey strategy to improve the global exploration capability. In [32], an individuals' update with quantum-behaved mechanism was designed to enrich the diversity of the population, enhance the global exploration capability, and avoid the premature convergence. In [33], a chaotic search technique with probability update strategy was employed to improve the individuals' update scheme and avoid sinking to local optima. Recently, [34] invented a quantum-behaved individual update with periodic learning (QBIU-PL) strategy to improve the diversity of newly generated individuals and enhance global exploration and local exploitation capability.

In addition, some new individuals' creation schemes are introduced into the original BSO to improve the performance of K-means grouping. In [29], a simple grouping method (SGM) is invented to replace the K-means grouping scheme of the original BSO, causing the decline of the computational cost. Reference [35] used the K-medians grouping, instead of K-means, to decline the impact of the group centers. Reference [36] adopted the fitness values of individuals to replace the distance between individuals as a grouping standard, which can enhance the grouping efficiency. Reference [37] presented a dynamic K-means grouping and it is able to decrease the calculation cost and strengthen the exploration ability. Reference [38] employed affinity propagation (AP) grouping scheme instead of the K-means, which can dynamically adjust the number of groups according to iterative search conditions. Reference [39] proposed a random grouping (RG) scheme in place of the k-means grouping method to decrease the calculation burden of the K-means grouping and enhance the global exploration capability.

As stated previously, those BSO variants have provided a variety of improvements in the individuals' creation, individuals' update, or individuals' grouping, aiming to regulate the balance between the global exploration and local exploitation or decrease the computational cost. However, they may still need further improvements. For instance, for ADMBSO, the individuals' creation scheme is upgraded using information exchange between the individuals of the inter-group or intra-group; however, ADMBSO neglects the information interchange based on the individuals' dimensions, which might result in the lack of the population diversity and deteriorate the global exploration capability. On the other hand, for GBSO, the IC scheme can offer the information exchange based on the individuals' dimensions to enhance the exploration capability, whereas overemphasizing the information exchange between individuals' dimensions might cause numerous meaningless explorations, thereby affecting the convergence speed. Furthermore, in the original BSO, the individuals' update scheme employs the logarithmic sigmoid function with a Gaussian random number to emphasize the global exploration capability, however, it is unable to offer sufficient local exploitation performance. Conversely, for some BSO variants such as MBSO [29] and CLBSO [15], their individuals' update schemes employ the differential evolution strategy to increase the convergence speed, whereas they fail to supply suitable global exploration capability when tackling the complicated global problems [40].

In order to better balance the global exploration and local exploitation, the paper presents a vector grouping learning BSO (VGLBSO) algorithm. In VGLBSO, a new IC scheme, called the IC based on vector grouping learning (IC-VGL) scheme is first proposed to offer the rational diversity of the population and compromise the global exploration and local exploitation for VGLBSO. Unlike the individuals' creation scheme of most existing BSO algorithms, the IC-VGL scheme of VGLBSO stochastically splits the full dimensions of each individual in the whole swarm into sub-vectors that contain the partial dimensions; a new individual can be created by learning the sub-vectors of different individuals. Furthermore, a new hybrid individuals' update scheme for VGLBSO is developed by effectively combing two different individuals' update schemes affiliated with the original BSO and CLBSO, further compromising the global exploration and local exploitation capability. Besides, identical with the RGBSO, VGLBSO also utilizes the RG scheme instead of K-means grouping scheme to shrink the computational cost and maintain the diversity of the information exchange between different individuals.

The rest of the paper is arranged as follows. Section II illustrates the related works of BSO algorithms. Section III gives detailed descriptions for VGL-BSO. The experimental evaluations and discussions for VGL-BSO are demonstrated in Section IV. Section V gives conclusions.

II. RELATED WORKS

A. ORIGINAL BSO ALGORITHM

The original BSO is a novel swarm intelligence algorithm, proposed by Shi [13] in 2011. It consists of three fundamental schemes: the individuals' creation (IC), the individuals' update (IU), and the individuals' grouping (IG) scheme, as follows.

1) INDIVIDUALS' GROUPING SCHEME

Considering the D-dimensional solution space, we assume that the entire swarm of the original BSO comprises $\mathcal{N}$ individuals; each individual is called an idea and written as $\mathcal{X}_i^t = \left[x_{i1}^t, x_{i2}^t, \cdots, x_{i\mathcal{D}}^t\right]$, where $i \in \{1, 2, \cdots, \mathcal{N}\}$, and t signifies the current iteration number; the jth dimension of $\mathcal{X}_i^t$ is described as $x_{ij}^t \in \left[\ell_j, u_j\right], j \in \{1, 2, \cdots, \mathcal{D}\}$ where ℓ_j and u_j represent the lower and upper bound of the jth dimensional search space, respectively.

First, for each iteration search, the original BSO splits $\mathcal{N}$ ideas into $\mathcal{M}$ different groups using the K-means grouping scheme. For the mth group, $m \in \{1, 2, \cdots, \mathcal{M}\}$, the idea corresponding to the best fitness value is designated as the mth center, namely $\mathcal{C}_m^t = \left[c_{m1}^t, c_{m2}^t, \cdots, c_{mD}^t\right]$. Hence, all the $\mathcal{M}$ centers are given as $\left\{\mathcal{C}_1^t, \mathcal{C}_2^t, \cdots, \mathcal{C}_{\mathcal{M}}^t\right\}$.

Then, a group center $\mathcal{C}_a^t = \left[c_{a1}^t, c_{a2}^t, \cdots, c_{aD}^t\right]$, $a \in \{1, 2, \cdots, \mathcal{M}\}$, which is stochastically chosen from $\mathcal{M}$ group centers $\left\{\mathcal{C}_1^t, \mathcal{C}_2^t, \cdots, \mathcal{C}_{\mathcal{M}}^t\right\}$, is substituted by an individual generated by random initialization only if $r_{00} < p_{r00}$. Here, r_{00} is a random number, uniformly distributed in the interval [0, 1]; p_{r00} is a pre-specified probability value. Such a random substitution can diverge group centers to discover more promising solution regions and enhance the global search performance.

2) INDIVIDUALS' CREATION SCHEME

The original BSO employs the IC scheme to create new ideas, enhancing the population diversity. Note that p_{r0}, p_{r01}, and p_{r02} are three pre-specified probability values; r_0, r_{01}, and r_{02} are three random numbers, uniformly distributed in the interval [0, 1].

If $r_0 < p_{r0}$, a new individual $\mathcal{X}_{new\_i}^t$, $i \in \{1, 2, \cdots, \mathcal{N}\}$ is created by stochastically choosing an individual from a group as follows:

$$\mathcal{X}_{new\_i}^t = \begin{cases} \mathcal{X}_a^t, & r_{01} \geq p_{r01} \\ \mathcal{C}_a^t, & r_{01} < p_{r01} \end{cases} \tag{1}$$

where $\mathcal{X}_a^t$ is an individual stochastically chosen from the ath group; the ath group is stochastically chosen from one of $\mathcal{M}$ groups; $\mathcal{C}_a^t$ is the center of the ath group, $a \in \{1, 2, \cdots, \mathcal{M}\}$.

If $r_0 \geq p_{r0}$, a new individual $\mathcal{X}_{new\_i}^t$, $i \in \{1, 2, \cdots, \mathcal{N}\}$ is established by stochastically selecting two individuals from two different groups as follows:

$$\mathcal{X}_{new\_i}^t = \begin{cases} \mathcal{R} \odot \mathcal{X}_a^t + (1 - \mathcal{R}) \odot \mathcal{X}_b^t & r_{02} \geq p_{r02} \\ \mathcal{R} \odot \mathcal{C}_a^t + (1 - \mathcal{R}) \odot \mathcal{C}_b^t & r_{02} < p_{r02} \end{cases} \tag{2}$$

where individuals $\mathcal{X}_a^t$ and $\mathcal{X}_b^t$ are stochastically chosen from the ath and bth group, respectively; the ath and bth group are stochastically chosen from the two different groups of $\mathcal{M}$ groups; both $\mathcal{C}_a^t$ and $\mathcal{C}_b^t$ indicate the group centers of the ath and bth group, respectively; $a, b \in \{1, 2, \cdots, \mathcal{M}\}$ and $a \neq b$; $\mathcal{R}$ is defined as random vector $[r_1, r_2, \cdots, r_{\mathcal{D}}]$; for each r_j, $j \in \{1, 2, \cdots, \mathcal{D}\}$, it is a uniformly distributed random number within interval [0, 1]; $\odot$ represents Hadamard product; t denotes the current iteration number.

3) INDIVIDUALS' UPDATE SCHEME

For each iteration search in the original BSO, each individual is updated according to the following update equation:

$$\mathcal{X}_{\text{temp}\_i}^t = \mathcal{X}_{new\_i}^t + \mathbb{N}(\mu, \sigma)\,\Phi(t). \tag{3}$$

Here, $\mathcal{N}(\mu, \sigma)$ is defined as a Gaussian random vector, $[n_1(\mu, \sigma), n_2(\mu, \sigma), \cdots, n_{\mathcal{D}}(\mu, \sigma)]$; for each $n_j(\mu, \sigma)$, $j \in \{1, 2, \cdots, \mathcal{D}\}$, it is a Gaussian random number with mean μ and variance σ; $\Phi(t)$ is defined as a step size vector, $[\phi_1(t), \phi_2(t), \cdots, \phi_D(t)]$ where each $\phi_j, j \in \{1, 2, \cdots, \mathcal{D}\}$ is defined as

$$\phi_j(t) = r_j \text{logsig}\left[(0.5 \times \text{T-t})/\eta\right]. \tag{4}$$

Here, $\mathcal{T}$ and t are defined as the maximum and current iteration number, respectively; η is used for regulating the slope of function logsig[·] and improving the global and local search performance; each $r_j, j \in \{1, 2, \cdots, \mathcal{D}\}$ is a uniform random number within the interval [0, 1]

Furthermore, the selection scheme is executed to obtain the competitive individual of the entire swarm in the original BSO. Without loss of generality, the considered fitness function $\mathcal{F}$ is for minimization. Therefore, for $i \in \{1, 2, \cdots, \mathcal{N}\}$, the ith individual is chosen as

$$\mathcal{X}_i^{t+1} = \begin{cases} \mathcal{X}_{\text{temp}\_i}^t, & \mathcal{F}\left[\mathcal{X}_{\text{temp}\_i}^t\right] < \mathcal{F}\left[\mathcal{X}_i^t\right] \\ \mathcal{X}_i^t, & \mathcal{F}\left[\mathcal{X}_{\text{temp}\_i}^t\right] \geq \mathcal{F}\left[\mathcal{X}_i^t\right] \end{cases} \tag{5}$$

Here, $\mathcal{F}\left[\mathcal{X}_{\text{temp}\_i}^t\right]$ and $\mathcal{F}\left[\mathcal{X}_i^t\right]$ are the fitness function values of $\mathcal{X}_{\text{temp}\_i}^t$ an $\mathcal{X}_i^t$, respectively.

After the original BSO has performed the IU scheme for all individuals in each iteration, the termination criterion for the original BSO is checked. If such a criterion is satisfied, the original BSO is to cease the iteration search. Otherwise, the iteration search is to hold on.

B. INDIVIDUALS' CREATION SCHEME BASED ON INTER-GROUP AND INTRA-GROUP DISCUSSION

To balance the global exploration and local exploitation capability, ADMBSO used a new IC scheme based on the inter-group and intra-group discussion as follows.

If $r_0 < p_{r0}$, a new individual $\mathcal{X}_{new\_i}^t$, $i \in \{1, 2, \cdots, \mathcal{N}\}$ is established by stochastically choosing an individual from a group as follows:

$$\mathcal{X}_{new\_i}^t = \begin{cases} \mathcal{C}_a^t, r_{01} < p_{r01} \\ \mathcal{X}_{a_1}^t, r_{01} \geq p_{r01}, r_{11} \geq p_{r11} \\ \mathcal{R} \odot \mathcal{X}_{a_1}^t + (1 - \mathcal{R}) \\ \quad \odot \mathcal{X}_{a_2}^t, r_{01} \geq p_{r01}, r_{11} < p_{r11} \end{cases} \tag{6}$$

where both $\mathcal{X}_{a_1}^t$ and $\mathcal{X}_{a_2}^t$ are the individuals stochastically chosen from the ath group; the ath group is stochastically chosen from one of $\mathcal{M}$ groups; $\mathcal{C}_a^t$ is the center of the ath group, $a \in \{1, 2, \cdots, \mathcal{M}\}$; t denotes the current iteration number.

If $r_0 \geq p_{r0}$, a new individual $\mathcal{X}_{new\_i}^t$, $i \in \{1, 2, \cdots, \mathcal{N}\}$ is established by stochastically selecting an individual from two different groups as follows:

$$\mathcal{X}_{new\_i}^t = \begin{cases} \mathcal{L} + \mathcal{R} \odot (U - \mathcal{L}), & r_{02} < p_{r02} \\ \mathcal{R} \odot \mathcal{X}_a^t + (1 - \mathcal{R}) \odot \mathcal{X}_b^t, & r_{02} \geq p_{r02}, r_{12} \geq p_{r12} \\ \mathcal{R} \odot \mathcal{C}_a^t + (1 - \mathcal{R}) \odot \mathcal{C}_b^t, & r_{02} \geq p_{r02}, r_{12} < p_{r12} \end{cases} \tag{7}$$

where $\mathcal{L} = [l_1, l_2, \cdots, l_{\mathcal{D}}]$ and $U = [u_1, u_2, \cdots, u_{\mathcal{D}}]$ are the lower and upper bound vector of $\mathcal{D}$ dimension search space, respectively; l_j and $u_j, j \in \{1, 2, \cdots, \mathcal{D}\}$ represent the lower and upper bound of the jth dimensional search space, respectively; individuals $\mathcal{X}_a^t$ and $\mathcal{X}_b^t$ are stochastically chosen from the ath and bth group, respectively; the ath and bth group are stochastically chosen from the different two groups of $\mathcal{M}$ groups; both $\mathcal{C}_a^t$ and $\mathcal{C}_b^t$ indicate the group centers of the ath and bth group, respectively; $a, b \in \{1, 2, \cdots, \mathcal{M}\}$ and $a \neq b$; $\mathcal{R}$ is defined as random vector $[r_1, r_2, \cdots, r_{\mathcal{D}}]$; for each r_j, $j \in \{1, 2, \cdots, \mathcal{D}\}$, it is a uniformly distributed random number within interval [0, 1]; $\odot$ represents Hadamard product; t denotes the current iteration number.

Particularly, p_{r0}, p_{r01}, p_{r02}, p_{r11}, and p_{r12} are five pre-specified probability values; r_0, r_{01}, r_{02}, r_{11}, and r_{12} are five random numbers, uniformly distributed in the interval [0, 1].

C. INDIVIDUALS' CREATION SCHEME BASED ON INDIVIDUALS' DIMENSIONS

To improve the global exploration capability, GBSO used a new IC scheme based on individuals' dimension as follows.

If $r_0 < p_{r0}$, the dimension $x_{\text{new}\_ij}^t$, $j \in \{1, 2, \cdots, \mathcal{D}\}$ of a new individual $\mathcal{X}_{new\_i}^t$, $i \in \{1, 2, \cdots, \mathcal{N}\}$ is obtained based on the dimension x_{aj}^t of $\mathcal{X}_a^t = \left[x_{a1}^t, x_{a2}^t, \cdots, x_{a\mathcal{D}}^t\right]$, or c_{aj}^t of $\mathcal{C}_a^t = \left[c_{a1}^t, c_{a2}^t, \cdots, c_{a\mathcal{D}}^t\right]$ as follows:

$$x_{\text{new}\_ij}^t = \begin{cases} x_{aj}^t, & r_{01} \geq p_{r01} \\ c_{aj}^t, & r_{01} < p_{r01} \end{cases} \tag{8}$$

In formula (8), for each $x^t_{\text{new}\_ij}$, $j \in \{1, 2, \cdots, D\}$, group a, $a \in \{1, 2, \cdots, M\}$ is stochastically chosen from M groups; then, X^t_a is stochastically chosen from group a and C^t_a is the center of the group a; ultimately, either x^t_{aj} of X^t_a or c^t_{aj} of C^t_a is applied to create $x^t_{\text{new}\_ij}$; t denotes the current iteration number.

If $r_0 \geq p_{r0}$, the dimension $x^t_{\text{new}\_ij}$, $j \in \{1, 2, \cdots, D\}$ of a new individual $X^t_{new\_i}$, $i \in \{1, 2, \cdots, N\}$ is obtained based on the combination of x^t_{aj} and x^t_{bj} corresponding to X^t_a and X^t_b respectively, or the combination of c^t_{aj} and c^t_{bj} corresponding to C^t_a and C^t_b respectively as follows:

$$x^t_{\text{new}\_ij} = \begin{cases} r x^t_{aj} + (1-r)\, x^t_{bj}, & r_{02} \geq p_{r02} \\ r c^t_{aj} + (1-r)\, c^t_{bj}, & r_{02} < p_{r02} \end{cases} \quad (9)$$

In formula (9), for each $x^t_{\text{new}\_ij}$, $j \in \{1, 2, \cdots, D\}$, groups a and b, $a, b \in \{1, 2, \cdots, M\}$ are first stochastically chosen from M groups; then, $X^t_a = \left[x^t_{a1}, x^t_{a2}, \cdots, x^t_{aD}\right]$ and $X^t_b = \left[x^t_{b1}, x^t_{b2}, \cdots, x^t_{bD}\right]$ are stochastically chosen from groups a and b, respectively; $C^t_a = \left[c^t_{a1}, c^t_{a2}, \cdots, c^t_{aD}\right]$ and $C^t_b = \left[c^t_{b1}, c^t_{b2}, \cdots, c^t_{bD}\right]$ are the center of groups a and b, respectively.

Specially, p_{r0}, p_{r01}, and p_{r02} are three pre-specified probability values; r_0, r_{01}, r_{02}, and r are four stochastic numbers, uniformly distributed in the interval [0, 1].

