

## Research Article

# Dimensional Learning Strategy-Based Grey Wolf Optimizer for Solving the Global Optimization Problem

Xinyang Liu , Yifan Wang , and Miaolei Zhou 

*Department of Control Science and Engineering, Jilin University, Changchun 130022, China*

Correspondence should be addressed to Miaolei Zhou; [zml@jlu.edu.cn](mailto:zml@jlu.edu.cn)

Received 12 August 2021; Revised 6 December 2021; Accepted 28 December 2021; Published 30 January 2022

Academic Editor: *Ciro Castiello*

Copyright © 2022 Xinyang Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Grey wolf optimizer (GWO) is an up-to-date nature-inspired optimization algorithm which has been used for solving many of the real-world applications since it was proposed. In the standard GWO, individuals are guided by the three dominant wolves alpha, beta, and delta in the leading hierarchy of the swarm. These three wolves provide their information about the potential locations of the global optimum in the search space. This learning mechanism is easy to implement. However, when the three wolves are in conflicting directions, an individual may not obtain better knowledge to update its position. To improve the utilization of the population knowledge, in this paper, we proposed a grey wolf optimizer based on the dimensional learning strategy (DLGWO). In the DLGWO, the three dominant wolves construct an exemplar wolf through the dimensional learning strategy (DLS) to guide the grey wolves in the swarm. Thereafter, to reinforce the exploration ability of the algorithm, the Levy flight is also utilized in the proposed method. 23 classic benchmark functions and engineering problems are used to test the effectiveness of the proposed method against the standard GWO, variants of the GWO, and other metaheuristic algorithms. The experimental results show that the proposed DLGWO has good performance in solving the global optimization problems.

## 1. Introduction

In the real world, there are many optimization problems to be solved, and thus, the development of optimization techniques is of great importance. Most of these techniques rely on the derivatives of the functions involved in the problem, but for some reasons, derivatives of the functions are sometimes hard to obtain. As an important part of derivative-free methods, nature-inspired algorithms are attracting more and more attention for their robust searching performance. Generally, nature-inspired algorithms can be classified into three categories: evolutionary algorithms (EAs), swarm intelligence (SI), and physics-based (PB) algorithms [1–5].

SI-based algorithms, as one category of nature-inspired algorithms, are becoming increasingly popular due to their various characteristics such as strong searching ability, simple implementation, few parameters, and the ability to avoid getting trapped at local optima [6]. In these algorithms, the search agents cooperate and exchange

information with each other, which guarantees that the information about the search space is efficiently utilized. This helps the whole swarm to move towards more promising area in the search space. With the progress of the algorithm, there are roughly two patterns of behavior for the swarm to perform: the exploration and the exploitation. The exploration is the process of discovering new regions of the search space. The exploitation is the process of excavating the promising area in the search space for potentially better solutions. These two patterns are conflicting [7], and maintaining an appropriate balance between them is a great challenge for any SI-based algorithm. In the past few decades, numerous SI-based algorithms have been proposed in the literature and applied to real-life problems. Some of these algorithms are very representative such as particle swarm optimization (PSO) [8], ant colony optimization (ACO) [9], artificial bee colony (ABC) algorithm [10–14], cuckoo search (CS) [15, 16], and bat algorithm (BA) [17, 18]. Some of the recent SI are teaching-learning-based optimization (TLBO) [19], salp swarm algorithm (SSA) [20], sine cosine algorithm

(SCA) [21], whale optimization algorithm (WOA) [22], butterfly optimization algorithm (BOA) [23], and Harris hawks optimization (HHO) [24]. The grey wolf optimizer (GWO) was proposed by Mirjalili et al. [25] in 2014. The search mechanism of the GWO is established by mimicking the rigorous leadership hierarchy of a grey wolf pack and their group behavior when hunting. This mechanism guarantees a reliable exploration ability in the algorithm. Owing to its various advantages such as easy implementation, low computational complexity, and high search efficiency, the GWO has become quite notable and has been successfully applied to many fields in the past seven years [26, 27]. Despite the fact that the applications of the GWO technique in many fields have proved to be successful, previous works have shown that the GWO has some defects such as premature convergence and underutilization of population information and is prone to getting trapped at local optima, which are also common to some other SI-based algorithms [28, 29]. Many studies on the GWO method have been presented to address these shortcomings, which can be roughly categorized into four classes:

- (1) Modification of the algorithm parameters: in [30], a dynamically varying parameter  $A$  was adopted to maintain the proportion of wolves staying in the feasible area. In [31], Taguchi method was utilized to tune the parameters of the GWO. Rodríguez et al. [32] adopted the fuzzy inference system to dynamically adapt parameters  $A$  and  $C$ . In [33], a chaotic map was incorporated into the standard GWO to control parameters in the algorithm. Moreover, Kumar [34] provided an enhanced variant where parameter  $C$  was dynamically updating with the number of iterations. These modifications can improve the performance of the GWO on some problems, but the global exploration ability and local exploitation ability of the standard GWO are directly determined and balanced by the parameters  $A$  and  $C$ ; thus, the modification of these parameters can break the balance between exploration and exploitation.
- (2) Employ novel search strategies: in [35], an enhanced GWO variant was designed based on a new wolf position update mechanism. This significantly improved the local exploitation ability of the GWO, though it might suffer from premature stagnation when dealing with multimodal problems. Cai et al. [36] introduced the random local search strategy into the standard GWO to reinforce exploration. Yang et al. [37] employed the backward learning strategy to heighten the exploration in the GWO. Despite the contributions of this work, the proposed algorithm had difficulty in solving problems with more than three objectives, and its computational consumption was higher than the standard GWO. In [38], crossover and mutation operators were incorporated into the standard GWO to improve the diversity of the population, but these operators can prevent the proposed method from converging fast. Gupta and Deep [39] modified the original GWO algorithm

based on random walk. Although the proposed method cannot exhibit outstanding ability in local exploitation, the global exploration ability of the leader wolves was enhanced. In [40], the authors employed different selection methods and investigated the effects of them on the GWO. Liu et al. [41] proposed a modified GWO variant based on multiple search strategies such as the adaptive chaotic mutation strategy, boundary mutation strategy, and elitism strategy. The modified method can enhance the performance of the GWO for solving many-objective optimization problems, but it was only equipped with one type of constraint-handling technique, which might be a limitation in dealing with other practical engineering problems. In [42], the enhanced global-best lead strategy, the adaptable cooperative strategy, and the disperse foraging strategy were embedded into the GWO. Dhargupta et al. [43] combined opposition-based learning with the GWO to guarantee its exploration performance and convergence rate. Although the improvement of population diversity in the proposed algorithm is significant, it is challenging for the initial population to converge fast at the beginning of the iterations.