Owing to the IC scheme based on individuals' dimension for $X^t_{new\_i}$ in (8) and (9), the GBSO can acquire the effective population diversity and global exploration capability.

III. PROPOSED VGLBSO ALGORITHM

VGLBSO contains the RG, IC-VGL, and H-IU scheme, illustrated as follows.

A. RANDOM GROUPING FOR INDIVIDUALS' GROUPING

For most of the existing BSO algorithms, their IG schemes like K-means, SGM, and K-medians, and AP grouping need to calculate the distances between different individuals, which can result in the high computational cost.

In VGLBSO, instead of employing the K-means grouping, the IG scheme employs the random grouping (RG) introduced from [39]. Because the RG scheme does not calculate the distance between any two different individuals for grouping the entire swarm, it has the low computational cost.

Additionally, the RG scheme can stochastically choose different individuals from the whole swarm for each group, so it can maintain the diversity of the information exchange between different individuals for the IC scheme.

The RG scheme is simple but effective, illustrated as follows.

All N individuals of the entire swarm are written as $\left\{X^t_1, X^t_2, \cdots, X^t_N\right\}$, and then stochastically sorted as $\left\{\mathbb{X}^t_1, \mathbb{X}^t_2, \cdots, \mathbb{X}^t_N\right\}$. These N individuals are further split into M groups. For $m \in \{1, 2, \cdots, M\}$, the mth group covers ρ individuals $\left\{\mathbb{X}^t_{(m-1)\rho+1}, \mathbb{X}^t_{(m-1)\rho+2}, \cdots, \mathbb{X}^t_{(m-1)\rho+\rho}\right\}$ where $\rho = N/M$. Furthermore, for the mth group, the individual corresponding the best fitness function value is selected as the mth group center $C^t_m = \left[c^t_{m1}, c^t_{m2}, \cdots, c^t_{mD}\right]$, $C^t_m \in \left\{C^t_1, C^t_2, \cdots, C^t_M\right\}$.

B. INDIVIDUALS' CREATION BASED ON VECTOR GROUPING LEARNING

The IC schemes of the existing BSO algorithms generally adopted two different ways for information exchange. One is the information exchange between individuals like the IC schemes of BSO and ADMBSO, which ignores the information interchange between the individuals' dimensions, leads to the loss of population diversity, and may decline the global exploration capability. The other is the information exchange between the individuals' dimensions such as the IC scheme of GBSO, which may cause numerous meaningless explorations and attenuate the exploitation capability.

To compromise the information exchange between the above two different information ways, we propose the IC-VGL scheme, which adopts the information exchange neither between individuals nor between individuals' dimensions, but the information exchange between the sub-vectors of individuals. IC-VGL consists of a stochastic vector grouping mechanism and two vector grouping learning patterns to maintain the rational compromise between the global exploration and local exploitation as follows.

1) STOCHASTIC VECTOR GROUPING MECHANISM

The stochastic vector grouping mechanism is developed to generate the sufficient sub-vectors that are employed to provide the rational diversity of information for creating new individuals.

The individual $X^t_i = \left[x^t_{i1}, x^t_{i2}, \cdots, x^t_{iD}\right]$, $i \in \{1, 2, \cdots, N\}$ is a D-dimensional vector. Its D dimensions are stochastically separated into τ sub-vectors $\left\{S^t_{i1}, S^t_{i2}, \cdots, S^t_{i\tau}\right\}$ and each sub-vector S^t_{ik}, $k \in \{1, 2, \cdots, \tau\}$ contains either at least Δ dimensions or at most $D - \Delta$ dimensions, Δ is the minimum dimensionality of each sub-vector. Thus, the possible value of τ is $2\sim D/\Delta$. The starting dimension of each S^t_{ik} is indexed in X^t_i as s_k. In this case, each sub-vector S^t_{ik} can be written as

$$S^t_{ik} = \begin{cases} \left[x^t_{i(s_k)}, x^t_{i(s_k+1)}, \cdots, x^t_{i(s_{k+1}-1)}\right], \\ \qquad\qquad k \in \{1, 2, \cdots, \tau - 1\} \\ \left[x^t_{i(s_k)}, x^t_{i(s_k+1)}, \cdots, x^t_{iD}\right], \quad k = \tau \end{cases} \quad (10)$$

where s_{k+1} denotes the starting dimension index of $S^t_{i(k+1)}$; particularly if $k = 1$, then $s_k = 1$.

Furthermore, the s_{k+1} can be computed as

$$\begin{cases} s_{k+1} = s_k + (\Delta - 1) + \lceil r[\mathcal{D} - s_k - (2\Delta - 2)]\rceil \\ \qquad k \in \{1, 2, \cdots, \tau - 1\} \end{cases} \tag{11}$$

where s_k is less than or equal to $\mathcal{D} - (2\Delta - 2)$, r is random numbers, uniformly distributed in the interval [0, 1], and $\lceil\ \rceil$ represents that it rounds a number to the next larger integer.

According to (10) and (11), each sub-vector can be acquired. For instance, considering $\mathcal{D} = 30$ and $\Delta = 3$, $s_2 = s_1 + 2 + \lceil r[26 - s_1]\rceil$; Due to $s_1 = 1$, $s_2 = 3 + \lceil 25r\rceil$; if $r = 0.9$, $\lceil 25r\rceil = 25$ and $s_2 = 28$; thus $\mathcal{X}_i^t = \left[x_{i1}^t, x_{i2}^t, \cdots, x_{i\mathcal{D}}^t\right]$ can be separated into two sub-vectors, $\mathcal{S}_{i1}^t = \left[x_{i1}^t, x_{i2}^t, \cdots, x_{i27}^t\right]$ and $\mathcal{S}_{i2}^t = \left[x_{i28}^t, x_{i29}^t, x_{i30}^t\right]$ with $\tau = 2$.

2) VECTOR GROUPING LEARNING PATTERNS

Two different vector grouping learning patterns A and B are developed to generate new individuals and provide the compromise between the global exploration and local exploitation as follows.

a: Pattern A

If $r_0 < p_{r0}$ where r_0 and p_{r0} denote a uniform random number within the interval [0, 1] and a predefined probability, respectively, we first stochastically choose a group a, $a \in \{1, 2, \cdots, \mathcal{M}\}$ from $\mathcal{M}$ groups with its corresponding center $\mathcal{C}_a^t = \left\{\mathcal{SC}_{a1}^t, \mathcal{SC}_{a2}^t, \cdots, \mathcal{SC}_{a\tau}^t\right\}$; we then stochastically select $\mathcal{X}_{a_1}^t = \left\{\mathcal{S}_{a_1 1}^t, \mathcal{S}_{a_1 2}^t, \cdots, \mathcal{S}_{a_1 \tau}^t\right\}$ from the group a; finally, for a new idea $\mathcal{X}_{new\_i}^t = \left\{\mathcal{S}_{new\_i1}^t, \mathcal{S}_{new\_i2}^t, \cdots, \mathcal{S}_{new\_i\tau}^t\right\}$, $i \in \{1, 2, \cdots, \mathcal{N}\}$, its each sub-vector $\mathcal{S}_{new\_ik}^t$, $k \in \{1, 2, \cdots, \tau\}$ can be created by performing a vector grouping learning between either $\mathcal{S}_{a_1 k}^t$ and $\mathcal{S}_{a_2 k}^t$, or $\mathcal{S}_{a_1 k}^t$ and $\mathcal{SC}_a^t$ as follows:

$$\mathcal{S}_{new\_ik}^t = \begin{cases} (1 - r_1)\,\mathcal{S}_{a_1 k}^t + r_1 \mathcal{S}_{\theta(k)k}^t, & r \le 0.5 \\ (1 - r_1)\,\mathcal{S}_{a_1 k}^t + r_1 \mathcal{SC}_{ak}^t, & r > 0.5 \end{cases} \tag{12}$$

Here, $\mathcal{S}_{a_1 k}^t$, $\mathcal{S}_{\theta(k)k}^t$, and $\mathcal{SC}_{ak}^t$ are the kth sub-vector of $\mathcal{X}_{a_1}^t$, $\mathcal{X}_{\theta(k)}^t$, and $\mathcal{C}_a^t$, respectively.

Particularly, for each $\mathcal{S}_{new\_ik}^t$, $k \in \{1, 2, \cdots, \tau\}$ in formula (12), we first stochastically select an individual $\mathcal{X}_{\theta(k)}^t = \left\{\mathcal{S}_{\theta(k)1}^t, \mathcal{S}_{\theta(k)2}^t, \cdots, \mathcal{S}_{\theta(k)\tau}^t\right\}$ from the group a; the corresponding sub-vector $\mathcal{S}_{\theta(k)k}^t$ of $\mathcal{X}_{\theta(k)}^t$ is employed to create the $\mathcal{S}_{new\_ik}^t$. In other words, for the $\mathcal{S}_{new\_ik}^t$ with different values of k, the corresponding $\mathcal{S}_{\theta(k)k}^t$ in formula (12) may be chosen from different individuals of the group a. For instance, without loss of generality, we assume that $\tau = 2$ and $k \in \{1, 2\}$. When $k = 1$, $\mathcal{S}_{new\_ik}^t = \mathcal{S}_{new\_i1}^t$ and $\mathcal{X}_{\theta(k)}^t = \mathcal{X}_{\theta(1)}^t = \left\{\mathcal{S}_{\theta(1)1}^t, \mathcal{S}_{\theta(1)2}^t\right\}$ are stochastically chosen from the group a. In this case, $\mathcal{S}_{\theta(k)k}^t = \mathcal{S}_{\theta(1)1}^t$. With $k = 2$, $\mathcal{S}_{new\_ik}^t = \mathcal{S}_{new\_i2}^t$ and $\mathcal{X}_{\theta(k)}^t = \mathcal{X}_{\theta(2)}^t = \left\{\mathcal{S}_{\theta(2)1}^t, \mathcal{S}_{\theta(2)2}^t\right\}$ are also stochastically chosen from the group a. Thus, $\mathcal{S}_{\theta(k)k}^t = \mathcal{S}_{\theta(2)2}^t$. Since both $\mathcal{X}_{\theta(1)}^t$ and $\mathcal{X}_{\theta(2)}^t$ are stochastically chosen from the group a, they may be the same individual. In this way, $\theta(1) = \theta(2)$; otherwise, $\theta(1) \ne \theta(2)$. Here, using the $\mathcal{S}_{\theta(k)k}^t$ in formula (12) aims to improve the diversity of the $\mathcal{S}_{new\_ik}^t$, contributing to enhance the diversity of $\mathcal{X}_{new\_i}^t$.

Additionally, from formula (12), if $r_1 = 0$, $\mathcal{S}_{new\_ik}^t = \mathcal{S}_{a_1 k}^t$; if $r_1 = 1$, $\mathcal{S}_{new\_ik}^t = \mathcal{S}_{\theta(k)k}^t$ or $\mathcal{SC}_{ak}^t$; if $0 < r_1 < 1$, $\mathcal{S}_{new\_ik}^t$ is equal to an arbitrary combination of either $\mathcal{S}_{a_1 k}^t$ and $\mathcal{S}_{\theta(k)k}^t$ or, $\mathcal{S}_{a_1 k}^t$ and $\mathcal{SC}_{ak}^t$, such as $\mathcal{S}_{new\_ik}^t = \mathcal{S}_{a_1 k}^t + 0.5\left(\mathcal{S}_{a_2 k}^t - \mathcal{S}_{a_1 k}^t\right)$ with $r_1 = 0.5$ and $r \le 0.5$, or $\mathcal{S}_{new\_ik}^t = \mathcal{S}_{a_1 k}^t + 0.5\left(\mathcal{SC}_{ak}^t - \mathcal{S}_{a_1 k}^t\right)$ with $r_1 = 0.5$ and $r > 0.5$. Since various potential $\mathcal{S}_{new\_ik}^t$ can be acquired via formula (12), more promising $\mathcal{X}_{new\_i}^t = \left\{\mathcal{S}_{new\_i1}^t, \mathcal{S}_{new\_i2}^t, \cdots, \mathcal{S}_{new\_i\tau}^t\right\}$ will also be obtained during the whole the iteration process.

In IC-VGL, *pattern A* emphasizes the vector grouping learning between different individuals from one group; that is to say, the information exchanges between different sub-vectors also mostly focus on local regions (one group). Therefore, *pattern A* plays a chief role in the local exploitation.

b: Pattern B

If $r_0 \ge p_{r0}$, two different groups a and b, $a, b \in \{1, 2, \cdots, \mathcal{M}\}$ are first stochastically chosen from $\mathcal{M}$ groups with their corresponding centers $\mathcal{C}_a^t = \left\{\mathcal{SC}_{a1}^t, \mathcal{SC}_{a2}^t, \cdots, \mathcal{SC}_{a\tau}^t\right\}$ and $\mathcal{C}_b^t = \left\{\mathcal{SC}_{b1}^t, \mathcal{SC}_{b2}^t, \cdots, \mathcal{SC}_{b\tau}^t\right\}$, respectively; then, two different individuals $\mathcal{X}_a^t = \left\{\mathcal{S}_{a1}^t, \mathcal{S}_{a2}^t, \cdots, \mathcal{S}_{a\tau}^t\right\}$ and $\mathcal{X}_b^t = \left\{\mathcal{S}_{b1}^t, \mathcal{S}_{b2}^t, \cdots, \mathcal{S}_{b\tau}^t\right\}$ are stochastically selected from the groups a and b, respectively; finally, for a new idea $\mathcal{X}_{new\_i}^t = \left\{\mathcal{S}_{new\_i1}^t, \mathcal{S}_{new\_i2}^t, \cdots, \mathcal{S}_{new\_i\tau}^t\right\}$, $i \in \{1, 2, \cdots, \mathcal{N}\}$, its each sub-vector $\mathcal{S}_{new\_ik}^t$, $k \in \{1, 2, \cdots, \tau\}$ can be created by consistently executing the following vector grouping learning:

$$\mathcal{S}_{new\_ik}^t = \begin{cases} \mathcal{SC}_{ak}^t + r_1\left(\mathcal{S}_{ak}^t - \mathcal{SC}_{ak}^t\right) + r_2\left(\mathcal{S}_{bk}^t - \mathcal{SC}_{ak}^t\right), \\ \qquad 0 < r \le 0.25 \\ \mathcal{SC}_{bk}^t + r_1\left(\mathcal{S}_{ak}^t - \mathcal{SC}_{bk}^t\right) + r_2\left(\mathcal{S}_{bk}^t - \mathcal{SC}_{bk}^t\right), \\ \qquad 0.25 < r \le 0.5 \\ \mathcal{SC}_{\mathrm{M}k}^t + r_1\left(\mathcal{S}_{ak}^t - \mathcal{SC}_{\mathrm{M}k}^t\right) + r_2\left(\mathcal{S}_{bk}^t - \mathcal{SC}_{\mathrm{M}k}^t\right), \\ \qquad 0.5 < r \le 0.75 \\ \mathcal{S}_{\theta(k)k}^t + r_1\left(\mathcal{S}_{ak}^t - \mathcal{S}_{\theta(k)k}^t\right) + r_2\left(\mathcal{S}_{bk}^t - \mathcal{S}_{\theta(k)k}^t\right), \\ \qquad 0.75 < r \le 1 \end{cases} \tag{13}$$

where $\mathcal{S}_{a}^{t}$ and $\mathcal{S}_{bk}^{t}$ are the kth sub-vector of $\mathcal{X}_{a}^{t}$ and $\mathcal{X}_{b}^{t}$ respectively; $\mathcal{SC}_{ak}^{t}$ and $\mathcal{SC}_{bk}^{t}$ are the kth sub-vector of $\mathcal{C}_{a}^{t}$ and $\mathcal{C}_{b}^{t}$ respectively;$\mathcal{SC}_{Mk}^{t}$ is the kth sub-vector of $\mathcal{C}_{M}^{t} = \left\{\mathcal{SC}_{M1}^{t}, \mathcal{SC}_{M2}^{t}, \cdots, \mathcal{SC}_{M\tau}^{t}\right\}$ represents the mean value of all the $\mathcal{M}$ group centers $\left\{\mathcal{C}_{1}^{t}, \mathcal{C}_{2}^{t}, \cdots, \mathcal{C}_{\mathcal{M}}^{t}\right\}$; $\mathcal{S}_{\theta(k)k}^{t}$ is the kth sub-vector of $\mathcal{X}_{\theta(k)k}^{t} = \left\{\mathcal{S}_{\theta(k)1}^{t}, \mathcal{S}_{\theta(k)2}^{t}, \cdots, \mathcal{S}_{\theta(k)\tau}^{t}\right\}$ that is stochastically selected from the entire swarm for each $\mathcal{S}_{new\_ik}^{t}$.

Note that the purpose of using $\mathcal{SC}_{Mk}^{t}$ and $\mathcal{S}_{\theta(k)k}^{t}$ in formula (13) is to augment the diversity for $\mathcal{X}_{new\_i}^{t} = \left\{\mathcal{S}_{new\_i1}^{t}, \mathcal{S}_{new\_i2}^{t}, \cdots, \mathcal{S}_{new\_i\tau}^{t}\right\}, i \in \{1, 2, \cdots, \mathcal{N}\}$ by further enriching the diversity of the $\mathcal{S}_{new\_ik}^{t}$, $k \in \{1, 2, \cdots, \tau\}$. Furthermore, the $\mathcal{S}_{\theta(k)k}^{t}$ of formula (13) is almost exactly the identical with that of formula (12), the only difference being that the former is selected from the entire swarm, whereas the latter from a group randomly selected.

Moreover, *pattern B* focuses on the vector grouping learning between different individuals from two different groups to the entire swarm. Therefore, *pattern B* plays a crucial role in the global exploration. From this reason, formula (13) uses the three items such as $\mathcal{SC}_{ak}^{t} + r_1\left(\mathcal{S}_{ak}^{t} - \mathcal{SC}_{ak}^{t}\right) + r_2\left(\mathcal{S}_{bk}^{t} - \mathcal{SC}_{ak}^{t}\right)$ instead of two items such as $(1 - r_1)\,\mathcal{S}_{a1k}^{t} + r_1\mathcal{S}_{\theta(k)k}^{t}$ in formula (12) to enhance the possibility of various information exchanges between sub-vectors. Note that if $r_1 \neq 0$ and $r_2 \neq 0$, the $\mathcal{S}_{new\_ik}^{t}$ can reflect the information exchanges between three sub-vectors like $\mathcal{SC}_{ak}^{t} + r_1\left(\mathcal{S}_{ak}^{t} - \mathcal{SC}_{ak}^{t}\right) + r_2\left(\mathcal{S}_{bk}^{t} - \mathcal{SC}_{ak}^{t}\right)$. However, if $r_1 = 0$ and $r_2 \neq 0$, or $r_1 \neq 0$ and $r_2 = 0$, the $\mathcal{S}_{new\_ik}^{t}$ can embody the information exchanges between two sub-vectors. For example, if $r_1 = 0$ and $r_2 \neq 0$,

$$\mathcal{S}_{new\_ik}^{t} = \begin{cases} \mathcal{SC}_{ak}^{t} + r_2\left(\mathcal{S}_{bk}^{t} - \mathcal{SC}_{ak}^{t}\right), & 0 < r \leq 0.25 \\ \mathcal{SC}_{bk}^{t} + r_2\left(\mathcal{S}_{bk}^{t} - \mathcal{SC}_{bk}^{t}\right), & 0.25 < r \leq 0.5 \\ \mathcal{SC}_{Mk}^{t} + r_2\left(\mathcal{S}_{bk}^{t} - \mathcal{SC}_{Mk}^{t}\right), & 0.5 < r \leq 0.75 \\ \mathcal{S}_{rk}^{t} + r_2\left(\mathcal{S}_{bk}^{t} - \mathcal{S}_{\theta(k)k}^{t}\right), & 0.75 < r \leq 1 \end{cases}$$

In addition, *pattern B* should have more opportunities to operate in the early iteration to focus more on the global exploration and discover more promising solution regions; in contrast, *pattern A*should be more likely to execute in the later iteration to emphasize more on local exploitation and accelerate the convergence speed. Thus, p_{r0} is configured as a dynamic adaptive form as follows:

$$p_{r0} = p_l + p_h \times t/\mathcal{T} \quad (14)$$

where p_l and p_h represent two constants, defined as the lower and higher boundaries of p_{r0}, respectively; t and $\mathcal{T}$ denote the current iteration number and maximum iteration number, respectively.

During the early iterations, with the relatively small value of p_{r0}, the condition $r_0 \geq p_{r0}$ is more easily satisfied than $r_0 < p_{r0}$, so IC-VGL achieves more opportunities to conduct *Pattern B*; on the other hand, during the latter iteration, with the value of p_{r0} consistently increasing, the condition $r_0 < p_{r0}$ is more easily true than $r_0 \geq p_{r0}$ so that IC-VGL acquires more opportunities to execute *Pattern A*. Therefore, formula (14) can play the important role in compromising the global exploration and local exploitation capability during the entire iteration process.

Ultimately, each new idea $\mathcal{X}_{new\_i}^{t} = \left\{\mathcal{S}_{new\_i1}^{t}, \mathcal{S}_{new\_i2}^{t}, \cdots, \mathcal{S}_{new\_i\Gamma}^{t}\right\}$, $i \in \{1, 2, \cdots, \mathcal{N}\}$ can be obtained via formulas (12) and (13).

C. HYBRID INDIVIDUALS' UPDATE

As for most of the existing BSO algorithms, their IU schemes adopt either the step size of the logarithmic sigmoid function with a Gaussian random number such as the IU scheme of the original BSO, or the differential step size between individuals like the IU scheme of CLBSO. The former emphasizes more on global exploration owing to using the Gaussian random number. Although the latter can provide the effective local exploitation, it cannot supply sufficient global exploration capability when tackling the complicated global problems.

To further compromise the global exploration and local exploitation, the H-IU scheme is developed by hybridizing the individuals' update schemes of the original BSO and CLBSO as follows.

$$\mathcal{X}_{\text{temp}\_i}^{t} = \begin{cases} \mathcal{X}_{new\_i}^{t} + \mathbb{N}(\mu, \sigma)\,\Phi(t), & 1 \leq i \leq \lceil \lambda \mathcal{N} \rceil \\ \mathcal{X}_{new\_i}^{t} + \mathcal{R} \cdot \left(\mathcal{X}_{f}^{t} - \mathcal{X}_{g}^{t}\right), & \lceil \lambda \mathcal{N} \rceil < i \leq \mathcal{N}. \end{cases} \quad (15)$$

Here, the first row of formula (15) is from the individuals' update scheme of the original BSO; $\mathbb{N}(\mu, \sigma)$ and $\Phi(t)$ are the Gaussian random vector and step size vector, respectively, their detailed definitions given by formulas (3) and (4). The second row of formula (15) is from the individuals' update scheme of CLBSO; $\mathcal{X}_{f}^{t}$ and $\mathcal{X}_{g}^{t}$ are two individuals stochastically chosen from the entire swarm; $\mathcal{R}$ is defined as random vector $[r_1, r_2, \cdots, r_{\mathcal{D}}]$; for each $r_j, j \in \{1, 2, \cdots, \mathcal{D}\}$, it is a uniformly distributed random number within interval [0, 1]; $\odot$ represents Hadamard product; t denotes the current iteration number; λ is a scale factor; $\mathcal{N}$ is the individual number of the entire swarm; $\lceil\ \rceil$ represents that it rounds a number to the next larger integer.

On the one hand, due to the randomness of $\mathbb{N}(\mu, \sigma)\,\Phi(t)$, $\mathcal{X}_{new\_i}^{t}$ in the first row of formula (15) has more opportunities to perform the global exploration than that in the second row; on the other hand, owing to the differential step size $\left(\mathcal{X}_{f}^{t} - \mathcal{X}_{g}^{t}\right)$, $\mathcal{X}_{new\_i}^{t}$ in the second row has more opportunities to execute the local exploitation towards $\mathcal{X}^t_f$ and $\mathcal{X}^t_g$ than that in the first row, which can allow $\mathcal{X}^t_{new\_i}$ in the second row of formula (15) to quickly converge towards $\mathcal{X}^t_f$ and $\mathcal{X}^t_g$. Therefore, the first and second row of formula (15) are employed to focus on the global exploration and local exploitation, respectively.

Further, as a scale factor, λ is applied to manage the number of individuals using the first and second row of formula (15). Note that $\lceil \lambda \mathcal{N} \rceil$ indicates the number of individuals that are updated using the individuals' update scheme of the original BSO (the first row of formula (15)); clearly, $\mathcal{N} - \lceil \lambda \mathcal{N} \rceil$ denotes the number of individuals updated using CLBSO (the second row of formula (15)). Therefore, by regulating the value of λ, the H-IU scheme can further supply the effective balance between the global exploration and local exploitation.

Eventually, the selection operator is employed to find the competitive individuals in the entire swarm. Without loss of generality, let's consider the minimum fitness value for function $\mathcal{F}$. Thus, for $i \in \{1, 2, \cdots, \mathcal{N}\}$, the ith individual is chosen as

$$\mathcal{X}^{t+1}_i = \begin{cases} \mathcal{X}^t_{\text{temp}\_i}, & \mathcal{F}\left[\mathcal{X}^t_{\text{temp}\_i}\right] < \mathcal{F}\left[\mathcal{X}^t_i\right] \\ \mathcal{X}^t_i, & \mathcal{F}\left[\mathcal{X}^t_{\text{temp}\_i}\right] \geq \mathcal{F}\left[\mathcal{X}^t_i\right]. \end{cases} \tag{16}$$

Here, $\mathcal{F}\left[\mathcal{X}^t_{\text{temp}\_i}\right]$ and $\mathcal{F}\left[\mathcal{X}^t_i\right]$ are the fitness function values of $\mathcal{X}^t_{\text{temp}\_i}$ and $\mathcal{X}^t_i$, respectively.

After the VGLBSO has completed all individuals' update in each iteration, the stopping criterion is confirmed. If such a criterion is true, the operation of VGLBSO is to be terminated. Otherwise, the iteration search is to continue.

D. PROCEDURE OF VGLBSO

The pseudo code of VGLBSO is provided as follows.

Fig. 2 presents the framework of VGLBSO, and its implementation procedure is illustrated as follows.

Step 1): In VGLBSO, all $\mathcal{N}$ individuals are stochastically initialized; their corresponding fitness values are evaluated. The corresponding pseudo code is described in the lines 2-4 of Algorithm 1, where $\mathcal{G}$ is the individual corresponding to the minimum fitness value; *min* $\left\{\mathcal{F}\left[\mathcal{X}^t_1\right], \mathcal{F}\left[\mathcal{X}^t_2\right], \cdots, \mathcal{F}\left[\mathcal{X}^t_{\mathcal{N}}\right]\right\}$ is the minimum fitness value of all individuals; *arg* is the inverse function of $\mathcal{F}$.

Step 2): For every iteration, the RG scheme is executed. The corresponding pseudocode is written in the lines 6-12 of Algorithm 1.

Step 3): For every iteration, the stochastic vector grouping mechanism is executed for each individual and the corresponding pseudo code is provided in the lines 14-27 of Algorithm 1; $\mathcal{N}$ new individuals are created by two different vector grouping learning patterns and the corresponding pseudo code is given in the lines 29-62 of Algorithm 1.

Algorithm 1 VGLBSO

1: /***Initialization***/
2: All $\mathcal{N}$ individuals $\left\{\mathcal{X}^t_1, \mathcal{X}^t_2, \cdots, \mathcal{X}^t_{\mathcal{N}}\right\}$ are stochastically initialized;
3: Their fitness values are evaluated as $\left\{\mathcal{F}\left[\mathcal{X}^t_1\right], \mathcal{F}\left[\mathcal{X}^t_2\right], \cdots, \mathcal{F}\left[\mathcal{X}^t_{\mathcal{N}}\right]\right\}$;
4: $G = \arg\left[\min\left\{\mathcal{F}\left[\mathcal{X}^t_1\right], \mathcal{F}\left[\mathcal{X}^t_2\right], \cdots, \mathcal{F}\left[\mathcal{X}^t_{\mathcal{N}}\right]\right\}\right]$;
5: **while** *(stopping condition is not true)* **do**
6: /***RG scheme***/
7: Stochastically sort $\left\{\mathcal{X}^t_1, \mathcal{X}^t_2, \cdots, \mathcal{X}^t_{\mathcal{N}}\right\}$ into $\left\{\mathcal{X}^t_1, \mathcal{X}^t_2, \cdots, \mathcal{X}^t_{\mathcal{N}}\right\}$;
8: $\rho = \mathcal{N}/\mathcal{M}$;
9: **for** $m = 1$ **to** $\mathcal{M}$
10: $\left\{\mathbb{X}^t_{(m-1)\rho+1}, \mathbb{X}^t_{(m-1)\rho+2}, \cdots, \mathbb{X}^t_{(m-1)\rho+\rho}\right\}$ are allocated into the mth group;
11: $\mathcal{C}^t_m = \arg\left[\min\left\{\mathcal{F}\left[\mathbb{X}^t_{(m-1)\rho+1}\right], \mathcal{F}\left[\mathbb{X}^t_{(m-1)\rho+2}\right], \cdots, \mathcal{F}\left[\mathbb{X}^t_{(m-1)\rho+\rho}\right]\right\}\right]$;
12: end for
13: /***IC-VGL scheme***/
14: for $i = 1$ **to** $\mathcal{N}$/* **Stochastic Vector grouping Mechanism** */
15: $s_1 = 1$;
16: **for** $k = 1$ **to** $\lfloor \mathcal{D}/\Delta \rfloor$
17: **if** $s_k < \mathcal{D} - (2\Delta - 2)$
18: $\mathcal{S}^t_{ik} = \left[x^t_{i(s_k)}, x^t_{i(s_k+1)}, \cdots, x^t_{i(s_{k+1}-1)}\right]$;
19: $s_{k+1} = s_k + (\Delta - 1) + \lceil r[\mathcal{D} - s_k - (2\Delta - 2)]\rceil$;
20: **else**
21: $\mathcal{S}^t_{ik} = \left[x^t_{i(s_k)}, x^t_{i(s_k+1)}, \cdots, x^t_{i\mathcal{D}}\right]$;
22: $\Gamma = k$;
23: **break;**
25: **end if**
26: **end for**
27: end for
28: for $i = 1$ **to** $\mathcal{N}$
29: **if** $r_0 < p_{r0}$/***Vector grouping learning pattern A** */
30: Stochastically choose group a, $\left\{\mathcal{X}^t_{(a-1)\rho+1}, \mathcal{X}^t_{(a-1)\rho+2}, \cdots, \mathcal{X}^t_{(a-1)\rho+\rho}\right\}$;
31: Stochastically choose an individual $\mathcal{X}^t_{a_1} = \left\{\mathcal{S}^t_{a_1 1}, \mathcal{S}^t_{a_1 2}, \cdots, \mathcal{S}^t_{a_1 \tau}\right\}$ from group a, $\left\{\mathcal{X}^t_{(a-1)\rho+1}, \mathcal{X}^t_{(a-1)\rho+2}, \cdots, \mathcal{X}^t_{(a-1)\rho+\rho}\right\}$;
32: $\mathcal{C}^t_a = \left\{\mathcal{SC}^t_{a1}, \mathcal{SC}^t_{a2}, \cdots, \mathcal{SC}^t_{a\tau}\right\}$;
33: **for** k=1**to** τ
34: **if** $r \leq 0.5$
35: Stochastically choose an individual $\mathcal{X}^t_{\theta(k)} = \left\{\mathcal{S}^t_{\theta(k)1}, \mathcal{S}^t_{\theta(k)2}, \cdots, \mathcal{S}^t_{\theta(k)\tau}\right\}$ from group a;
36: $\mathcal{S}^t_{new\_ik} = (1 - r_1)\,\mathcal{S}^t_{a_1 k} + r_1 \mathcal{S}^t_{\theta(k)k}$;

Algorithm 1 *(Continued.)* VGLBSO

37: **else**
38: $\mathcal{S}^t_{new\_ik} = (1 - r_1)\,\mathcal{S}^t_{a_1k} + r_1 \mathcal{SC}^t_{a\tau}$;
39: **end if**
40: **end for**
41: **else** /* **Vector grouping learning pattern B** */
42: Stochastically choose group a, $\left\{\mathcal{X}^t_{(a-1)\rho+1}, \mathcal{X}^t_{(a-1)\rho+2}, \cdots, \mathcal{X}^t_{(a-1)\rho+\rho}\right\}$;
43: Stochastically choose group b, $\left\{\mathcal{X}^t_{(b-1)\rho+1}, \mathcal{X}^t_{(b-1)\rho+2}, \cdots, \mathcal{X}^t_{(b-1)\rho+\rho}\right\}$;
44: $a \neq b$;
45: Stochastically choose two individuals $\mathcal{X}^t_a = \left\{\mathcal{S}^t_{a1}, \mathcal{S}^t_{a2}, \cdots, \mathcal{S}^t_{a\tau}\right\}$ and $\mathcal{X}^t_b = \left\{\mathcal{S}^t_{b1}, \mathcal{S}^t_{b2}, \cdots, \mathcal{S}^t_{b\tau}\right\}$ from groups a and b;
46: **for** k=1**to** Γ
47: **if** $r \leq 0.25$
48: $\mathcal{S}^t_{new\_ik} = \mathcal{SC}^t_{ak} + r_1\left(\mathcal{S}^t_{ak} - \mathcal{SC}^t_{ak}\right) + r_2\left(\mathcal{S}^t_{bk} - \mathcal{SC}^t_{ak}\right)$;
49: **end if**
50: **if** $0.25 < r \leq 0.5$
51: $\mathcal{S}^t_{new\_ik} = \mathcal{SC}^t_{bk} + r_1\left(\mathcal{S}^t_{ak} - \mathcal{SC}^t_{bk}\right) + r_2\left(\mathcal{S}^t_{bk} - \mathcal{SC}^t_{bk}\right)$;
52: **end if**
53: **if** $0.5 < r \leq 0.75$
54: Calculate the mean value of all $\mathcal{M}$ group centers $\left\{\mathcal{C}^t_1, \mathcal{C}^t_2, \cdots, \mathcal{C}^t_{\mathcal{M}}\right\}$ as $\mathcal{C}^t_{\mathrm{M}} = \left\{\mathcal{SC}^t_{\mathrm{M}1}, \mathcal{SC}^t_{\mathrm{M}2}, \cdots, \mathcal{SC}^t_{\mathrm{M}\tau}\right\}$;
55: $\mathcal{S}^t_{new\_ik} = \mathcal{SC}^t_{\mathrm{M}k} + r_1\left(\mathcal{S}^t_{ak} - \mathcal{SC}^t_{\mathrm{M}k}\right) + r_2\left(\mathcal{S}^t_{bk} - \mathcal{SC}^t_{\mathrm{M}k}\right)$;
56: **end if**
57: **if** $r > 0.75$
58: Stochastically choose an individual $\mathcal{X}^t_{\theta(k)} = \left\{\mathcal{S}^t_{\theta(k)1}, \mathcal{S}^t_{\theta(k)2}, \cdots, \mathcal{S}^t_{\theta(k)\tau}\right\}$ from the entire swarm;
59: $\mathcal{S}^t_{new\_ik} = \mathcal{S}^t_{\theta(k)k} + r_1\left(\mathcal{S}^t_{ak} - \mathcal{S}^t_{\theta(k)k}\right) + r_2\left(\mathcal{S}^t_{bk} - \mathcal{S}^t_{\theta(k)k}\right)$;
60: **end if**
61: **end for**
62: **end if**
63: /* **H-IU scheme** */
64: **if** $1 \leq i \leq \lceil \lambda \mathcal{N} \rceil$
65: $\mathcal{X}^t_{\mathrm{temp}\_i} = \mathcal{X}^t_{new\_i} + \mathcal{N}(\mu, \sigma)\,\Phi(t)$;
66: **else**
67: Stochastically choose two individuals $\mathcal{X}^t_f$ and $\mathcal{X}^t_g$ from the entire swarm;

Algorithm 1 *(Continued.)* VGLBSO

68: $\mathcal{X}^t_{\mathrm{temp}\_i} = \mathcal{X}^t_{new\_i} + \mathcal{R} \odot \left(\mathcal{X}^t_f - \mathcal{X}^t_g\right)$;
69: **end if**
70: **if** $\mathcal{F}\left[\mathcal{X}^t_{\mathrm{temp}\_i}\right] < \mathcal{F}\left[\mathcal{X}^t_i\right]$
71: $\mathcal{X}^{t+1}_i = \mathcal{X}^t_{\mathrm{temp}\_i}$;
72: **else**
73: $\mathcal{X}^{t+1}_i = \mathcal{X}^t_i$;
74: **end if**
75: **end for**
76: $G^{t+1} = \arg\left[\min\left\{f\left[X^{t+1}_1\right], f\left[X^{t+1}_2\right], \cdots, f\left[X^{t+1}_N\right]\right\}\right]$;
77: end while

Step 4): For every iteration, all $\mathcal{N}$ individuals are updated according to H-IU scheme, the corresponding pseudo code shown in the lines 64-69 of Algorithm 1; the promising individuals are selected for the next iteration, the corresponding pseudo code displayed in the lines 70-74.