- (3) Combine with other metaheuristics: Wen et al. [44] hybridized the GWO with cuckoo search algorithm. Although there was extra computational consumption, the hybrid algorithm obtained solutions with higher quality. In [45], the GWO was combined with min-conflict algorithm. The hybrid method shows a good search performance, which can be further improved by adjusting the selection method of min-conflict algorithm. Qu et al. [46] simplified the GWO and combined it with symbiotic organisms' search. The combination of these two approaches accelerated the convergence speed of the algorithm, but individuals in the proposed method updated their positions only according to  $\alpha$ , instead of the best three wolves. This put the population under the risk of trapping in the local optimum.
- (4) Adjustment of the hierarchy or population structure: in [47], two additional leader wolves were introduced to the leadership hierarchy to heighten the exploration ability of the standard GWO. Miao et al. [48] proposed a GWO variant based on an enhanced leading hierarchy. Yang et al. [49] divided the wolf pack into two independent groups. Fehmi Burcin Ozsoydan [50] investigated the effects of dominant wolves and made modifications to the GWO based on the variations of dominant wolves. Wang et al. [51] provided a GWO variant based on chaotic initialization. Despite the effectiveness of these adjustments, the changes in the hierarchy structure or population structure can affect the information flow mechanism of the GWO. For instance, a new leadership hierarchy might change the update process of the population, and the utilization of the multipopulation strategy might limit the utilization

and exchange of information within the subpopulations.

According to various works mentioned above, the main common purpose of these studies is to boost the convergence speed and accuracy of the GWO while maintaining population diversity to keep an appropriate balance between exploration and exploitation. However, the population diversity is strongly affected by the information flow mechanism of the algorithm. In the standard GWO, the search process is guided by the three dominant wolves. This leads the population to converge towards these wolves. Therefore, the information of these wolves is essential for the population, and it would reduce the search efficiency of the algorithm when the information of these wolves is conflicting.

To improve the information utilization of the population in GWO algorithm, in this paper, the dimensional learning strategy (DLS) is implemented in the standard GWO. DLS was proposed by Xu et al. [52] to protect the potential useful information of the population best solution in PSO. In the DLS, a learning exemplar is constructed for each particle. During the constructing process, each dimension of a particle's personal best solution learns from the corresponding dimension of the population best solution. Therefore, the learning exemplar achieves the combination between the excellent information of the personal best experience and of the population best experience. Inspired by the DLS in PSO, we introduce the DLS into the leading hierarchy of GWO algorithm. There are some similarities and differences between our work and the DLS for PSO. The main purpose of employing the DLS in GWO is to efficiently protect the potential useful knowledge of the three dominant wolves, which is similar to the protection of the population best solution in PSO. The implementation of the DLS in GWO is by constructing an exemplar wolf within the three dominant wolves, which is similar to the construction of a learning exemplar between a particle's personal best solution and the population best solution in PSO. However, DLS in PSO constructs a learning exemplar for each particle based on its personal best solution and the population best solution in the current iteration. To reduce computational cost, we only construct one exemplar wolf at each iteration to guide the whole population. On each dimension, the delta wolf learns from the corresponding dimensions of both alpha and beta wolves to determine the corresponding dimension of the exemplar wolf. By employing the DLS, the potential useful knowledge of the three dominant wolves is well protected, and the situation that their information on some dimensions is conflicting is effectively avoided, which allows the GWO to utilize the knowledge of the three dominant wolves comprehensively and efficiently. In addition, to maintain an appropriate balance between exploration and exploitation, the Levy flight [15] is also employed in the algorithm. Levy flights generate steps that are drawn from the Levy distribution. Levy distribution has a good chance to randomly generate long-distance movement, which can introduce random figures to the population and improve the diversity of the population. Based on the above

modifications, an enhanced grey wolf optimizer based on the dimensional learning strategy, namely, DLGWO, is proposed.

The remainder of this paper is organized as follows: the related work is introduced in Section 2. In Section 3, the proposed method is presented and illustrated in detail. Several groups of experiments to test the proposed DLGWO are conducted and reported in Section 4. In Section 5, the applications of the DLGWO to address real-world optimization problems are given. Finally, we provide the conclusion of this study in Section 6.

## 2. Related Work

*2.1. GWO.* GWO is a metaheuristic optimization algorithm which is inspired from social hierarchy and hunting behavior of grey wolves. Figure 1 shows the hierarchical structure in a grey wolf pack and the hunting behavior of the grey wolves in the 2D search space.

To mathematically present the social hierarchy of a wolf pack, the fittest three solutions of the population are considered as alpha ( $\alpha$ ), beta ( $\beta$ ), and delta ( $\delta$ ), respectively. Other solutions in the population are considered as omega ( $\omega$ ).

The encircling behavior is mathematically modelled using the following equations:

$$X(t+1) = X_{\text{prey}}(t) - A \cdot B, \quad (1)$$

$$B = |C \cdot X_{\text{prey}}(t) - X(t)|, \quad (2)$$

$$A = 2a \cdot r_1 - a, \quad (3)$$

$$C = 2 \cdot r_2, \quad (4)$$

$$a = 2 - 2 \cdot \left(\frac{t}{T}\right), \quad (5)$$

where  $X$  and  $X_{\text{prey}}$  are the position vectors of the wolf and the prey, respectively.  $A$  and  $C$  are coefficient vectors, and  $r_1$  and  $r_2$  are random vectors in  $[0, 1]$ .  $t$  and  $T$  are the current iteration and the max iteration, respectively. The value of  $a$  is linearly decreasing from 2 to 0.