Step 5): Steps 2)∼4) are executed repeatedly in the different iterations until the specific stopping condition is met.

Particularly, with different $i \in \{1, 2, \cdots, \mathcal{N}\}$ and $j \in \{1, 2, \cdots, \mathcal{D}\}$, the x^t_{ij} of $\mathcal{X}^t_i$ is limited in $\min\left\{u_j, \max\left\{l_j, x^t_{ij}\right\}\right\}$ where l_j and u_j represent the lower and upper bound of the jth dimensional search space, respectively.

In general, by combining the above RG, IC-VGL, and H-IU scheme, VGLBSO can provide the reasonable diversity of population, decrease the computational burden, and improve the global exploration and local exploitation capability.

E. COMPUTATIONAL COMPLEXITY OF VGLBSO

Let's assume that $\mathcal{N}$, $\mathcal{D}$, and $\mathcal{T}$ represent the population size, the dimensionality of individual, and the maximum iteration number, respectively. The computational complexity of the original BSO consists of four main components: the individuals' initialization ($\mathbb{T}_{ii}$), individuals' grouping ($\mathbb{T}_{ig}$), individuals' creation ($\mathbb{T}_{ic}$), and individuals' update ($\mathbb{T}_{iu}$). Thus, the total computational cost of the original BSO can be written the following formula:

$$\begin{aligned}\mathbb{T}_{\mathrm{BSO}} &= \mathbb{T}_{ii} + \mathbb{T}_{ig} + \mathbb{T}_{ic} + \mathbb{T}_{iu} \\ &= \mathcal{ND} + \mathcal{TNDYM} + \mathcal{TND} + \mathcal{TND}\end{aligned} \tag{17}$$

where $\mathcal{ND}$, $\mathcal{TNDYM}$, $\mathcal{TND}$, and $\mathcal{TND}$ represent $\mathbb{T}_{ii}$, $\mathbb{T}_{ig}$, $\mathbb{T}_{ic}$, and $\mathbb{T}_{iu}$, respectively; $\mathcal{M}$ and $\mathcal{Y}$ denote the total number of the groups and the maximum iteration number of the K-means grouping scheme, respectively. Accordingly, the original BSO's total computational complexity is given as $o(\mathcal{TNDYM})$.

Similarly, the computational complexity of VGLBSO also covers four main components: $\mathbb{T}_{ii}$, $\mathbb{T}_{ig}$, $\mathbb{T}_{ic}$, and $\mathbb{T}_{iu}$. With the same as $\mathbb{T}_{ii}$ and $\mathbb{T}_{iu}$ of the original BSO, those of the VGLBSO are also measured as $\mathcal{ND}$ and $\mathcal{TND}$. Furthermore, the $\mathbb{T}_{ig}$ of

TABLE 1. CEC2013 benchmark suite consists of 28 extremely complex shifted or shifted rotated functions.

Types	No.	Benchmark Functions	$F_{min}[\mathcal{X}^*]$
Unimodal	F_1	Shifted Sphere	-1400
	F_2	Shifted Rotated High Conditioned Elliptic	-1300
	F_3	Shifted Rotated Bent Cigar	-1200
	F_4	Shifted Rotated Discus	-1100
	F_5	Shifted Different Powers	-1000
Multimodal	F_6	Shifted Rotated Rosenbrock	-900
	F_7	Shifted Rotated Schaffers F_7	-800
	F_8	Shifted Rotated Ackley	-700
	F_9	Shifted Rotated Weierstrass	-600
	F_{10}	Shifted Rotated Griewank	-500
	F_{11}	Shifted Rastrigin	-400
	F_{12}	Shifted Rotated Rastrigin	-300
	F_{13}	Shifted Non-Continuous Rotated Rastrigin	-200
	F_{14}	Shifted Schwefel	-100
	F_{15}	Shifted Rotated Schwefel	100
	F_{16}	Shifted Rotated Katsuura	200
	F_{17}	Shifted Lunacek Bi_Rastrigin	300
	F_{18}	Shifted Rotated Lunacek Bi_Rastrigin	400
	F_{19}	Shifted Rotated Expanded Groewank's plus Rosenbrock	500
	F_{20}	Shifted Rotated Expanded Scaffer's F_6	600
Composition	F_{21}	Composition Instance 1 (n=5, Rotated)	700
	F_{22}	Composition Instance 2 (n=3, Unrotated)	800
	F_{23}	Composition Instance 3 (n=3, Rotated)	900
	F_{24}	Composition Instance 4 (n=3, Rotated)	1000
	F_{25}	Composition Instance 5 (n=3, Rotated	1100
	F_{26}	Composition Instance 6 (n=5, Rotated)	1200
	F_{27}	Composition Instance 7 (n=5, Rotated)	1300
	F_{28}	Composition Instance 8 (n=5, Rotated)	1400

VGLBSO is measured as $\mathcal{N}\mathcal{M}$ due to using the RG scheme for grouping the individuals. In addition, the individuals' creation ($\mathbb{T}_{ic}$) of VGLBSO contains two sub-components: the stochastic vector grouping ($\mathbb{T}_{ic\_01}$) and two vector grouping learning patterns ($\mathbb{T}_{ic\_02}$); $\mathbb{T}_{ic\_01}$ are measured as $\mathcal{T}\mathcal{N}\lfloor\mathcal{D}/\Delta\rfloor$ and $\lfloor\mathcal{D}/\Delta\rfloor$ is the worst situation for the vector grouping; $\mathbb{T}_{ic\_02}$ is measured as $\mathcal{T}\mathcal{N}\mathcal{D}$. In this case, VGLBSO's total computational cost is expressed as

$$\begin{aligned}\mathbb{T}_{\text{BSO}} &= \mathbb{T}_{ii} + \mathbb{T}_{ig} + \mathbb{T}_{ic} + \mathbb{T}_{iu} \\ &= \mathcal{N}\mathcal{D} + \mathcal{T}\mathcal{M} + \mathcal{T}\mathcal{N}\left(\lfloor\mathcal{D}/\Delta\rfloor + \mathcal{D}\right) + \mathcal{T}\mathcal{N}\mathcal{D}. \end{aligned} \quad (18)$$

Therefore, VGLBSO's total computational complexity is characterized as $o\left(\mathcal{T}\mathcal{N}\mathcal{D}\right)$. Clearly, VGLBSO has lower computational complexity than the original BSO. The fundamental reason is that VGLBSO utilizes the RG scheme.

IV. EXPERIMENTS AND DISCUSSIONS

A. EXPERIMENTAL ESTABLISHMENT

We test the property of VGL-BSO through a set of prevalent test functions, named CEC2013 benchmark suit [26], displayed in Table 1, where the 28 functions cover shifted or shifted rotated functions used for real parameter optimization in extraordinary complex conditions. Notice that F_1-F_5, F_6-F_{20}, and F_{21}-F_{28} are affiliated with the unimodal, multimodal and composition functions. For F_1-F_{28}, both their initialization and search ranges vary within the interval $[-100, 100]^{\mathcal{D}}$; $\mathcal{D}$ and $F_{min}[\mathcal{X}*]$ are defined as the dimensional number and the minimum value for each of F_1-F_{28}, respectively.

Firstly, VGLBSO is compared with 12 peer BSO algorithms: the original BSO [13], the modified BSO (MBSO) [29], CLBSO [15], predator prey BSO (PPBSO) [31], SBSO [24], BSO with differential evolution (BSODE) [30], random grouping BSO (RGBSO) [39], ADMBSO [26], BSO with dynamic grouping strategy (BSODCS) [37], BSO with chaotic operation (BSOCO) [33], GBSO [27], and MIIBSO [28] to validate its effectiveness and efficiency. Particularly, the 12 BSO algorithms have exhibited the positive global exploration and local exploitation capability in the literature. Additionally, the logistic map for chaotic operation in BSOCO is performed 200 times, after every five iterations of BSOCO are done.

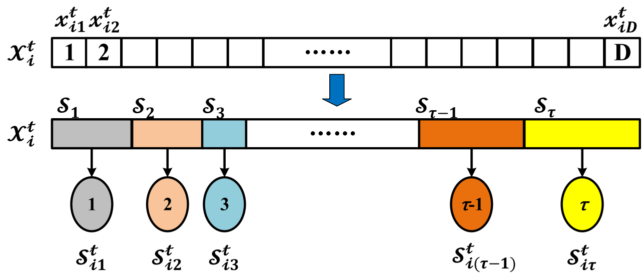


FIGURE 1. Schematic diagram of stochastic vector grouping.

Secondly, VGLBSO is further compared with the nine representative swarm intelligence algorithms: ABC [4], CMA-ES [11], DE [10], self-adaptive DE (SADE) [41], PSO [3], comprehensive learning PSO (CLPSO) [42], continuous ant colony optimization (ACO_R) [43], genetic learning PSO (GLPSO) [46], and multi-population ensemble DE (MPEDE) [47] to further verify its effectiveness. Specially, DE and PSO utilize the DE/rand/1/bin and global mode, respectively.

To obtain equitable comparisons among the aforesaid algorithms, they are evaluated independently 30 times on each of the 28 CEC2013 functions. The maximum number of fitness evaluations (MAXFES) is assigned to $10000\mathcal{D}$. For the aforesaid algorithms excluding CMA-ES, the population size is allocated to $\mathcal{N} = 50$; however, the population size of CMA-ES is set to $4 + 3\ln\lfloor\mathcal{N}\rfloor = 4 + 3\ln\lfloor 50\rfloor$ according to its definition in [11]. Considering the problem dimension of $\mathcal{D} = 50$ for the previous algorithms excluding CMA-ES, their maximum number of iterations is designated as $\mathcal{T} = 10000$ in view of $\mathcal{T} = \text{MAXFES}/\mathcal{N}$. For CMA-ES, however, its maximum number of iterations is equal to $\mathcal{T} = \text{MAXFES}/(4 + 3\ln\lfloor\mathcal{N}\rfloor)$.

Table 2 lists the parameter configurations for those algorithms, following the corresponding references. Moreover, the previous algorithms are all coded and executed in MATLAB R2017a based on a PC with Intel Core (TM) CPU i7-4790U CPU @ 3.60 GHz with 8 GB RAM.

B. PERFORMANCE INDEX

The mean value of error (Mean) and standard deviation value are employed to measure and rank the above compared algorithms' performance. Such two indicators are given in the following formulas (19) and (20), respectively.

$$\text{Mean} = \sum\nolimits_{\kappa=1}^{K}\left[\mathcal{F}\left[\mathcal{X}\right] - \mathcal{F}_{min}\left[\mathcal{X}^*\right]\right]/K \quad (19)$$

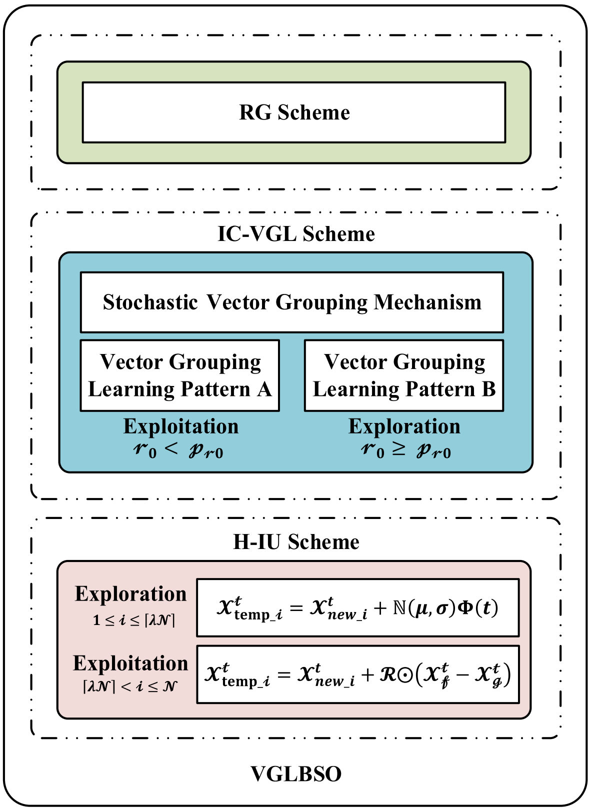


FIGURE 2. Framework of VGLBSO.

TABLE 2. Parameter settings for all comparative algorithms.

Algorithm	Parameters Settings	Reference
BSO	p_{r00}=0.2, p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, η=25, $\mathcal{M}$=5	[13]
MBSO	p_{r00}=0.2, p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, η=25, $\mathcal{M}$=5, p_r= 0.005	[29]
CLBSO	p_{r00}=0.2, p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, $\mathcal{M}$=5	[16]
PPBSO	p_{r00}=0.2, p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, $\mathcal{M}$=5, p_{prey}=0.1, $W_{predator}$=0.05	[31]
SBSO	p_{r00}=0.2, p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, η=25, $\mathcal{M}$=5	[24]
BSODE	p_{r00}=0.2, p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, $\mathcal{M}$=5, CR=0.5, F=0.5	[30]
RGBSO	p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, $\mathcal{M}$=5	[39]
ADMBSO	p_{cen}=0.7, p_{ind}=0.2, p_{rnd}=0.1, p_{cens}=0.7, p_{low}=0.2, p_{high}=0.2, $\mathcal{M}$=5	[26]
BSODCS	p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, $\mathcal{M}$=5, $p_{dynamic}$=0.5	[37]
BSOCO	p_{r00}=0.2, p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, η=25, $\mathcal{M}$=5, r=4	[33]
GBSO	p_{r0}=0.8, p_{r01}=0.4, p_{r02}=0.5, η=25, $\mathcal{M}$=5, ∇=10, F=0.5, C_{min}=0.2, C_{max}=0.8	[27]
MIIBSO	p_{r0}=0.8, $\mathcal{M}$=5, Δ=20, ∇=30	[28]
ABC	α=1, limit=100	[4]
CMA-ES	$\lambda = 4 + \lfloor 3 \ln n \rfloor$, $\mu = \lfloor 0.5\,\lambda \rfloor$	[11]
DE	CR=0.5, F=0.5	[10]
SADE	CR=$\mathcal{N}(CR_{mk}, 0.1)$, F=$\mathcal{N}(0.5, 0.3)$, CR_m=0.5	[41]
PSO	ω: 0.9-0.4, c1=2, c2=2	[3]
CLPSO	ω: 0.9-0.4, c=1.49445, sg=5, p_c = 0.05-0.5	[42]
ACO_R	q=0.5, σ=1	[43]
GLPSO	ω: 0.7298, c=1.49618, *sg*=7, *pm* = 0.01	[46]
MPEDE	CR=0.5, F=0.5	[47]
VGLBSO	$\mathcal{p}_l$=0.3, $\mathcal{p}_h$=0.6, $\mathcal{M}$=10, Δ =3, λ=0.1, η=25	

where Mean denotes the mean value of the error between $\mathcal{F}[\mathcal{X}]$ and $\mathcal{F}_{min}[\mathcal{X}^*]$ over K independent runs on each benchmark function of Table 1; $\mathcal{F}_{min}[\mathcal{X}^*]$ and $\mathcal{F}[\mathcal{X}]$ represent the fitness values of the global optimum solution $\mathcal{X}^*$ and the best solution $\mathcal{X}$ acquired by an algorithm, respectively.

$$\text{Std} = \sqrt[2]{\sum_{\kappa=1}^{K} [\mathcal{F}[\mathcal{X}] - \text{Mean}]^2 / (K-1)}. \quad (20)$$

Here, the value of K is set to 30.

Moreover, the Wilcoxon Signed-rank Test (WSRT) [44] is applied to identify the difference between a pair of the above different algorithms at a significant statistical level of 0.05. If a p-value from WSRT is below 0.05, there is a remarkable difference between a pair of algorithms on a benchmark function listed in Table 1. Otherwise, there is not a remarkable difference between them.

C. EXPERIMENTAL COMPARISONS BETWEEN BSO ALGORITHMS ON CEC2013 TEST SUITE

1) SOLUTIONS' COMPARISONS WITH MEAN AND *STD*

Table 3 lists the values of Mean and *Std* for the 13 BSO algorithms on the 28 CEC2013 functions (F_1-F_{28}). According to the values of Mean and *Std*, each BSO algorithm has been ranked on F_1-F_{28}. Note that the values of the best rank are highlighted in bold.

From Table 3, we can observe that the proposed VGLBSO has obtained the best rank 20 times on F_1-F_{28}. GBSO has acquired the best rank on F_{14}, F_{22}, and F_{23}. RGBSO, ADMBSO, BSODE, and BSODCS on F_{10} and F_{16}, on F_2, on F_4, and on F_8, respectively. VGLBSO fails to receive the best rank on F_2, F_4, F_8, F_{10}, F_{14}, F_{16}, F_{22}, and F_{23}, whereas it has never been ranked from the last to the bottom fifth on these functions. Evidently, VGLBSO has won the best overall performance on F_1-F_{28} among the 13 BSO algorithms. MIIBSO has received the second and third best rank for 12 and 5 times on F_1-F_{28}, respectively, so it wins the second best overall performance. GBSO does the third.

Considering the unimodal functions F_1-F_5, VGLBSO has obtained the best rank on F_1, F_3, and F_5; ADMBSO and BSODE have done the best rank on F_2 and F_4, respectively. MIIBSO has gained the second best rank on F_3-F_5. ADMBSO has received the third best results on F_3 and F_4. Table 3 has shown that the values of the final overall rank for VGLBSO, MIIBSO, and ADMBSO are the first, second, and third on F_1-F_5, respectively.

Given the multimodal functions F_6-F_{20}, VGLBSO has received the best rank for 11 times; RGBSO has done the best on F_{10} and F_{16}; BSODCS and GBSO have done the best on F_8 and F_{14}, respectively. Moreover, MIIBSO has received the second rank for 7 times and the third rank for twice on F_6-F_{20}; GBSO has done either the second or the third rank for three times on F_6-F_{20}, however, it has never received the first, second and third worst. Therefore, VGLBSO, MIIBSO, and GBSO have received the first, second, and third rank on the multimodal problems, respectively.

As for the composition functions F_{21}-F_{28}, VGLBSO has received the best rank for 6 times and the second rank for twice; GBSO has done the first and third rank for twice and three times on F_{21}-F_{28}, respectively; MIIBSO has obtained the second and third rank for twice and three times, respectively. For this reason, VGLBSO, GBSO, and MIIBSO have obtained the first, second, and third best rank on the composition functions, respectively.

TABLE 3. Comparisons of VGLBSO and 12 BSO Algorithms on the 28 CEC2013 benchmark functions with dimension 50.

Function	Evaluation Criteria	BSO	MBSO	CLBSO	PPBSO	SBSO	BSODE	RGBSO	ADMBSO	BSODCS	BSOCO	GBSO	MIIBSO	VGLBSO
F_1	Mean	1.51E-03	1.28E-12	5.68E-13	4.61E-01	2.52E-05	2.58E-13	2.73E-13	1.13E-12	2.27E-13	1.79E-03	2.27E-13	2.43E-13	**1.97E-13**
	Std	9.53E-04	1.19E-12	2.59E-13	9.74E-02	7.72E-05	7.86E-14	9.25E-14	1.63E-12	0.00E+00	1.75E-03	0.00E+00	5.77E-14	7.86E-14
	rank	10	9	7	13	10	5	6	8	2	12	2	4	**1**
F_2	Mean	2.05E+06	5.15E+05	6.08E+05	3.34E+06	2.24E+06	1.02E+06	7.22E+05	**4.16E+05**	7.11E+05	2.15E+06	2.76E+06	1.25E+06	8.64E+05
	Std	6.68E+05	1.98E+05	2.61E+05	1.28E+06	6.26E+05	3.50E+05	2.56E+05	1.44E+05	2.18E+05	6.64E+05	9.09E+05	5.98E+05	3.22E+05
	rank	8	2	3	13	11	7	5	**1**	4	10	12	8	6
F_3	Mean	2.20E+08	1.59E+08	1.24E+08	3.43E+08	7.89E+07	8.49E+07	8.12E+07	5.26E+07	8.06E+07	1.41E+08	3.84E+08	2.79E+07	**1.06E+07**
	Std	1.68E+08	2.17E+08	2.15E+08	2.16E+08	7.72E+07	1.02E+08	7.20E+07	5.63E+07	1.30E+08	1.09E+08	2.89E+08	3.94E+07	1.14E+07
	rank	10	10	8	12	4	7	6	3	5	9	13	2	**1**
F_4	Mean	1.55E+04	5.22E+02	4.55E+03	9.69E+04	4.45E+04	**2.04E+00**	6.51E+03	3.21E+02	1.27E+04	1.45E+04	9.33E+02	9.52E+00	1.60E+03
	Std	6.02E+03	4.62E+02	6.15E+03	2.17E+04	8.13E+03	1.55E+00	3.27E+03	1.53E+02	6.25E+03	4.38E+03	5.35E+02	1.49E+01	7.10E+02
	rank	10	4	7	13	12	**1**	8	3	9	10	5	2	6
F_5	Mean	4.27E-02	1.30E-12	7.73E-13	2.65E-01	1.84E-02	2.93E-03	3.48E-03	2.19E-12	3.29E-03	3.89E-02	9.16E-03	3.33E-13	**1.86E-13**
	Std	8.51E-03	1.09E-12	3.33E-13	6.74E-02	3.68E-03	3.11E-04	3.71E-04	5.50E-12	3.89E-04	7.73E-03	6.72E-04	8.92E-14	5.57E-14
	rank	11	4	3	13	10	6	8	5	7	11	9	2	**1**
F_6	Mean	6.13E+01	5.34E+01	5.20E+01	7.00E+01	7.62E+01	6.60E+01	6.05E+01	4.42E+01	5.53E+01	8.43E+01	5.98E+01	4.41E+01	**4.39E+01**
	Std	2.48E+01	1.96E+01	2.21E+01	2.48E+01	2.38E+01	2.43E+01	2.41E+01	1.32E+01	2.29E+01	3.46E+01	2.33E+01	1.73E+00	1.28E+00
	rank	8	5	4	11	12	10	8	3	6	13	7	2	**1**
F_7	Mean	2.13E+02	6.58E+01	6.57E+01	1.82E+02	1.69E+02	1.97E+02	2.52E+02	5.45E+01	2.19E+02	1.87E+02	4.34E+01	2.58E+01	**8.27E+00**
	Std	1.33E+02	1.67E+01	2.19E+01	7.91E+01	8.69E+01	6.92E+01	1.64E+02	1.60E+01	1.26E+02	8.63E+01	8.99E+00	2.11E+01	4.22E+00
	rank	10	6	5	8	7	10	13	4	12	9	3	2	**1**
F_8	Mean	21.10372	21.12322	21.12172	21.11800	21.13514	21.09116	21.09139	21.12339	**21.05193**	21.11095	21.13265	21.13572	21.09608
	Std	7.62E-02	4.05E-02	3.97E-02	4.46E-02	2.98E-02	4.85E-02	3.94E-02	4.57E-02	6.48E-02	5.62E-02	3.11E-02	3.57E-02	5.66E-02
	rank	4	9	8	7	12	2	3	10	**1**	6	11	13	4
F_9	Mean	5.81E+01	4.50E+01	4.67E+01	5.85E+01	4.84E+01	5.53E+01	5.89E+01	5.45E+01	5.72E+01	5.70E+01	3.38E+01	3.22E+01	**2.85E+01**
	Std	5.19E+00	6.98E+00	6.03E+00	4.72E+00	5.30E+00	4.88E+00	3.33E+00	8.75E+00	3.71E+00	4.04E+00	2.79E+00	6.07E+00	7.91E+00
	rank	10	4	5	12	6	8	13	7	10	9	3	2	**1**
F_{10}	Mean	1.28E+00	2.00E-01	1.74E-01	1.88E+00	1.19E+00	1.87E-02	**1.02E-02**	1.88E-01	1.36E-02	1.25E+00	4.67E-01	4.29E-02	1.29E-01
	Std	1.22E-01	9.85E-02	8.57E-02	2.49E-01	9.30E-02	1.01E-02	7.31E-03	9.43E-02	1.08E-02	9.98E-02	2.09E-01	2.10E-02	5.12E-02
	rank	11	8	6	13	10	3	**1**	7	2	11	9	4	5
F_{11}	Mean	7.06E+02	1.59E+02	1.33E+02	3.34E+02	5.17E+02	6.46E+02	8.37E+02	1.14E+02	7.52E+02	6.87E+02	7.33E+01	6.07E+01	**2.92E+01**
	Std	1.42E+02	4.03E+01	2.85E+01	6.69E+01	6.43E+01	1.17E+02	1.51E+02	2.76E+01	1.27E+02	1.17E+02	1.28E+01	1.48E+01	6.82E+00
	rank	10	6	5	7	8	9	13	4	12	10	3	2	**1**
F_{12}	Mean	6.75E+02	1.65E+02	1.39E+02	6.36E+02	5.27E+02	6.60E+02	8.98E+02	1.24E+02	7.87E+02	7.55E+02	1.51E+02	9.23E+01	**6.46E+01**
	Std	1.22E+02	3.48E+01	2.83E+01	9.54E+01	5.29E+01	1.28E+02	1.76E+02	2.65E+01	1.28E+02	9.31E+01	1.09E+02	2.16E+01	1.73E+01
	rank	9	6	4	8	7	9	13	3	12	11	5	2	**1**
F_{13}	Mean	9.06E+02	3.04E+02	2.56E+02	7.83E+02	7.24E+02	8.18E+02	9.36E+02	2.53E+02	9.71E+02	8.61E+02	3.05E+02	2.40E+02	**1.49E+02**
	Std	1.40E+02	5.05E+01	4.94E+01	1.32E+02	8.97E+01	1.19E+02	1.44E+02	4.45E+01	1.32E+02	1.06E+02	7.15E+01	9.08E+01	4.38E+01
	rank	10	5	4	8	7	9	12	3	13	10	6	2	**1**
F_{14}	Mean	7.32E+03	5.43E+03	5.80E+03	5.78E+03	6.73E+03	6.78E+03	7.37E+03	6.58E+03	7.38E+03	7.20E+03	**1.38E+02**	5.11E+03	2.29E+03
	Std	1.02E+03	1.18E+03	1.19E+03	9.79E+02	8.55E+02	1.04E+03	7.27E+02	9.90E+02	7.71E+02	8.40E+02	1.16E+02	2.52E+03	6.41E+02
	rank	10	4	6	5	8	9	12	7	13	10	**1**	3	2
F_{15}	Mean	7.94E+03	1.08E+04	1.15E+04	7.71E+03	8.18E+03	7.89E+03	8.18E+03	1.33E+04	7.92E+03	7.96E+03	9.80E+03	8.37E+03	**6.89E+03**
	Std	9.09E+02	2.74E+03	2.66E+03	8.29E+02	8.17E+02	8.55E+02	7.24E+02	1.94E+03	8.73E+02	7.81E+02	4.15E+03	2.43E+03	7.06E+02
	rank	4	11	12	2	7	3	8	13	4	6	10	9	**1**
F_{16}	Mean	3.56E-01	3.36E+00	3.46E+00	7.45E-01	1.14E+00	5.42E-02	**1.76E-02**	3.36E+00	1.96E-02	3.50E-01	3.29E+00	3.04E+00	1.24E+00
	Std	1.32E-01	3.03E-01	2.25E-01	1.61E-01	5.46E-01	2.11E-02	6.75E-03	2.92E-01	8.62E-03	1.07E-01	3.40E-01	1.02E+00	4.29E-01
	rank	5	12	13	6	7	3	**1**	11	2	4	10	9	8
F_{17}	Mean	9.44E+02	2.32E+02	1.91E+02	6.48E+02	7.77E+02	9.76E+02	1.16E+03	1.73E+02	1.13E+03	9.27E+02	1.13E+02	2.51E+02	**8.33E+01**
	Std	1.53E+02	5.46E+01	3.59E+01	7.73E+01	1.15E+02	1.74E+02	1.27E+02	3.61E+01	2.04E+02	1.50E+02	1.30E+01	1.16E+02	1.06E+01
	rank	9	5	4	7	8	11	13	3	12	9	2	6	**1**
F_{18}	Mean	7.68E+02	3.81E+02	4.03E+02	7.29E+02	9.10E+02	8.55E+02	9.88E+02	4.24E+02	9.57E+02	7.64E+02	3.82E+02	3.26E+02	**9.44E+01**
	Std	1.01E+02	1.10E+02	7.37E+01	8.70E+01	1.14E+02	1.93E+02	1.56E+02	6.79E+01	1.17E+02	7.88E+01	1.38E+01	1.03E+02	1.76E+01
	rank	8	3	5	7	11	10	13	6	12	8	4	2	**1**
F_{19}	Mean	1.71E+01	3.18E+01	2.42E+01	2.33E+01	2.98E+01	1.93E+01	1.42E+01	3.11E+01	1.51E+01	1.73E+01	7.73E+00	8.10E+00	**4.78E+00**
	Std	2.93E+00	1.28E+01	7.80E+00	3.76E+00	4.58E+00	2.93E+00	2.23E+00	1.63E+01	2.23E+00	2.96E+00	1.82E+00	6.33E+00	9.71E-01
	rank	5	13	10	9	11	8	4	12	5	7	2	3	**1**
F_{20}	Mean	2.41E+01	2.08E+01	2.08E+01	2.40E+01	2.21E+01	2.38E+01	2.43E+01	2.13E+01	2.43E+01	2.41E+01	2.07E+01	2.13E+01	**1.92E+01**
	Std	3.89E-01	1.45E+00	1.10E+00	7.96E-01	7.55E-01	5.84E-01	3.51E-01	4.87E-01	2.73E-01	6.54E-01	5.61E-01	9.89E-01	9.95E-01
	rank	10	3	4	9	7	8	13	6	12	10	2	5	**1**
F_{21}	Mean	7.64E+02	7.67E+02	7.50E+02	8.45E+02	9.16E+02	8.75E+02	8.83E+02	9.63E+02	7.70E+02	8.10E+02	8.54E+02	7.96E+02	**6.54E+02**
	Std	4.25E+02	4.21E+02	3.86E+02	3.46E+02	2.79E+02	3.01E+02	3.36E+02	2.86E+02	3.51E+02	3.91E+02	3.26E+02	4.34E+02	4.41E+02
	rank	2	4	2	8	12	10	11	13	5	7	9	6	**1**
F_{22}	Mean	1.03E+04	5.72E+03	6.42E+03	5.76E+03	9.68E+03	8.69E+03	1.03E+04	6.39E+03	1.05E+04	1.04E+04	**4.95E+02**	5.33E+03	2.14E+03
	Std	1.40E+03	1.13E+03	1.29E+03	9.97E+02	1.59E+03	1.46E+03	1.49E+03	1.71E+03	1.41E+03	1.43E+03	2.38E+02	1.78E+03	6.27E+02
	rank	9	4	7	5	9	8	11	6	13	12	**1**	3	2
F_{23}	Mean	1.03E+04	8.91E+03	9.66E+03	9.95E+03	1.04E+04	1.03E+04	1.04E+04	1.18E+04	1.03E+04	1.03E+04	**6.08E+03**	8.79E+03	6.99E+03
	Std	1.06E+03	2.09E+03	1.88E+03	1.14E+03	1.11E+03	1.01E+03	1.27E+03	2.84E+03	1.12E+03	1.16E+03	2.64E+03	2.21E+03	8.44E+02
	rank	9	4	5	6	12	9	11	13	8	7	**1**	3	2
F_{24}	Mean	4.05E+02	3.13E+02	3.10E+02	3.73E+02	4.07E+02	3.70E+02	4.22E+02	2.97E+02	4.19E+02	3.88E+02	2.87E+02	2.79E+02	**2.43E+02**
	Std	3.04E+01	1.43E+01	1.37E+01	2.08E+01	2.52E+01	1.77E+01	3.29E+01	1.40E+01	3.45E+01	2.26E+01	1.14E+01	1.46E+01	1.34E+01
	rank	9	6	5	8	11	7	13	4	12	9	3	2	**1**
F_{25}	Mean	5.15E+02	3.70E+02	3.62E+02	4.14E+02	5.17E+02	4.26E+02	5.10E+02	3.42E+02	5.03E+02	4.46E+02	3.50E+02	3.85E+02	**2.90E+02**
	Std	3.57E+01	2.40E+01	2.01E+01	1.57E+01	2.19E+01	1.85E+01	3.15E+01	2.29E+01	3.12E+01	1.81E+01	1.01E+01	2.75E+01	1.06E+01
	rank	11	5	4	7	13	8	11	2	10	9	3	6	**1**
F_{26}	Mean	4.33E+02	3.85E+02	4.00E+02	4.19E+02	3.79E+02	4.03E+02	4.24E+02	3.58E+02	3.93E+02	4.24E+02	3.84E+02	3.72E+02	**3.36E+02**
	Std	6.67E+01	6.46E+01	4.11E+01	7.49E+01	1.01E+02	9.25E+01	9.21E+01	9.22E+01	1.09E+02	7.22E+01	1.26E+01	5.02E+01	3.98E+01
	rank	12	6	8	10	4	9	12	2	7	11	5	3	**1**
F_{27}	Mean	2.01E+03	1.33E+03	1.41E+03	1.87E+03	1.90E+03	1.85E+03	2.11E+03	1.28E+03	2.07E+03	1.97E+03	1.22E+03	1.19E+03	**7.97E+02**
	Std	1.19E+02	1.22E+02	1.67E+02	1.65E+02	1.29E+02	1.58E+02	1.64E+02	1.86E+02	1.60E+02	1.22E+02	1.12E+02	1.69E+02	1.00E+02
	rank	10	5	6	8	9	7	13	4	12	10	3	2	**1**
F_{28}	Mean	7.15E+03	1.19E+03	9.31E+02	6.75E+03	6.34E+03	5.29E+03	7.19E+03	8.46E+02	7.21E+03	6.97E+03	1.28E+03	1.39E+03	**7.94E+02**
	Std	6.71E+02	1.47E+03	1.21E+03	1.37E+03	5.93E+02	2.05E+03	9.33E+02	1.16E+03	8.02E+02	7.87E+02	1.50E+03	1.43E+03	1.02E+03
	rank	10	4	3	9	8	7	12	2	13	10	5	6	**1**
Unimodal	Average Rank	9.8	5.8	5.6	12.8	9.4	5.2	6.6	4	5.4	10.4	8.2	3.6	**3**
Unimodal	Final Rank	11	7	6	13	10	4	8	3	5	12	9	2	**1**
Multimodal	Average Rank	8.2	6.67	6.33	7.93	8.53	7.47	9.33	6.6	8.53	8.87	5.2	4.4	**2**
Multimodal	Final Rank	9	6	4	8	10	7	13	5	10	12	3	2	**1**
Multimodal Composition	Average Rank	9	4.75	5	7.62	9.75	8.12	11.75	5.75	10	9.37	3.75	3.87	**1.25**
Multimodal Composition	Final Rank	9	4	5	7	11	8	13	6	12	10	2	3	**1**
Overall	Average Rank	8.71	5.96	5.82	8.71	9.04	7.25	9.54	5.89	8.39	9,29	5.32	4.11	**1.96**
Overall	Final Rank	9	6	4	9	11	7	13	5	8	12	3	2	**1**

TABLE 4. Comparisons between VGLBSO and each of 12 BSO algorithms on the 28 CEC2013 benchmark functions with 50-$\mathcal{D}$ by WSRT.