Assume that alpha, beta, and delta have better knowledge about the prey's position, and the other wolves update their positions according to the positions of alpha, beta, and delta. To mathematically express the hunting behavior, the equations are as follows:

$$X_1 = |X_\alpha - A_1 \cdot B_\alpha|, \quad X_2 = |X_\beta - A_2 \cdot B_\beta|, \quad (6)$$

$$X_3 = |X_\delta - A_3 \cdot B_\delta|,$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}, \quad (7)$$

where  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$  are the best three solutions.  $A_1$ ,  $A_2$ , and  $A_3$  are determined by equation (3).  $B_\alpha$ ,  $B_\beta$ , and  $B_\delta$  are calculated using the following equations:

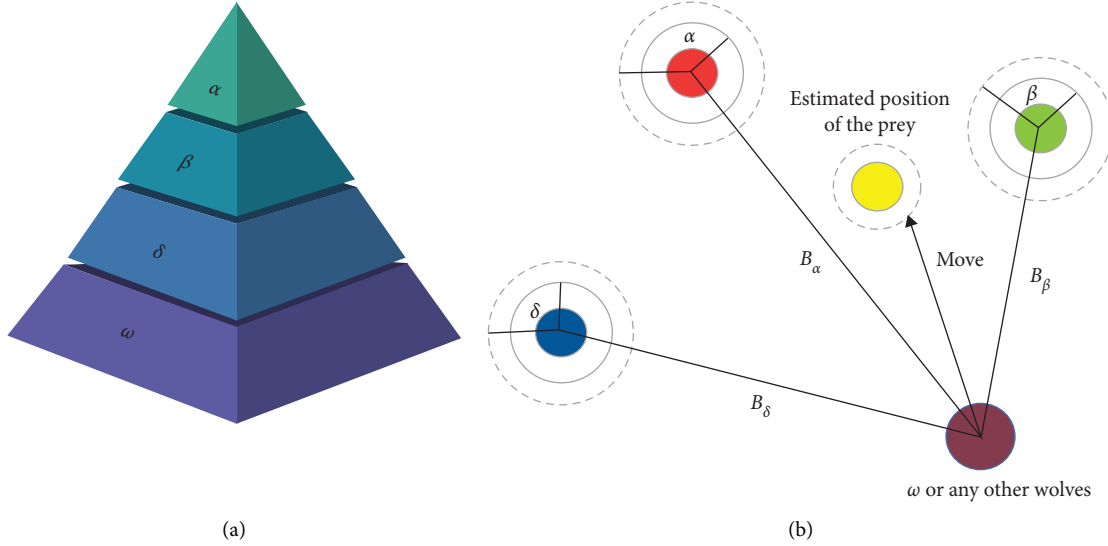


FIGURE 1: Hunting behavior of grey wolves.

$$B_\alpha = |C_1 \cdot X_\alpha - X|, B_\beta = |C_2 \cdot X_\beta - X|, B_\delta = |C_3 \cdot X_\delta - X|, \quad (8)$$

where  $C_1$ ,  $C_2$ , and  $C_3$  are random vectors. In the GWO, the search behavior of the grey wolves is determined by parameter  $A$ . Notice that  $A$  is generated randomly in the interval  $[-a, a]$ .  $|A| > 1$  leads the search agents to expand their searching areas, and  $|A| < 1$  leads the search agents to converge to the areas that have already been explored. The pseudo-code of the standard GWO is presented in Algorithm 1.

**2.2. PSO.** PSO [8] mimics the swarm behavior, and the particles in the swarm represent search agents in the search space. Each particle updates its velocity and position by learning from its personal best position and the population best position in the search space, and the update equations are as follows:

$$v_{i,j} = \omega v_{i,j} + c_1 r_{1,j} (x_{i,j}^{pbest} - x_{i,j}) + c_2 r_{2,j} (x_j^{gbest} - x_{i,j}), \quad (9)$$

$$x_{i,j} = x_{i,j} + v_{i,j},$$

where  $v_{i,j}$  and  $x_{i,j}$  are the velocity and position of the  $j$ th dimension of particle  $i$ , respectively.  $x_{i,j}^{pbest}$  is the historical best position of particle  $i$ , and  $x_j^{gbest}$  is the population historical best position.  $c_1$  and  $c_2$  are the acceleration coefficients, and  $r_{1,j}$  and  $r_{2,j}$  are two uniformly distributed random numbers independently generated in  $[0, 1]$  for the  $j$ th dimension. The inertial weight  $\omega$  is used to control the velocity, which decreases linearly from 0.9 to 0.4 over iterations.

**2.3. DLS for PSO.** Xu et al. [52] proposed the dimensional learning strategy (DLS) to protect the potential helpful information of the particles in PSO. In the standard PSO, the particles learn from their personal best experience and

population best experience. This learning strategy can lead to the phenomena of “oscillation.” [53] When the personal best position  $x_i^{pbest}$  and the population best position  $x_j^{gbest}$  locate in two opposite directions of the current position  $x_i$ , after particle  $i$  moves towards  $x_j^{gbest}$ , it will move closer to  $x_i^{pbest}$  at the next iteration since the difference  $x_i^{pbest} - x_i$  is larger than the difference  $x_j^{gbest} - x_i$ . A particle will constantly wander between the personal best position and the population best position, which can cause “oscillation” and limit the search efficiency of PSO. DLS is different from the learning strategy of the standard PSO. In the DLS, the personal best position  $x_i^{pbest}$  learns from the population best position  $x_j^{gbest}$  dimension by dimension to construct a learning exemplar  $x_i^{dl}$  which allows the excellent information of  $x_j^{gbest}$  to be inherited by the exemplar  $x_i^{dl}$ , improving the information utilization of  $x_j^{gbest}$ . By replacing  $x_i^{pbest}$  by  $x_i^{dl}$  in equation (9) of the standard PSO, the improved velocity update equation is as follows:

$$v_{i,j} = \omega v_{i,j} + c_1 r_{1,j} (x_{i,j}^{dl} - x_{i,j}) + c_2 r_{2,j} (x_j^{gbest} - x_{i,j}). \quad (10)$$

The process of constructing an exemplar  $x_i^{dl}$  is shown in Algorithm 2.

### 3. The Proposed Method

**3.1. Motivation of the Work.** In the standard GWO, other wolves update their positions according to the three dominant wolves in the leading hierarchy. These three wolves provide their information about the potential location of the prey, and their positions are closer to the global optimum or local optimum in the search space. However, when alpha, beta, and delta are located in conflicting directions, an individual may not obtain better knowledge about the promising area, as demonstrated in Figure 2. Thus, it is of great importance to protect the potential helpful knowledge of the leading hierarchy and improve the information utilization of the population.