	Pairwise Comparison: VGLBSO Versus											
	BSO	MBSO	CLBSO	PPBSO	SBSO	BSODE	RGBSO	ADMBSO	BSO_DCS	BSOCO	GBSO	MIIBSO
F_1	+	+	+	+	+	+	+	+	=	+	=	+
F_2	+	-	-	+	+	=	=	-	-	+	+	+
F_3	+	+	+	+	+	+	+	+	+	+	+	=
F_4	+	-	+	+	+	-	+	-	+	+	-	-
F_5	+	+	+	+	+	+	+	+	+	+	+	+
F_6	+	+	=	+	+	+	+	=	+	+	+	+
F_7	+	+	+	+	+	+	+	+	+	+	+	+
F_8	=	+	=	=	+	=	=	=	-	=	+	+
F_9	+	+	+	+	+	+	+	+	+	+	+	=
F_{10}	+	+	+	+	+	-	-	+	-	+	+	-
F_{11}	+	+	+	+	+	+	+	+	+	+	+	+
F_{12}	+	+	+	+	+	+	+	+	+	+	+	+
F_{13}	+	+	+	+	+	+	+	+	+	+	+	+
F_{14}	+	+	+	+	+	+	+	+	+	+	-	+
F_{15}	+	+	+	+	+	+	+	+	+	+	+	+
F_{16}	-	+	+	-	=	-	-	+	-	-	+	+
F_{17}	+	+	+	+	+	+	+	+	+	+	+	+
F_{18}	+	+	+	+	+	+	+	+	+	+	+	+
F_{19}	+	+	+	+	+	+	+	+	+	+	+	+
F_{20}	+	+	+	+	+	+	+	+	+	+	+	+
F_{21}	=	+	+	+	+	+	+	+	=	+	+	+
F_{22}	+	+	+	+	+	+	+	+	+	+	-	+
F_{23}	+	+	+	+	+	+	+	+	+	+	-	+
F_{24}	+	+	+	+	+	+	+	+	+	+	+	+
F_{25}	+	+	+	+	+	+	+	+	+	+	+	+
F_{26}	+	+	+	+	=	+	+	=	=	+	+	+
F_{27}	+	+	+	+	+	+	+	+	+	+	+	+
F_{28}	+	+	+	+	+	+	+	+	+	+	+	+
"+" / "-" / "="	25/1/2	26/2/0	25/1/2	26/1/1	26/0/2	23/3/2	24/2/2	23/2/3	21/4/3	26/1/1	23/4/1	24/2/2

2) STATISTICAL ANALYSIS WITH WSRT

Table 4 has listed the statistical results of WSRT between VGLBSO and each of the 12 BSO algorithms on F_1-F_{28}. From Table 4, signs "+", "−", and "=" signify that VGLBSO is superior to, inferior to, and almost equivalent to the compared BSO algorithm, respectively. For instance, a pairwise comparison between VGLBSO and MIIBSO is given in the first column on the right of Table 4, where 24 signs "+" imply that VGLBSO is superior to MIIBSO on 24 out of the 28 functions, two signs "−" suggest that VGLBSO is inferior to MIIBSO on F_4 and F_{10}, and two signs "=" signify that VGLBSO is almost equivalent to MIIBSO on F_3 and F_9. From 12 sets of pairwise comparative results shown in Table 4, VGLBSO outperforms the 12 BSO algorithms.

3) CONVERGENCE ANALYSIS

We utilize convergence curves to characterize the convergence characteristics of 13 BSO algorithms, which are sketched in Fig. s1 of Section S-I of the supplementary file owing to space constraints. Fig. s1 shows that VGLBSO wins the best convergence characteristics for 20 times on F_1-F_{28}, excluding F_2, F_4, F_8, F_{10}, F_{14}, F_{16}, F_{22}, and F_{23} among the 13 BSO algorithms. Besides, VGLBSO has the second best convergence speed on F_{14}, F_{22}, and F_{23}, among all the 13 BSO algorithms. In addition, VGLBSO has provided the fourth, the fifth, the sixth, and the eight best convergence speed on F_8, on F_{10}, on F_2 and F_4, and on F_{16}, respectively. Fortunately, VGLBSO never received from the first to fifth worst convergence speed among the 13 BSO algorithms.

Furthermore, we also give pairwise comparisons involving the convergence characteristics between VGLBSO and each of the 12 BSO algorithms. Fig. s1 of Section S-I of the supplementary file exhibits that VGLBSO has acquired faster convergence speed than each of the 12 BSO algorithms. Taking the comparisons of convergence speed between VGLBSO and GBSO as an example, the former has faster speed than the latter on 24 out of the 28 functions excluding on F_4, F_{14}, F_{22}, and F_{23}.

Even though VGLBSO cannot attain the best convergence property on every function among the 13 BSO algorithms, it can achieve the relatively better convergence property compared with the each of the 12 BSO algorithms on most of the 28 CEC2013 functions.

D. EXPERIMENTAL COMPARISONS BETWEEN VGLBSO AND NINE ALGORITHMS ON CEC2013 TEST SUITE

1) SOLUTIONS' COMPARISONS WITH MEAN AND STD

Table 5 illustrates the comparative results between the proposed VGLBSO and the nine swarm intelligence algorithms

TABLE 5. Comparisons of VGLBSO and nine swarm intelligence algorithms on the 28 CEC2013 benchmark functions with dimension 50.

Function	Evaluation Criteria	ABC	CMAES	DE	CLPSO	PSO	ACOR	SADE	GLPSO	MPEDE	VGLBSO
F_1	Mean	1.38E-12	**0.00E+00**	2.27E-13	2.27E-13	9.09E-13	7.93E+00	**0.00E+00**	6.82E-13	3.26E-13	1.97E-13
	Std	1.78E-13	0.00E+00	0.00E+00	0.00E+00	1.89E-13	3.55E+00	0.00E+00	2.31E-13	1.29E-13	7.86E-14
	rank	9	**1**	4	4	8	10	**1**	7	6	3
F_2	Mean	2.53E+07	3.18E+07	2.77E+08	4.10E+07	3.36E+07	4.35E+09	1.97E+05	7.77E+05	**6.05E+04**	8.64E+05
	Std	4.02E+06	1.35E+07	3.99E+07	7.99E+06	1.38E+07	7.84E+08	4.97E+04	4.47E+05	3.36E+04	3.22E+05
	rank	5	6	9	8	7	10	2	3	**1**	4
F_3	Mean	4.11E+09	**0.00E+00**	1.81E+03	2.42E+09	1.33E+09	5.32E+12	2.36E+07	4.38E+08	5.81E+07	1.06E+07
	Std	1.90E+09	0.00E+00	8.46E+03	7.67E+08	1.09E+09	1.69E+12	3.09E+07	5.82E+08	5.79E+07	1.14E+07
	rank	9	**1**	2	8	7	10	4	6	5	3
F_4	Mean	1.36E+05	2.86E+05	4.41E+04	2.80E+04	9.49E+03	7.57E+05	1.20E+03	1.21E+04	**3.41E-01**	1.60E+03
	Std	1.17E+04	4.81E+04	5.30E+03	4.00E+03	1.93E+03	4.19E+05	7.47E+02	4.39E+03	9.08E-01	7.10E+02
	rank	8	9	7	6	4	10	2	5	**1**	3
F_5	Mean	3.08E-12	6.69E-04	1.14E-13	3.22E-13	8.68E-13	1.45E+03	**0.00E+00**	8.37E-13	2.80E-13	1.86E-13
	Std	6.91E-13	1.33E-04	0.00E+00	4.31E-14	1.88E-13	4.01E+02	0.00E+00	1.90E-13	7.75E-14	5.57E-14
	rank	8	9	2	5	7	10	**1**	6	4	3
F_6	Mean	4.18E+01	**1.24E+01**	4.34E+01	4.66E+01	5.20E+01	7.03E+01	5.05E+01	4.56E+01	1.46E+01	4.39E+01
	Std	3.97E+00	4.16E+00	2.10E-12	5.38E-01	1.60E+01	1.93E+01	2.06E+01	1.71E+01	6.74E+00	1.28E+00
	rank	3	**1**	4	7	9	10	8	6	2	5
F_7	Mean	1.68E+02	**2.27E-14**	1.48E+01	9.71E+01	8.09E+01	1.63E+03	4.27E+01	7.72E+01	8.01E+01	8.27E+00
	Std	1.34E+01	4.63E-14	5.43E+00	9.06E+00	2.29E+01	3.31E+02	9.20E+00	1.28E+01	1.84E+01	4.22E+00
	rank	9	**1**	3	8	7	10	4	5	6	2
F_8	Mean	21.13343	21.16516	21.13160	21.12422	21.13141	21.13185	21.13672	21.13962	21.12713	21.09608
	Std	4.25E-02	3.29E-02	4.56E-02	4.18E-02	2.88E-02	3.77E-02	3.22E-02	5.86E-02	4.10E-02	5.66E-02
	rank	7	10	5	2	4	6	8	9	3	**1**
F_9	Mean	5.92E+01	7.24E+01	7.26E+01	5.41E+01	4.61E+01	7.31E+01	3.80E+01	5.05E+01	5.25E+01	**2.85E+01**
	Std	1.82E+00	1.26E+01	1.07E+00	2.13E+00	4.17E+00	1.46E+00	4.63E+00	5.79E+00	5.31E+00	7.91E+00
	rank	7	8	9	6	3	10	2	4	5	**1**
F_{10}	Mean	3.54E+00	1.58E+00	**2.47E-02**	8.10E+00	2.65E+00	1.23E+04	2.34E-01	1.29E-01	3.97E-02	1.29E-01
	Std	8.94E-01	2.73E-01	2.70E-02	1.86E+00	1.65E+00	2.21E+03	1.15E-01	7.32E-02	3.30E-02	5.12E-02
	rank	8	6	**1**	9	7	10	5	3	2	4
F_{11}	Mean	1.23E-08	5.69E+01	2.42E+02	**1.08E-13**	5.48E+01	5.09E+02	3.98E-01	1.34E-11	2.12E+01	2.92E+01
	Std	6.68E-08	1.09E+02	1.19E+01	2.29E-14	8.92E+00	2.14E+01	5.60E-01	6.99E-11	1.76E+01	6.82E+00
	rank	3	8	9	**1**	7	10	4	2	5	6
F_{12}	Mean	7.19E+02	3.10E+02	3.89E+02	2.78E+02	2.12E+02	5.60E+02	1.25E+02	1.57E+02	1.68E+02	**6.46E+01**
	Std	8.45E+01	1.53E+01	1.13E+01	2.47E+01	9.22E+01	2.70E+01	2.19E+01	2.84E+01	3.56E+01	1.73E+01
	rank	10	7	8	6	5	9	2	3	4	**1**
F_{13}	Mean	7.82E+02	3.14E+02	3.79E+02	3.56E+02	3.64E+02	5.52E+02	2.38E+02	3.01E+02	3.22E+02	**1.49E+02**
	Std	7.28E+01	9.74E+00	1.70E+01	2.26E+01	6.27E+01	3.59E+01	3.34E+01	5.53E+01	5.86E+01	4.38E+01
	rank	10	4	8	6	7	9	2	3	5	**1**
F_{14}	Mean	1.55E+01	1.36E+04	9.66E+03	1.15E+01	1.74E+03	1.59E+04	1.86E+01	**1.32E+00**	2.41E+02	2.29E+03
	Std	3.54E+00	4.51E+02	4.12E+02	3.35E+00	4.68E+02	4.93E+02	2.53E+01	7.82E-01	1.18E+02	6.41E+02
	rank	3	9	8	2	6	10	4	**1**	5	7
F_{15}	Mean	8.40E+03	1.38E+04	1.40E+04	8.90E+03	1.36E+04	1.63E+04	8.81E+03	8.28E+03	7.53E+03	**6.89E+03**
	Std	4.38E+02	4.08E+02	2.63E+02	4.75E+02	5.84E+02	4.86E+02	2.08E+03	7.92E+02	6.57E+02	7.06E+02
	rank	4	8	9	6	7	10	5	3	2	**1**
F_{16}	Mean	2.16E+00	**0.00E+00**	3.32E+00	2.31E+00	3.16E+00	3.25E+00	2.98E+00	1.69E+00	2.77E+00	1.24E+00
	Std	2.10E-01	0.00E+00	3.06E-01	2.77E-01	3.55E-01	2.28E-01	3.03E-01	6.80E-01	8.15E-01	4.29E-01
	rank	4	**1**	10	5	8	9	7	3	6	2
F_{17}	Mean	**5.10E+01**	3.58E+02	2.96E+02	5.43E+01	1.35E+02	5.69E+02	5.20E+01	5.21E+01	5.62E+01	8.33E+01
	Std	5.38E-02	9.75E+00	1.28E+01	5.57E-01	3.11E+01	1.87E+01	1.47E+00	5.13E-01	2.25E+00	1.06E+01
	rank	**1**	9	8	4	7	10	2	3	5	6
F_{18}	Mean	8.13E+02	3.60E+02	4.33E+02	4.16E+02	5.04E+02	5.72E+02	2.28E+02	1.78E+02	1.71E+02	**9.44E+01**
	Std	5.51E+01	1.20E+01	1.29E+01	2.28E+01	4.17E+01	2.11E+01	8.00E+01	2.86E+01	3.18E+01	1.76E+01
	rank	10	5	7	6	8	9	4	3	2	**1**
F_{19}	Mean	**2.65E+00**	1.59E+01	2.80E+01	2.87E+00	9.46E+00	5.95E+05	9.34E+00	3.06E+00	7.93E+00	4.78E+00
	Std	4.78E-01	6.77E+00	1.10E+00	4.57E-01	1.93E+00	5.36E+05	1.52E+00	8.86E-01	2.08E+00	9.71E-01
	rank	**1**	8	9	2	7	10	6	3	5	4
F_{20}	Mean	2.44E+01	2.49E+01	2.23E+01	2.30E+01	2.37E+01	2.45E+01	1.99E+01	2.26E+01	2.02E+01	**1.92E+01**
	Std	2.03E-01	4.42E-01	2.42E-01	4.14E-01	1.51E+00	9.88E-02	6.47E-01	1.46E+00	1.28E+00	9.95E-01
	rank	8	10	4	6	7	9	2	5	3	**1**
F_{21}	Mean	**2.43E+02**	6.00E+02	7.36E+02	3.86E+02	6.52E+02	1.72E+03	8.71E+02	8.59E+02	7.15E+02	6.54E+02
	Std	5.53E+01	4.65E+02	4.27E+02	1.89E+02	4.62E+02	5.86E+02	3.61E+02	3.85E+02	4.38E+02	4.41E+02
	rank	**1**	3	7	2	4	10	9	8	6	5
F_{22}	Mean	**4.56E+01**	1.38E+04	1.01E+04	5.29E+01	2.00E+03	1.65E+04	7.26E+01	6.59E+01	3.69E+02	2.14E+03
	Std	2.31E+01	2.59E+02	4.78E+02	3.00E+01	5.05E+02	5.11E+02	2.55E+02	8.53E+01	2.01E+02	6.27E+02
	rank	**1**	9	8	2	6	10	4	3	5	7
F_{23}	Mean	1.06E+04	1.38E+04	1.43E+04	1.10E+04	1.34E+04	1.66E+04	8.22E+03	9.70E+03	7.63E+03	**6.99E+03**
	Std	4.54E+02	4.17E+02	4.13E+02	6.51E+02	9.47E+02	3.89E+02	2.07E+03	1.14E+03	1.18E+03	8.44E+02
	rank	5	8	9	6	7	10	3	4	2	**1**
F_{24}	Mean	3.73E+02	3.82E+02	**2.01E+02**	3.43E+02	3.24E+02	3.96E+02	2.65E+02	3.23E+02	3.36E+02	2.43E+02
	Std	6.83E+00	3.52E+00	1.82E-01	6.25E+00	1.61E+01	3.90E+00	1.13E+01	1.23E+01	1.28E+01	1.34E+01
	rank	8	9	**1**	7	5	10	3	4	6	2
F_{25}	Mean	4.14E+02	3.82E+02	4.14E+02	3.82E+02	3.62E+02	3.90E+02	3.40E+02	3.59E+02	3.43E+02	**2.90E+02**
	Std	8.05E+00	3.28E+00	5.82E+00	7.60E+00	1.49E+01	2.76E+00	9.73E+00	1.42E+01	1.59E+01	1.06E+01
	rank	9	7	10	6	5	8	2	4	3	**1**
F_{26}	Mean	**2.02E+02**	4.56E+02	3.21E+02	2.04E+02	4.01E+02	4.97E+02	2.27E+02	4.06E+02	4.17E+02	3.36E+02
	Std	4.36E-01	7.13E+01	1.13E+02	1.31E+00	5.52E+01	2.85E+00	6.23E+01	5.87E+01	4.32E+01	3.98E+01
	rank	**1**	9	4	2	6	10	3	7	8	5
F_{27}	Mean	**5.31E+02**	2.12E+03	1.45E+03	1.60E+03	1.49E+03	2.22E+03	1.09E+03	1.55E+03	1.63E+03	7.97E+02
	Std	4.05E+02	5.23E+01	5.07E+02	4.67E+02	1.17E+02	5.59E+01	1.09E+02	1.35E+02	1.45E+02	1.00E+02
	rank	**1**	9	4	7	5	10	3	6	8	2
F_{28}	Mean	4.00E+02	2.92E+03	**4.00E+02**	4.00E+02	1.85E+03	3.44E+03	**4.00E+02**	7.41E+02	6.10E+02	7.94E+02
	Std	6.01E-10	1.00E+03	3.10E-13	5.81E-04	1.69E+03	1.36E+03	4.22E-14	1.04E+03	7.98E+02	1.02E+03
	rank	3	9	**2**	4	8	10	**1**	6	5	7
unimodal	Average Rank	7.8	5.2	4.8	6.2	6.6	10	**2**	5.4	3.4	3.2
	Final Rank	9	5	4	7	8	10	**1**	6	3	2
multimodal	Average Rank	5.93	6.40	6.87	5.13	6.67	9.47	4.40	3.80	4.07	**2.93**
	Final Rank	6	7	9	5	8	10	4	2	3	**1**
composition	Average Rank	3.625	7.875	5.5	4.5	5.75	9.75	**3.5**	5.25	5.375	3.75
	Final Rank	2	9	7	4	8	10	**1**	5	6	3
overall	Average Rank	5.57	6.57	6.07	5.11	6.36	9.61	3.68	4.46	4.28	**3.18**
	Final Rank	6	9	7	5	8	10	2	4	3	**1**

to further validate its advantages. From Table 5, we can see that the ten algorithms have been ranked through the values of Mean and *Std* on F_1-F_{28}. The best rank is highlighted in bold. Interestingly, VGLBSO has received the best rank on nine out of the 28 functions F_1-F_{28}, corresponding to F_8, F_9, F_{12}, F_{13}, F_{15}, F_{18}, F_{20}, F_{23}, and F_{25}; ABC has done the best rank on six, namely F_{17}, F_{19}, F_{21}, F_{22}, F_{26}, and F_{27}; CMA-ES has also done the best rank five times, involving F_1, F_3, F_6, F_7, and F_{16}; SADE has done the best rank three times, including F_1, F_5, and F_{28}; MPEDE and DE have obtained the best rank on F_2 and F_4 and on F_{10} and F_{24}, respectively. CLPSO and GLPSO have done the best on F_{11} and F_{14}, respectively. Moreover, from Table 5, SADE has done the second best rank for eight times, covering F_2, F_4, F_9, F_{12}, F_{13}, F_{17}, F_{20}, and F_{25}; CLPSO has done the second best rank on six test functions: F_8, F_{14}, F_{19}, F_{21} F_{22}, and F_{26}; MPEDE has done the second best rank for five times, containing F_6, F_{10}, F_{15}, F_{18}, and F_{23}; VGLBSO has received the second best rank for four times, corresponding to F_7, F_{16}, F_{24}, and F_{27}; DE has obtained the second best rank on F_3, F_5, and F_{28}; GLPSO has the second best rank on F_{11}.

Furthermore, on F_1-F_{28}, ABC has been ranked from the first to the third worst for three, four, and five times, respectively. CMA-ES has been ranked from the first to third worst for twice, nine, and five times, respectively. DE has the first, second, and third worst rank for one, six, and five times, respectively; CLPSO has the second and third worst rank for one and three times, respectively; GLPSO has been ranked the second or third worst for one time; MPEDE has been ranked the third worst for twice. SADE, MPEDE, GLPSO, and CLPSO have never received the worst rank on F_1-F_{28}. Interestingly, among the ten swarm intelligence method, VGLBSO is the only method that has never received the first, the second, and the third worst rank on F_1-F_{28}.

Thus, among the ten algorithms, VGLBSO has received the best overall performance; SADE and MPEDE have done the second and third, respectively.

Given the unimodal problems F_1-F_5, SADE has received the best rank on F_1, and F_5; CMA-ES has done the best on F_1 and F_3, MPEDE has received the best on F_2, and F_4. Moreover, SADE and DE have acquired the second rank on F_2 and F_4 and on F_3 and F_5, respectively. In addition, VGLBSO has provided the third rank on F_1, F_3, F_4, and F_5. Unfortunately, CMA-ES has received the ninth rank on F_4 and F_5; MPEDE has done the sixth and fifth on F_1 and F_3, respectively. Therefore, SADE, VGLBSO, and MPEDE win the first, second, and third overall rank on F_1-F_5, respectively.

Considering the multimodal problems F_6-F_{20}, an interesting observation is that VGLBSO has won the best rank seven times; CMA-ES has done the best rank on F_6, F_7, and F_{16}; ABC have done the best rank on F_{17} and F_{19}. CLPSO, GLPSO, and DE have received the best only on F_{11}, on F_{14} and on F_{10}, respectively. Furthermore, SADE has received the second rank five times; MPEDE has done the second on F_6, F_{10}, F_{15}, and F_{18}; CLPSO has done the second on F_8, F_{14}, and F_{19}; VGLBSO has done the second on F_7 and F_{16}. Particularly, GLPSO has received the third rank eight times. Fortunately, excluding F_8, GLPSO has never been ranked the last on F_6-F_{20}; more interestingly, either VGLBSO or MPEDE has never received the first, the second, and the third worst rank on F_6-F_{20} among all the ten algorithms. Therefore, VGLBSO has received the first overall rank on the multimodal problems, followed by GLPSO and MPEDE that achieve the second and third overall rank, respectively.

Involving the composition problems F_{21}-F_{28}, ABC has won the best rank on four functions: F_{21}, F_{22}, F_{26}, and F_{27}; VGLBSO has done the best on F_{23} and F_{25}; DE and SADE have done the best on F_{24} and F_{28}, respectively. Next, CLPSO has received the second best rank on F_{21}, F_{22}, and F_{26}; VGLBSO and SADE have done the second best on F_{24} and F_{27} and on F_{25}, respectively; MPEDE and DE have done the second best only on F_{23} and F_{28}. In addition, SADE has obtained the third best rank on F_{23}, F_{24}, F_{26}, and F_{27}; CMA-ES, GLPSO, MPEDE, and ABC have acquired the third best on F_{21}, F_{22}, F_{25}, and F_{28}, respectively. Interestingly, excluding F_{21}, SADE have never gained received a rank greater than 5; further, VGLBSO has never received the first, the second, and the third worst rank on F_{21}-F_{28} among all the ten algorithms. Therefore, SADE, ABC, and VGLBSO have achieved the first, second, and third overall rank on the composition problems F_{21}-F_{28}.

2) STATISTICAL ANALYSIS WITH WSRT

Table 6 has presented the experimental results of WSRT between VGLBSO and each of the nine swarm intelligence algorithms on F_1-F_{28}. Here, signs "+", "−", and "=" illustrate that VGLBSO is superior to, inferior to, and almost equivalent to the compared swarm intelligence algorithm, respectively. As an example, the pairwise comparison between VGLBSO and SADE is given in the first column on the right of Table 6, where 17 signs "+" show that VGLBSO is superior to SADE on 17 out of the 28 functions, 10 signs "−" suggest that VGLBSO is inferior to SADE on 10, and one sign "=" signifies that VGLBSO is almost equivalent to SADE on F_{21}. According to the seven sets of pairwise comparisons shown in Table 6, VGLBSO clearly outperforms the nine swarm intelligence algorithms.

3) CONVERGENCE ANALYSIS

The convergence curves are utilized to characterize the convergence characteristics of the ten swarm intelligence algorithms, sketched in Fig. s2 of Section S-II of the supplementary file due to space limitation. Fig. s2 has shown that VGLBSO wins the best convergence characteristics on nine functions F_8, F_9, F_{12}, F_{13}, F_{15}, F_{18}, F_{20}, F_{23}, and F_{25} among the ten algorithms. Moreover, VGLBSO has the second best convergence speed on F_7, F_{16}, F_{24}, and F_{27}. In addition, VGLBSO has the third best convergence speed on F_1, F_3, F_4, and F_5. Particularly, among all the ten algorithms, VGLBSO is the only algorithm that has never provided the first, second, and third worst convergence speed on F_1-F_{28}.

TABLE 6. Comparisons between VGLBSO and each of nine swarm intelligence algorithms on the 28 CEC2013 benchmark functions with 50-$\mathcal{D}$ by WSRT.

	Pairwise Comparison: VGLBSO Versus								
	ABC	CMAES	DE	CLPSO	PSO	ACOR	SADE	GLPSO	MPEDE
F_1	+	-	=	=	+	+	-	+	+
F_2	+	+	+	+	+	+	-	=	-
F_3	+	-	-	+	+	+	+	+	+
F_4	+	+	+	+	+	+	-	+	-
F_5	+	+	-	+	+	+	-	+	+
F_6	=	-	-	+	+	+	+	=	-
F_7	+	-	+	+	+	+	+	+	+
F_8	+	+	+	=	+	+	+	+	+
F_9	+	+	+	+	+	+	+	+	+
F_{10}	+	+	-	+	+	+	+	=	-
F_{11}	-	=	+	-	+	+	-	-	-
F_{12}	+	+	+	+	+	+	+	+	+
F_{13}	+	+	+	+	+	+	+	+	+
F_{14}	-	+	+	-	-	+	-	-	-
F_{15}	+	+	+	+	+	+	+	+	+
F_{16}	+	-	+	+	+	+	+	+	+
F_{17}	-	+	+	-	+	+	-	-	-
F_{18}	+	+	+	+	+	+	+	+	+
F_{19}	-	+	+	-	+	+	+	-	+
F_{20}	+	+	+	+	+	+	+	+	+
F_{21}	-	=	=	-	=	+	=	+	=
F_{22}	-	+	+	-	=	+	-	-	-
F_{23}	+	+	+	+	+	+	+	+	+
F_{24}	+	+	-	+	+	+	+	+	+
F_{25}	+	+	+	+	+	+	+	+	+
F_{26}	-	+	=	-	+	+	-	+	+
F_{27}	-	+	+	+	+	+	+	+	+
F_{28}	+	+	=	+	+	+	-	+	+
"+"/"-"/ "="	19/8/1	21/5/2	19/5/4	19/7/2	25/1/2	28/0/0	17/10/1	20/5/3	19/8/1

TABLE 7. Average rank, final rank, and WSRT of VGLBSO and 12 BSO algorithms on 28 CEC2013 functions with dimensions 30 and 50.

Function	Dimension	Evaluation Criteria	BSO	MBSO	CLBSO	PPBSO	SBSO	BSODE	RGBSO	ADMBSO	BSO_DCS	BSOCO	GBSO	MIIBSO	VGLBSO
Overall	30- $\mathcal{D}$	Average Rank	8.32	5.64	5.71	9.21	8.39	7.32	9.25	5.04	9.18	9.14	5.61	4.68	**2.36**
	30- $\mathcal{D}$	Final Rank	8	5	6	12	9	7	13	3	11	10	4	2	**1**
	50- $\mathcal{D}$	Average Rank	8.71	5.96	5.82	8.71	9.04	7.25	9.54	5.89	8.39	9,29	5.32	4.11	**1.96**
	50- $\mathcal{D}$	Final Rank	9	6	4	9	11	7	13	5	8	12	3	2	**1**
WSRT	30- $\mathcal{D}$	"+"/ "-" / "="	24/1/3	23/2/3	23/1/4	24/1/3	24/1/3	20/3/5	21/3/4	23/3/2	21/3/4	25/1/2	21/3/4	16/4/8	\
	50- $\mathcal{D}$	"+"/ "-" / "="	25/1/2	26/2/0	25/1/2	26/1/1	26/0/2	23/3/2	24/2/2	23/2/3	21/4/3	26/1/1	23/4/1	24/2/2	\

Additionally, we further conduct a set of pairwise comparisons involving the convergence characteristics between VGLBSO and each of the seven algorithms. Fig. s2 of Section S-II of the supplementary file presents that VGLBSO has faster convergence speed than each of them. For example, the comparison between VGLBSO and SADE shows that the former has faster speed than the latter on 18 out of the 28 functions.

Although VGLBSO is unable to gain the best convergence among all the ten algorithms, it does the relatively better on most of the 28 CEC2013 functions.

E. SCALABILITY ANALYSIS

The scalability analysis is allowed to distinguish whether the overall performance of the algorithms may strikingly deteriorate with the dimension of the benchmark functions increasing from 30-$\mathcal{D}$ to 50-$\mathcal{D}$. Due to the space limitation, for the 30-$\mathcal{D}$ problems of the functions F_1-F_{28}, the detailed experimental results from VGLBSO, the 12 BSO, and the nine swarm intelligence algorithms have been given in Tables s1, s2 s3, and s4 of the Section S-III of the supplementary file. In addition, for 30-$\mathcal{D}$ problems, the population size and the maximum number of the iteration are set to 50 and 300000. Here, for simplicity, we only provide the average rank, final rank, and statistics of WSRT on the 30-$\mathcal{D}$ and 50-$\mathcal{D}$ problem of F_1-F_{28} for evaluating the scalability of the above algorithms.

First, Table 7 has presented the values of the average and final rank of VGLBSO and the 12 BSO variants on F_1-F_{28} with 30-$\mathcal{D}$ and 50-$\mathcal{D}$ problems. We can observe that as the

TABLE 8. Average rank, final rank, and WSRT of VGLBSO and nine swarm intelligence algorithms on 28 CEC2013 functions with dimensions 30 and 50.

Function	Dimension	Evaluation Criteria	ABC	CMAES	DE	CLPSO	PSO	ACOR	SADE	GLPSO	MPEDE	VGLBSO
overall	30-$\mathcal{D}$	Average Rank	5.68	6.93	5.61	5.53	6.5	9.18	3.36	4.68	4.07	3.14
	30-$\mathcal{D}$	Final Rank	7	9	6	5	8	10	2	4	3	**1**
	50-$\mathcal{D}$	Average Rank	5.57	6.57	6.07	5.11	6.36	9.61	3.68	4.46	4.28	3.18
	50-$\mathcal{D}$	Final Rank	6	9	7	5	8	10	2	4	3	**1**
WSRT	30-$\mathcal{D}$	"+"/"-"/"="	17/8/3	21/4/3	17/5/6	21/5/2	24/1/3	28/0/0	14/9/5	18/6/4	13/8/7	\
	50-$\mathcal{D}$	"+"/"-"/"="	19/8/1	21/5/2	19/5/4	19/7/2	25/1/2	28/0/0	17/10/1	20/5/3	19/8/1	\

dimensional number increases from 30-$\mathcal{D}$ to 50-$\mathcal{D}$, the overall performance of VGLBSO does not degenerate. Furthermore, Table 7 has also presented the statistics of WSRT between VGLBSO and the each of the 12 BSO variants on F_1-F_{28} with 30-$\mathcal{D}$ and 50-$\mathcal{D}$ problems. Likewise, with the dimensional number increasing from 30-$\mathcal{D}$ to 50-$\mathcal{D}$, the overall performance of VGLBSO does not produce significant attenuation.

Furthermore, Tables 8 provides further results for scalability analysis between VGLBSO and the nine swarm intelligence algorithms on F_1-F_{28} with the 30-$\mathcal{D}$ and 50-$\mathcal{D}$ problem. Similarly, we can see that the overall performance of VGLBSO still does not significantly decay when the dimensional number of F_1-F_{28} increasing from 30-$\mathcal{D}$ and 50-$\mathcal{D}$.

In summary, the proposed VGLBSO has the promising scalability performance.

F. INFLUENCES OF PROPOSED VGLBSO'S INDIVIDUAL COMPONENTS

To effectively evaluate the influences of RG, IC-VGL, and H-IU of VGLBSO, six VGLBSO variants, called VGLBSO-01, VGLBSO-02, VGLBSO-03, VGLBSO-04, VGLBSO-05, and VGLBSO-06 are developed. The VGLBSO and its six variants are ranked on the 28 CEC2013 functions F_1-F_{28}. Each algorithm is independently executed 30 times on each function. Considering 30-$\mathcal{D}$ problems, the population size and the maximum number of the iteration are set to 50 and 300000. Due to space limitation, the detailed experimental results including the values of Mean and Std on each function are given in Table s5 of the Section S-IV of the supplementary file. Table 9 only lists the average and final rank on each function.

From the first row of Table 9, we can observe that VGLBSO and its six variants all cover three components: IG, IC, and IU. However, each of the six variants has only one component that is different from any of the three components of VGLBSO. For instance, Table 9 shows that in terms of three components of VGLBSO01, only its IG using the K-means scheme is different from that of VGLBSO using the RG scheme. Note that the VGLBSO has better average and final rank than VGLBSO01, which suggests that introducing the RG scheme into VGLBSO can effectively improve the overall performance of VGLBSO.

TABLE 9. Influences of Individuals' grouping, individuals' creation, and individuals' update in VGLBSO.

Algorithm	IG	IC	IU	Average Rank	Final Rank
VGLBSO01	K-means	IC-VGL	H-IU	2.50	2
VGLBSO02	RG	BSO	H-IU	4.78	6
VGLBSO03	RG	ADMBSO	H-IU	3.96	3
VGLBSO04	RG	GBSO	H-IU	4.17	5
VGLBSO05	RG	IC-VGL	BSO	5.68	7
VGLBSO06	RG	IC-VGL	CLBSO	4.14	4
VGLBSO	RG	IC-VGL	H-IU	**2.32**	**1**

Furthermore, Table 9 has shown that the only difference between VGLBSO, VGLBSO02, VGLBSO03, and VGLBSO04 lies in that they adopt the IC-VGL, the IC schemes of BSO, ADMBSO, and GBSO, respectively. Interestingly, VGLBSO has the best average and final rank among the four algorithms, indicating that IC-VGL has the better performance compared with each of the IC scheme of BSO, ADMBSO, and GBSO.

In addition, we can notice from Table 9 that the only difference between VGLBSO, VGLBSO05, and VGLBSO06 is that the three algorithms use the IU scheme of the original BSO, CLBSO, and H-IU, respectively. Particularly, VGLBSO remains the best average and final rank among the three algorithms, denoting that H-IU has the better performance compared with the IU scheme of the original BSO or CLBSO.