- (1) Initialize the positions of the population  $X_i (i = 1, 2, \dots, n)$  and the parameters of the GWO (max iteration  $Max_{iter}$ , population size  $N$ ,  $A$ , and  $C$ )
- (2) Calculate the fitness of each wolf and record the best three wolves
- (3)  $X_\alpha$  = position of the best wolf
- (4)  $X_\beta$  = position of the second best wolf
- (5)  $X_\delta$  = position of the third best wolf
- (6) **while**  $t < Max_{iter}$  **do**
- (7)   **for** each wolf **do**
- (8)     Update the position of the current wolf using equations (6)–(8)
- (9)   **end for**
- (10)   Update parameters  $A$  and  $C$
- (11)   Check the boundaries and revise those wolves beyond the boundaries
- (12)   Calculate the fitness of all wolves
- (13)   Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$
- (14)    $t = t + 1$
- (15) **end while**
- (16) Return  $X_\alpha$

ALGORITHM 1: The pseudo-code of the standard GWO.

- (1)  $x_i^{dl} = x_i^{pbest}$
- (2) **for** each dimension  $j$  **do**
- (3)   temp =  $x_i^{dl}$
- (4)   **if** temp( $j$ ) ==  $x^{gest}(j)$  **then**
- (5)     Continue
- (6)   **end if**
- (7)   temp( $j$ ) =  $x^{gest}(j)$
- (8)   **if**  $f(temp) < x^{gest}(j)$  **then**
- (9)      $x_i^{dl}(j) = x^{gest}(j)$
- (10)   **end if**
- (11) **end for**

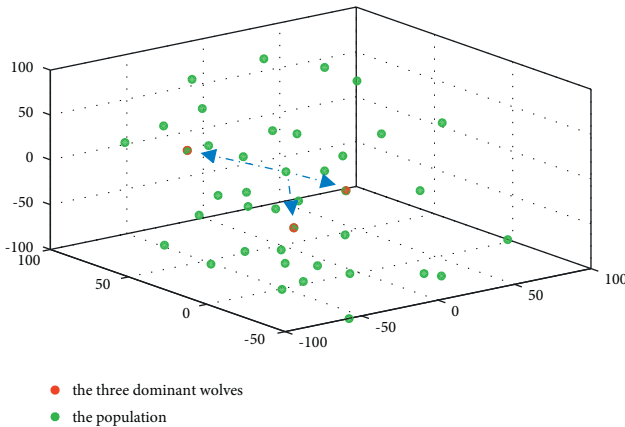
ALGORITHM 2: Constructing the exemplar  $x_i^{dl}$ .

FIGURE 2: Conflicting guidance of the three dominant wolves.

#### 4. DLS for the GWO

Inspired by the DLS in PSO [52], in this paper, DLS is employed in the leading hierarchy of the GWO to protect the potential helpful knowledge about the prey's location and guide

the individuals with more efficiency. In the DLS, delta learns from both alpha and beta dimension by dimension to construct an exemplar wolf; in this way, the excellent information of the three wolves can be passed to the exemplar wolf and other wolves in the pack. Figure 3 illustrates the process of the DLS; suppose that 4D sphere function  $f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2$  is the objective function of the minimum problem, which has the global minimum point  $(0, 0, 0, 0)^T$ .  $X_\alpha = (1, 2, 2, 2)^T$ ,  $X_\beta = (2, 4, 1, 3)^T$ , and  $X_\delta = (3, 0, 3, 4)^T$  are the positions of alpha, beta, and delta, respectively. The numbers filled in red, blue, green, and yellow are the values of the three dominant wolves and the exemplar wolf at the first, second, third, and fourth dimension, respectively. Initially, let the position vector of the exemplar wolf  $X_L = X_\delta = (3, 0, 3, 4)^T$ ; it is easy to calculate that  $f(X_L) = f(X_\delta) = 34$ ; then, two temporary vectors  $X^{\text{temp}1}$  and  $X^{\text{temp}2}$  are set for  $X_L$  at each dimension during the process of the DLS which can be expressed as follows:

- (1) For dimension 1: let  $X_1^{\text{temp}1} = X_{\alpha,1} = 1$ ,  $X_1^{\text{temp}2} = X_{\beta,1} = 2$ ,  $X^{\text{temp}1} = (1, 0, 3, 4)^T$ , and  $X^{\text{temp}2} = (2, 0, 3, 4)^T$ .

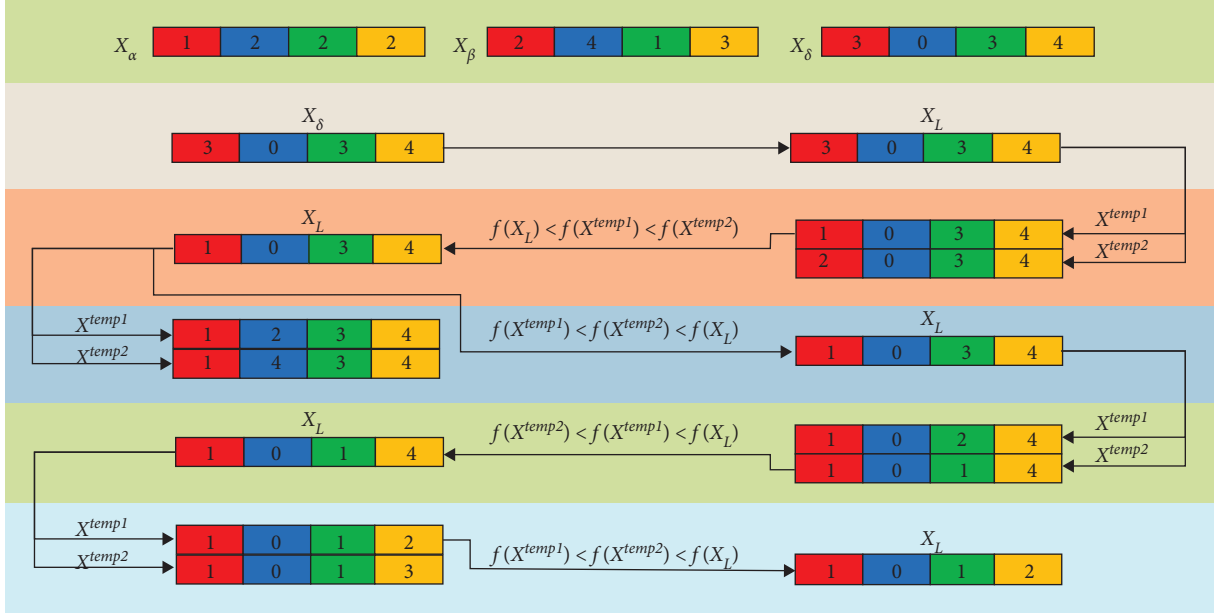


FIGURE 3: Process of the DLS.