G. PARAMETER SELECTION OF VGLBSO

VGLBSO contains five newly introduced parameters, namely $\mathcal{P}_l$, $\mathcal{P}_h$, $\mathcal{M}$, λ, and Δ. Note that $\mathcal{P}_l$ and $\mathcal{P}_h$ are used to regulate the dynamic range of $\mathcal{P}_{\mathcal{R}0}$ in formula (14) and compromise the global exploration and local exploitation capability during the entire iteration process for IC-VGL scheme; $\mathcal{M}$ is the number of groups in the RG scheme; λ is a scale factor that is used to regulate the proportion of all individuals using the two different individual update schemes in formula (15); Δ denotes at least the dimensional size of each sub-vector when the total dimensions of each individual are stochastically

TABLE 10. Comparisons among different settings of p_l and p_h with $\mathcal{M} = 10$, $\lambda = 0.1$, and $\Delta = 3$, and being unchanged.

Evaluation Criteria	p_l=0.1, p_h=0.8	p_l=0.2, p_h=0.7	p_l=0.3, p_h=0.6	p_l=0.4, p_h=0.5
Average Rank	2.39	2.57	**2.25**	2.46
Final Rank	2	4	**1**	3

TABLE 11. Comparisons among different settings of $\mathcal{M} = 10$ with $p_l = 0.3$, $p_h = 0.6$, $\lambda = 0.1$, and $\Delta = 3$ being unchanged.

Evaluation Criteria	$\mathcal{M}$=2	$\mathcal{M}$=5	$\mathcal{M}$=10
Average Rank	2.29	2.00	**1.57**
Final Rank	3	2	**1**

TABLE 12. Comparisons among different settings of λ with $p_l = 0.3$, $p_h = 0.6$, $\mathcal{M} = 10$, and $\Delta = 3$ being unchanged.

Evaluation Criteria	λ=0.1	λ=0.2	λ=0.3	λ=0.4	λ=0.5
Average Rank	**1.57**	2.29	2.86	3.71	4.5
Final Rank	**1**	2	3	4	5

separated into a set of sub-vectors in the stochastic vector grouping mechanism.

The different values of p_l, p_h, $\mathcal{M}$, λ, or Δ might trigger different effects on the performance of VGLBSO. Functions $\mathcal{F}_1$-$\mathcal{F}_{28}$ with the problem dimension of $\mathcal{D} = 30$ are used to determine the suitable values of these parameters. VGLBSO is operated 30 times on each function with different parameter values. The population size, MaxFEs and the maximum number of iterations are 50, 300000 and 6000, respectively. Due to space limitation, Tables s6, s7, s8, and s9 of Section S-VI of the supplementary file give the mean error and standard deviation values on $\mathcal{F}_1$-$\mathcal{F}_{28}$ for the different parameters.

First, when $\lambda = 0.1$, $\Delta = 3$, and $\mathcal{M} = 10$ are fixed, we execute four different sets of pairwise values of p_l and p_h inlcuding [$p_l = 0.1$, $p_h = 0.8$], [$p_l = 0.2$, $p_h = 0.7$], [$p_l = 0.3$, $p_h = 0.6$], and [$p_l = 0.4$, $p_h = 0.5$]. Table 10 exhibits that $p_l = 0.3$ and $=0.6$ wins best average and final rank for VGLBSO.

Second, considering that parameters $\lambda = 0.1$, $\Delta = 3 p_l = 0.3$, $p_h = 0.6$ are fixed, a set of different values of $\mathcal{M} = 2$, 5, and 10 is performed. Table 11 lists that $\mathcal{M} = 10$ wins the best average and final rank for VGLBSO.

Third, with $\Delta = 3$, $p_l = 0.3$, $p_h = 0.6$, and $\mathcal{M} = 10$ being fixed and a set of different values of $\lambda = 0.1$, 0.2, 0.3, 0.4, and 0.5, the results involving the overall performance of VGLBSO is listed in Table 12. Note that $\lambda = 0.1$ has provided the best average and final rank for VGLBSO.

Finally, given the values of $\Delta = 1$, 2, 3, 6, 9, 12, and 15 with $p_l = 0.3$, $p_h = 0.6$, $\mathcal{M} = 10$, and $\lambda = 0.1$ being invariant. Table 13 presents that $\Delta = 1$ can offer the best average and final rank for VGLBSO on $\mathcal{F}$1-$\mathcal{F}$28 among the different values of Δ.

TABLE 13. Comparisons among different settings of Δ with $p_l = 0.3$, $p_h = 0.6$, $\mathcal{M} = 10$, and $\lambda = 0.1$ being unchanged.

Evaluation Criteria	Δ=1	Δ=2	Δ=3	Δ=6	Δ=9	Δ=12	Δ=15
Average Rank	**2.89**	3.96	3.32	3.82	3.79	3.93	5.00
Final Rank	**1**	6	2	4	3	5	7

Generally, two sets of parameters [$p_l = 0.3$, $p_h = 0.6$, $\mathcal{M} = 10$, $\Delta = 1$, $\lambda = 0.1$] and [$p_l = 0.3$, $p_h = 0.6$, $\mathcal{M} = 10$, $\Delta = 3$, $\lambda = 0.1$] can give the first and second best overall performance for VGLBSO among different values of p_l, p_h, $\mathcal{M}$, λ and Δ, so they are recommended in the paper.

H. DISCUSSION ON RESULTS

The above experimental results indicate that VGLBSO achieves the best overall performance including the global search ability, convergence speed, and scalability amongst all the compared algorithms, which is attributed to the effective combination of the IC-VGL, H-IU, and the RG scheme. Their features and advantages are detailed below.

Firstly, in most of existing BSO algorithms, their IC schemes mainly adopted either the information exchange between individuals like those of BSO and ADMBSO, or the information exchange based on the individuals' dimensions such as that of GBSO. The former ignored the information interchange based on the individuals' dimensions and might decline the global exploration capability. However, the latter might cause numerous meaningless explorations and decline the local exploitation capability. The IC-VGL makes a rational compromise between the two individuals' creation schemes mentioned above. Specifically, the IC-VGL consists of stochastic vector grouping mechanism, vector grouping learning pattern A, and vector grouping learning pattern B. First, the stochastic vector grouping mechanism stochastically divides the full dimensions of each individual of the entire swarm into a set of sub-vectors, which can generate the sufficient sub-vectors and supply the rational diversity of information. Then, two vector grouping learning patterns A and B are employed to create a new individual for each individual; the pattern A highlights the vector grouping learning between different individuals from one group, so it can play a chief role in the local exploitation; however, the pattern B focuses on the vector grouping learning between different individuals from two different groups to the entire swarm so that it can play a crucial role in the global exploration. Finally, the probability selection mechanism in formula (14) can provide a dynamic adaptive selection between patterns A and B in the entire iteration process; to be more specific, in the early iteration process, the probability selection mechanism enables the pattern B to have more opportunities to create new individuals, which focuses more on the global exploration and discovers more promising solution regions; however, in the later iteration process, the probability selection mechanism makes the pattern A be more likely to create new individuals, emphasizing more on local exploitation and accelerating the convergence speed. By combing the stochastic vector grouping mechanism, two grouping learning patterns, and probability selection mechanism, IC-VGL not only avoids the numerous meaningless explorations and improves the global exploration capability, but also enhances the local exploitation capability. Therefore, IC-VGL can provide the rational balance between the global exploration and local exploitation capability for VGLBSO.

Secondly, most of the existing BSO algorithms adopted the IU scheme of either the original BSO or the differential evolution. Owing to using the logarithmic sigmoid function with a Gaussian random number as update step size, the former is unable to provide suitable local exploitation performance. On the other hand, the latter cannot offer sufficient global exploration capability when tackling the complicated global problems. Unlike the above two IU schemes, the H-IU has divided the newly created individuals of the entire swarm into two groups, and individuals in two groups are updated according to the IU scheme of the original BSO and the DE strategy, respectively. By combine such two update schemes, the H-IU can further improve the balance between the global exploration and local exploitation capability for VGLBSO.

Thirdly, we have introduced the RG scheme into VGLBSO to replace the K-means grouping scheme. This is due to the following two reasons. One reason is that the RG scheme has the low computational cost for VGLBSO due to such a fact that it does not compute the distance between two different individuals. The other more important reason is that the RG scheme can provide the diversity of the information exchange between different individuals for VGLBSO by allocating different individuals of the entire swarm into different groups.

Although VGLBSO contributes to the better overall performance on the above CEC2013 functions compared with the 12 BSO variants and nine swarm intelligence algorithms, it fails to provide the better result on each of all the functions. From the "No Free Lunch Theorems" [45], no single swarm intelligence algorithm is perfect for any optimization problem. In reality, for a bunch of publications involving swarm intelligence algorithms, it is exceedingly rare to find one algorithm that is superior to all other compared algorithms on each optimization problem of a well-established benchmark suit. In other words, for various sophisticated and efficient swarm intelligence algorithms, they can contribute their advantages to different optimization issues. For this reason, we will consider integrating various swarm intelligence algorithms such as ABC and CMA-ES variants into the BSO algorithms to create new BSO algorithm and further improve the global exploration and local exploitation capability in the future. In addition, the VGLBSO will be applied to multi-objective optimization issues from mass-spring model in virtual surgery.

V. CONCLUSION

The original BSO failed to effectively compromise the global exploration and local exploitation capability so that it suffered from the premature convergence for tackling various complicated optimization problems. To address this issue, we have developed a new VGLBSO including three components: the RG, IC-VGL, and H-IU scheme. To validate the performance of VGLBSO, we have executed comparisons between VGLBSO, 12 BSO variants, and nine swarm intelligence algorithms on the 28 CEC2013 benchmark functions. The experimental results suggest that the VGLBSO obtains the best the global search ability, convergence speed, and scalability amongst all the compared algorithms. Subsequently, we also evaluate the effects of the RG, IC-VGL, and H-IU scheme on VGLBSO based on the 28 functions. The results have confirmed the validity of such three components on VGLBSO. Finally, we have regulated the appropriate values of five newly introduced parameters in VGLBSO by performing experiments on the 28 functions. In summary, VGLBSO has provided the rational compromise between the global exploration and local exploitation capability via the effective combination of the IC-VGL, H-IU, and the RG scheme.

REFERENCES

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

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

[4] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Apr. 2007.

[5] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Beckington, U.K.: Luniver Press, 2008.

[6] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (Studies in Computational Intelligence), vol. 284. Berlin, Germany: Springer, 2010, pp. 65–74.

[7] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012.

[8] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994.

[9] R. G. Reynolds, "An introduction to cultural algorithms," in *Proc. 3rd Annu. Conf. Evol. Program.*, Singapore, 1994, pp. 131–139.

[10] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[11] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, Jun. 2001.

[12] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence* (Lecture Notes in Computer Science), vol. 6145. Berlin, Germany: Springer, 2010, pp. 355–364.

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

[14] H. Zhu and Y. Shi, "Brain storm optimization algorithm for full area coverage of wireless sensor networks," in *Proc. 8th Int. Conf. Adv. Comput. Intell.*, Chiang Mai, Thailand, Feb. 2016, pp. 14–20.

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

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

[17] S. Cheng, J. Chen, X. Lei, and Y. Shi, "Locating multiple optima via brain storm optimization algorithms," *IEEE Access*, vol. 6, pp. 17039–17049, 2018.

[18] Y. Shi, J. Xue, and Y. Wu, "Multi-objective optimization based on brain storm optimization algorithm," *Int. J. Swarm Intell. Res.*, vol. 4, no. 3, pp. 1–21, 2013.

[19] Y. Sun, "A hybrid approach by integrating brain storm optimization algorithm with grey neural network for stock index forecasting," *Abstr. Appl. Anal.*, vol. 2014, May 2014, Art. no. 759862.

[20] 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.

[21] A. R. Jordehi, "Brainstorm optimisation algorithm (BSOA): An efficient algorithm for finding optimal location and setting of FACTS devices in electric power systems," *Int. J. Elect. Power Energy Syst.*, vol. 69, pp. 48–57, Jul. 2015.

[22] R. Roy and J. Anuradha, "A modified brainstorm optimization for clustering using hard c-means," in *Proc. IEEE Int. Conf. Res. Comput. Intell. Commun. Netw.*, 2016, pp. 202–207.

[23] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–35, 2013.

[24] 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, Apr. 2015.

[25] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, "An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation," *Appl. Intell.*, vol. 40, pp. 256–272, Mar. 2014.

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

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

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

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

[30] 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.*, vol. 2015, Jul. 2015, Art. no. 923698.

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

[32] H. Duan and C. Li, "Quantum-behaved brain storm optimization approach to solving Loney's solenoid problem," *IEEE Trans. Magn.*, vol. 51, no. 1, Jan. 2015, Art. no. 7000307.

[33] Z. Yang and Y. Shi, "Brain storm optimization with chaotic operation," in *Proc. 7th Int. Conf. Adv. Comput. Intell.*, Mar. 2015, pp. 111–115.

[34] Z. Song, J. Peng, C. Li, and P. X. Liu, "A simple brain storm optimization algorithm with a periodic quantum learning strategy," *IEEE Access*, vol. 6, pp. 19968–19983, 2018.

[35] H. Zhu and Y. Shi, "Brain storm optimization algorithms with k-medians clustering algorithms," in *Proc. 7th Int. Conf. Adv. Comput. Intell.*, 2015, pp. 107–110.

[36] 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.

[37] Z. Cao, X. Rong, and Z. Du, "An improved brain storm optimization with dynamic clustering strategy," in *Proc. 3th Int. Conf. Mech. Eng.*, 2017, pp. 1–6.

[38] 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. Swarm Comput. Intell.*, vol. 9140. New York, NY, USA: Springer-Verlag, 2015, pp. 373–381.

[39] Z. Cao, Y. Shi, X. Rong, B. Liu, Z. Du, and B. Yang, "Random grouping brain storm optimization algorithm with a new dynamically changing step size," in *Proc. Int. Conf. Adv. Swarm Intell.*, 2015, pp. 357–364.

[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," Dept. Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep., Jan. 2013.

[41] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

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

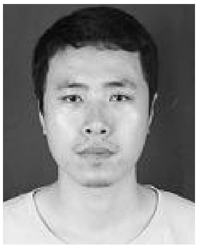
[43] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, Mar. 2008.

[44] 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.

[45] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[46] Y.-J. Gong *et al.*, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, Oct. 2016.

[47] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Inf. Sci.*, vol. 329, no. 1, pp. 329–345, 2016.



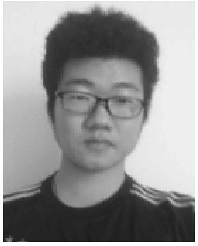
CHUNQUAN LI received the B.Sc., M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China, in 2002, 2007, and 2015, respectively.

He has been with the School of Information Engineering, Nanchang University, since 2002, where he is currently an Associate Professor and a Young Scholar of Ganjiang River. He has published over 30 research articles.

He is also currently a Visiting Professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. His current interests include computing intelligence, haptics, virtual surgery simulation, robotics, and their applications to biomedical engineering.

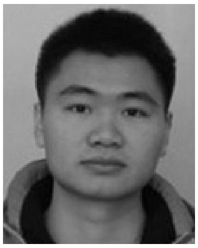
DUJUAN HU received the B.Sc. degree from the Nanchang Institute of Technology, Nanchang, China, in 2016. She is currently pursuing the M.Sc. degree in information and communication engineering with Nanchang University, Nanchang, Jiangxi.

Her current research interests include swarm intelligent algorithms and elgctromyographic signal processing.

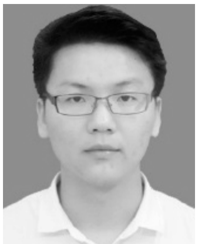


ZHENSHOU SONG is currently pursuing the degree with the School of Information Engineering, Nanchang University, Nanchang, Jiangxi, China.

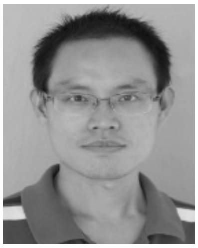
His current research interests include swarm particle swarm optimization, genetic algorithms, and other computational intelligence techniques.



FENG YANG received the B.Sc. degree from Nanchang University, Nanchang, Jiangxi, China, in 2016, where he is currently pursuing the M.Sc. degree in information and communication engineering. His current research interests include swarm intelligent algorithms, haptics, virtual surgical simulation, swarm particle swarm optimization, genetic algorithms, and other computational intelligence techniques.



ZU LUO received the B.Sc. degree from Nanchang University, Nanchang, Jiangxi, China, in 2017, where he is currently pursuing the M.Sc. degree in information and communication engineering. His current research interests include swarm intelligent algorithms, haptics, virtual surgical simulation, swarm particle swarm optimization, genetic algorithms, and other computational intelligence techniques.



JINGHUI FAN received the B.Sc. and M.Sc. degrees from Nanchang University, Nanchang, China, in 2001 and 2005, respectively.

He has been with the School of Information Engineering, Nanchang University, since 2002, where he is currently a Lecturer. He has published over 10 research articles. His research interests include swarm intelligent algorithms, haptics, and virtual surgical simulation.

PETER X. LIU (M'02–SM'07) received the B.Sc. and M.Sc. degrees from Northern Jiaotong University, Beijing, China, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Alberta, Edmonton, AB, Canada, in 2002.

He has been with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada, since 2002, where he is currently a Professor and the Canada Research Chair. He has published over 280 research articles. His current interests include computing intelligence, interactive networked systems and teleoperation, haptics, micro-manipulation, robotics, intelligent systems, context-aware intelligent networks, and their applications to biomedical engineering.

Dr. Liu was a recipient of the 2007 Carleton Research Achievement Award, the 2006 Province of Ontario Early Researcher Award, the 2006 Carty Research Fellowship, the Best Conference Paper Award of the 2006 IEEE International Conference on Mechatronics and Automation, and the 2003 Province of Ontario Distinguished Researcher Award. He serves as an Associate Editor for several journals, including the IEEE Transactions on Cybernetics, the IEEE/ASME Transactions on Mechatronics, the IEEE Transactions on Automation Science and Engineering, and the IEEE Access. He is a Licensed Member of the Professional Engineers of Ontario and a fellow of the Engineering Institute of Canada.