- $f(X^{\text{temp}1}) = 26 \langle f(X^{\text{temp}2}) = 29 \langle f(X_L) = 34$ ; thus,  $X_{L,1} = X_1^{\text{temp}1} = 1$ ,  $X_L = (1, 0, 3, 4)^T$ , and  $f(X_L) = 26$ .
- (2) For dimension 2: let  $X_2^{\text{temp}1} = X_{\alpha,2} = 2$ ,  $X_2^{\text{temp}2} = X_{\beta,2} = 4$ ,  $X^{\text{temp}1} = (1, 2, 3, 4)^T$ , and  $X^{\text{temp}2} = (1, 4, 3, 4)^T$ .  
 $f(X_L) = 26 \langle f(X^{\text{temp}1}) = 30 \langle f(X^{\text{temp}2}) = 42$ ; thus,  $X_{L,2}$  remains unchanged,  $X_L = (1, 0, 3, 4)^T$ , and  $f(X_L) = 26$ .
- (3) For dimension 3: let  $X_3^{\text{temp}1} = X_{\alpha,3} = 2$ ,  $X_3^{\text{temp}2} = X_{\beta,3} = 1$ ,  $X^{\text{temp}1} = (1, 0, 2, 4)^T$ , and  $X^{\text{temp}2} = (1, 0, 1, 4)^T$ .  
 $f(X^{\text{temp}2}) = 18 \langle f(X^{\text{temp}1}) = 21 \langle f(X_L) = 26$ ; thus,  $X_{L,3} = X_3^{\text{temp}2} = 1$ ,  $X_L = (1, 0, 1, 4)^T$ , and  $f(X_L) = 18$ .
- (4) For dimension 4: let  $X_4^{\text{temp}1} = X_{\alpha,4} = 2$ ,  $X_4^{\text{temp}2} = X_{\beta,4} = 3$ ,  $X^{\text{temp}1} = (1, 0, 1, 2)^T$ , and  $X^{\text{temp}2} = (1, 0, 1, 3)^T$ .  
 $f(X^{\text{temp}1}) = 6 \langle f(X^{\text{temp}2}) = 11 \langle f(X_L) = 18$ ; thus,  $X_{L,4} = X_4^{\text{temp}1} = 2$ ,  $X_L = (1, 0, 1, 2)^T$ , and  $f(X_L) = 6$ .

After the DLS,  $X_L = (1, 0, 1, 2)^T$ , and it learns from alpha at the first and the fourth dimensions, beta at the third dimension, and delta at the second dimension. Hence, the final  $X_L$  is constructed by combining  $X_\delta$  with the dimensions learned from  $X_\alpha$  and  $X_\beta$ , which indicates that the exemplar wolf is not worse than the three dominant wolves. To implement the DLS in the GWO, we substitute the best three wolves in the standard GWO with the exemplar wolf to update the positions of other wolves. The improved equation is given by

$$X(t+1) = X_L - A_L \cdot B_L, \quad B_L = |C_L \cdot X_L - X|, \quad (11)$$

where  $X$  is the position vector of a wolf.  $X_L$  is the position vector of the exemplar wolf.  $B_L$  is an intermediate variable.

$A_L$  and  $C_L$  can be obtained by equations (3) and (4), respectively. As illustrated by Figure 3, before the value of the exemplar wolf at the current dimension is updated,  $X_L$  is compared with the two temporary vectors  $X^{\text{temp}1}$  and  $X^{\text{temp}2}$ .  $X^{\text{temp}1}$  and  $X^{\text{temp}2}$  are set by substituting the value of  $X_L$  with the value of  $X_\alpha$  and  $X_\beta$  at the current dimension, respectively. If  $X^{\text{temp}1}$  or  $X^{\text{temp}2}$  is better than the exemplar wolf, the value of the exemplar wolf at the current dimension is updated. Otherwise, the exemplar wolf remains unchanged and continues the process of DLS at the next dimension. Therefore, the exemplar wolf learns only from the dimensions of the three dominant wolves that can help improve its fitness value, which guarantees that the exemplar wolf will not be degraded when alpha, beta, and delta are located in conflicting directions and hence improves the utilization of the population knowledge. The flowchart of the DLS is illustrated in Figure 4.

**4.1. Trial Solutions Based on Levy Flight.** For a SI-based algorithm, exploration and exploitation are performed simultaneously. Exploration is to discover more promising areas in the search space, and exploitation is to focus on the current optimal areas. Hence, it is important to keep an appropriate balance between exploration and exploitation. In the DLS, each grey wolf learns from the exemplar wolf; this leads the population to converge to the exemplar wolf which strengthens the exploitation ability of the algorithm and potentially causes premature convergence as well. To improve the exploration performance of the algorithm, the method proposed by Mantegna [54] to generate the Levy flight is utilized in the algorithm. In comparison to the Gaussian distribution which always generates small steps, the Levy distribution can occasionally generate long steps which is helpful for the exploration. The 2D and 3D

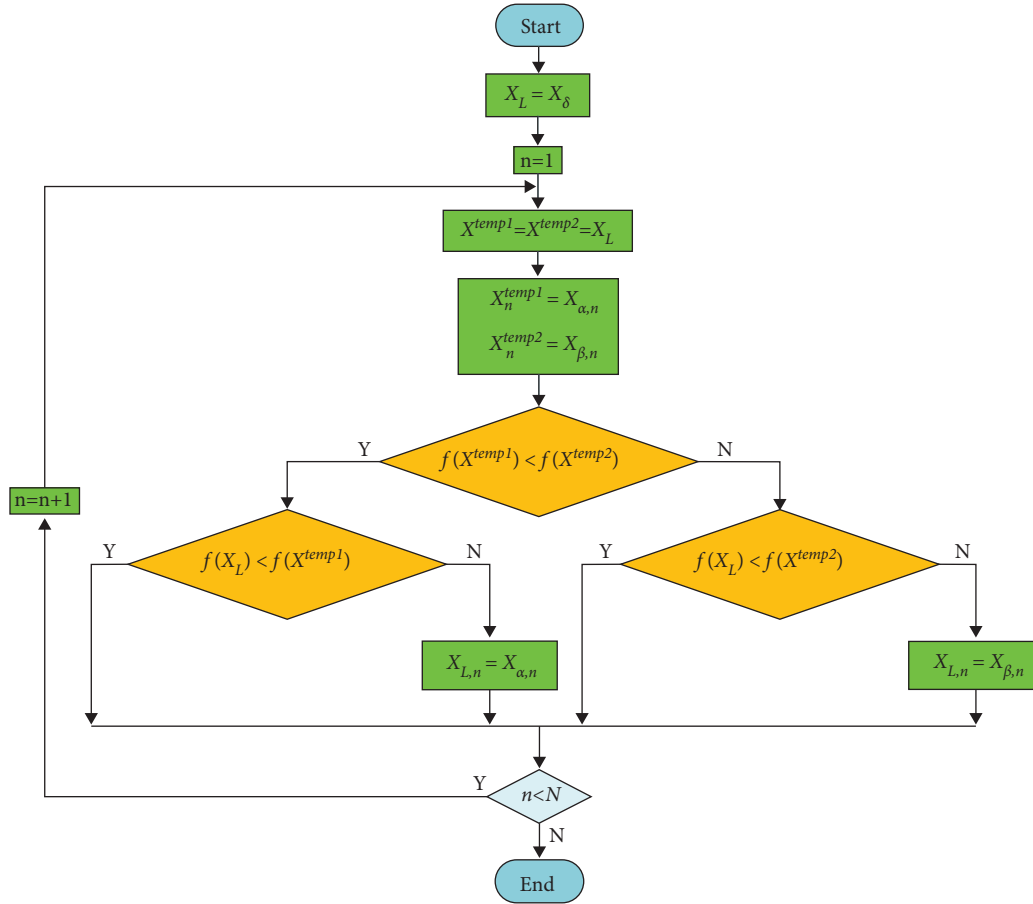


FIGURE 4: DLS flowchart.

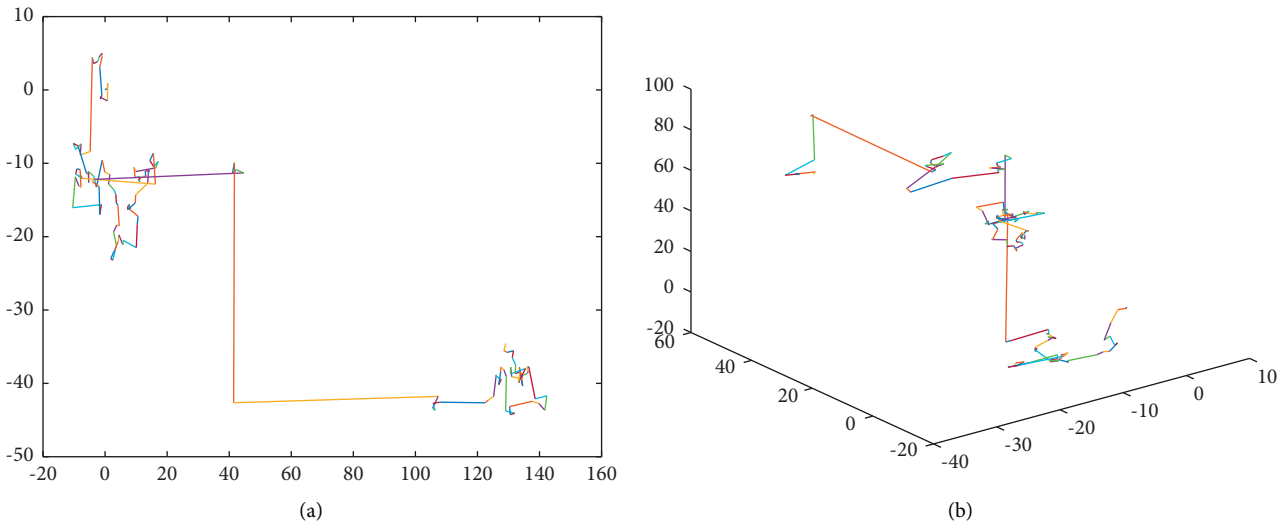


FIGURE 5: 2D and 3D trajectories drawn from the Levy distribution.

trajectories drawn from the Levy distribution are illustrated in Figure 5.

After a wolf updates its position according to the exemplar wolf by equation (12), a trial solution is obtained by Levy flight using the following formula:

$$X_{\text{trial}}(t + 1) = X(t + 1) + G \times \text{Levy}(D), \quad (12)$$

where  $D$  is the dimension size,  $G$  is a randomly generated vector with the size of  $1 \times D$ , and Levy is the Levy flight function, which is given by



$$\text{Levy} = 0.01 \times \frac{u \times \phi}{|v|^{1/z}}, \quad (13)$$

where  $u$  and  $v$  are random values in the interval  $[0,1]$  and  $z$  is a constant set to 1.5.  $\phi$  can be expressed as

$$\phi = \left\{ \frac{\Gamma(1+z) \times \sin(\pi z/2)}{\Gamma((1+z)/2) \times z \times 2^{(z-1)/2}} \right\}^{1/z}. \quad (14)$$

Then, the objective function value of the trial position  $f(X_{\text{trial}}(t+1))$  is compared with that of the updated position  $f(X(t+1))$ , and the position with the smaller value is preserved, which can be expressed by

$$X(t+1) = \begin{cases} X(t+1) & \text{if } f(X(t+1)) \leq f(X_{\text{trial}}(t+1)), \\ X_{\text{trial}}(t+1) & \text{if } f(X(t+1)) > f(X_{\text{trial}}(t+1)). \end{cases} \quad (15)$$

**4.2. The Proposed DLGWO Algorithm.** By employing the DLS and Levy flight, we proposed an enhanced variant of the GWO, named as dimension learning grey wolf optimizer (DLGWO). The framework of the proposed DLGWO is illustrated in Figure 6. Firstly, DLS is utilized to protect the potential useful knowledge of the three dominant wolves; this enhances search efficiency and reinforces the exploitation ability simultaneously. Secondly, Levy flight is embedded into the algorithm as an effective measure to guarantee population diversity and strengthen the exploration ability. The process of constructing the exemplar wolf is shown in Algorithm 3.

**4.3. Analysis of Computational Complexity.** The computational costs of the standard GWO include initialization  $O(nD)$ , evaluation at each iteration  $O(n)$ , selecting the dominant wolves  $O(n)$ , and position updating  $O(nD)$ ;  $n$  and  $D$  are the population and dimension size, respectively. For the DLGWO, the additional costs occur when the dimensional learning process is utilized in which the position vector of the exemplar wolf is constructed by comparing  $f(X_L)$  with the smaller value of  $f(X^{\text{temp1}})$  and  $f(X^{\text{temp2}})$ , and since the position vector of the exemplar wolf has been updated and recorded during the process, the employing of the DLS requires additional computation of  $2D$  fitness evaluations  $O(2D)$  in each iteration. From the above analysis, the computational complexity for the standard GWO and DLGWO is at the same level.

## 5. Experimental Verification and Analysis

In this section, 4 experiments are carried out and presented. First, the efficacy of the DLS and Levy flight in the proposed algorithm is verified. Then, we perform experiments to verify the capacity of the DLGWO on functions of different dimensions. After that, the proposed DLGWO is compared with the GWO and its 6 promising variants. Finally, DLGWO is examined with other well-established meta-heuristics. A widely utilized set of benchmark functions is employed [28, 30, 40, 48, 55], and the detailed information about the benchmark functions can be found in Table 1. As

presented in Table 1, the test set includes 7 unimodal functions ( $f1-f7$ ), 6 multimodal functions ( $f8-f13$ ), and 10 fixed-dimension multimodal functions ( $f14-f23$ ). There is only one global optimal solution in unimodal functions; thus, these functions are suitable for evaluating the local exploitation capability of an algorithm. With respect to the multimodal functions, they have several local optimal solutions besides the global optimal solution and thus are utilized to challenge the global exploration ability and the capacity of avoiding the local optimum. The fixed-dimension multimodal functions are composed of global optimum, local optimum, and many different characteristics such as rotation and shift and are used to examine the ability of an algorithm when disposing of complicated cases. Figure 7 demonstrates the landscapes of three benchmark functions. The indexes for comparing include the mean values (Mean), the standard deviations (SD), and rank (Rank) of the average best result for each method. Besides, to check if the improvements of the DLGWO over the other algorithms are significant, the Wilcoxon rank-sum tests at a 0.05 significance level are also utilized. The Wilcoxon rank-sum test is a paired test that checks for significant differences between two algorithms, where “+ /  $\approx$  / -” means that the proposed algorithm is significantly better, similar to, or significantly worse than the comparison algorithm. In addition, all the experiments are implemented using MATLAB R2016a and are run on a CPU Core i5-4210U, 4GB RAM, with Windows 10 operating system.

**5.1. Effect of Different Strategies.** Before comprehensive evaluations, it is essential to investigate the effect of each strategy on the performance of the proposed algorithm. As mentioned above, the DLGWO algorithm consists of two main improvement strategies: DLS and Levy flight. In this part, the effectiveness of these two strategies is validated. For this purpose, a comparison between the DLGWO and its variants is conducted. Herein, its variants are denoted as the DLSGWO and LFGWO, respectively. The algorithm adopting the DLS while the Levy flight is ignored is denoted as DLSGWO. The algorithm employing the Levy flight while the DLS is not utilized is denoted as LFGWO.

To more intuitively measure the exploration and exploitation abilities of the DLGWO and its variants, we compare the search history, trajectories of the first search agent at the first dimension, and population diversity of the DLGWO, DLSGWO, LFGWO, and the standard GWO. The population diversity is calculated using the following equations:

$$\text{diversity} = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{j=1}^D (x_{i,j} - \bar{x}_j)^2}, \quad \bar{x}_j = \frac{\sum_{i=1}^N x_{i,j}}{N}, \quad (16)$$

where  $N$  is the population size,  $D$  is the dimension of the search space,  $x_{i,j}$  denotes the  $j$ th dimension of the  $i$ th particle, and  $\bar{x}_j$  denotes the  $j$ th dimension of the center position of the population.

The obtained results are exhibited in Figure 8. These results record the search history, trajectories of the first search agent at its first dimension, and diversity of solutions



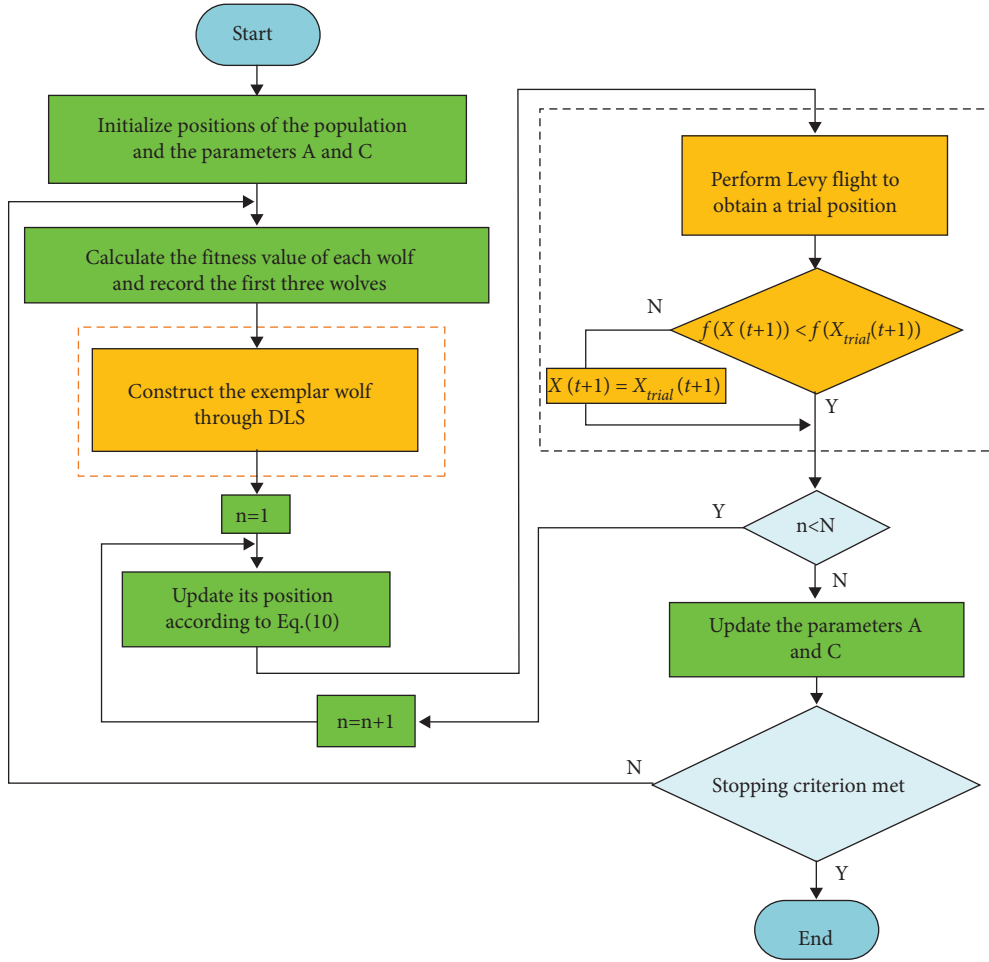


FIGURE 6: DLGWO method.

- (1)  $X_L = X_\delta$
- (2) **for** each dimension  $j$  **do**
- (3)  $X_j^{\text{temp1}} = X_{\alpha,j}$
- (4)  $X_j^{\text{temp2}} = X_{\beta,j}$
- (5) Calculate  $\min\{f(X^{\text{temp1}}), f(X^{\text{temp2}}), f(X_L)\}$  and record the corresponding vector
- (6) Replace  $X_{L,j}$  with the corresponding vector, and if  $f(X_L) = \min\{f(X^{\text{temp1}}), f(X^{\text{temp2}}), f(X_L)\}$ , then  $X_{L,j}$  remains unchanged
- (7) **end for**

ALGORITHM 3: Constructing the exemplar wolf according to the three dominant wolves.

based on 20 search agents with the maximum number of function evaluations of 30000 in dealing with two unimodal functions ( $f_1$  and  $f_2$ ) and two multimodal functions ( $f_{10}$  and  $f_{11}$ ) from Table 1. The search history and trajectories are displayed in two dimensions, more clearly. These plots are recorded in the same way demonstrated in the original work of the GWO.

From the search history and trajectory plots in Figure 8, it can be seen that the DLGWO and LFGWO exhibit a better distributed scatter plot in the initial stage of iteration compared with the DLSGWO. These two methods have more extensive coverage on those unexplored areas in the

search space that has not been covered efficiently by the DLSGWO. Those blue and yellow points away from the center of the search space are generated mostly based on Levy flight. Then, after initial iterations, search agents are guided towards areas with more high-quality solutions (in these four functions, the global optimum is at the center of the search space). In this process, the exemplar wolf constructed in the DLSGWO carries the information about potential areas with excellent solutions and leads the population to the center and its surrounding areas. Therefore, more search agents in the population are guided by the DLSGWO than other methods towards the vicinity of the

TABLE 1: Benchmark functions.

Function name	Expression	Dim	Range	$f_{\min}$
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	30	$[-100, 100]^D$	0
Schwefel's problem 2.22	$f_2(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	30	$[-10, 10]^D$	0
Schwefel's problem 1.2	$f_3(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]^D$	0
Schwefel's problem 2.21	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	30	$[-100, 100]^D$	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	$[-30, 30]^D$	0
Step	$f_6(x) = \sum_{i=1}^D ( x_i + 0.5 )^2$	30	$[-100, 100]^D$	0
Noisy quartic	$f_7(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1]$	30	$[-1.28, 1.28]^D$	0
Schwefel's problem 2.26	$f_8(x) = -\sum_{i=1}^D (x_i \sin(\sqrt{ x_i }))$	30	$[-100, 100]^D$	-12.569.5
Rastrigin	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^D$	0
Ackley	$f_{10}(x) = -20 \exp(-0.2 \sqrt{1/D \sum_{i=1}^D x_i^2}) - \exp(1/D \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]^D$	0
Griewank	$f_{11}(x) = 1/4000 \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$	30	$[-600, 600]^D$	0
Penalized 1	$f_{12}(x) = \pi/D \left\{ \sum_{i=1}^{D-1} [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 + (10 \sin^2(\pi y_1)) \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4) y_i = 1 + x_i + 1/4$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	$[-50, 50]^D$	0
Penalized 2	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	30	$[-50, 50]^D$	0
Shekel's Foxholes function	$f_{14}(x) = (1/500 + \sum_{j=1}^{25} 1/j + \sum_{i=1}^{25} (x_i - a_{i,j})^6)^{-1}$	2	$[-65, 65]^D$	0.998
Kowalik's function	$f_{15}(x) = \sum_{i=1}^{11} [a_i - ((x_1(b_i^2 + b_i x_2)) / (b_i^2 + b_i x_3 + x_4))]^2$	4	$[-5, 5]^D$	0.003
Six-hump camel back	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{5}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^D$	-1.0316
Branin	$f_{17}(x) = (x_2 - 5.1/4\pi^2 x_1^2 + 5/\pi x_1 - 6)^2 + 10(1 - 1/8\pi) \cos x_1 + 10$	2	$[-5, 5]^D$	0.398
Goldstein-Price function	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 14x_2^2 - 36x_1 x_2 + 27x_2^2)]$	2	$[-2, 2]^D$	3.00
Hartmann 1	$f_{19}(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2]$	3	$[1, 3]^D$	-3.86
Hartmann 2	$f_{20}(x) = -\sum_{i=1}^5 c_i \exp[-\sum_{j=1}^4 a_{ij} (x_j - p_{ij})^2]$	6	$[0, 1]^D$	-3.32
Shekel 1	$f_{21}(x) = -\sum_{i=1}^7 [c_i \exp[-\sum_{j=1}^T (X - a_{ij})(X - a_{ij})^T + c_i]]^{-1}$	4	$[-10, 10]^D$	-10.1532
Shekel 2	$f_{22}(x) = -\sum_{i=1}^{10} [c_i \exp[-\sum_{j=1}^T (X - a_{ij})(X - a_{ij})^T + c_i]]^{-1}$	4	$[-10, 10]^D$	-10.4028
Shekel 3	$f_{23}(x) = -\sum_{i=1}^{10} [c_i \exp[-\sum_{j=1}^T (X - a_{ij})(X - a_{ij})^T + c_i]]^{-1}$	4	$[-10, 10]^D$	-10.5363

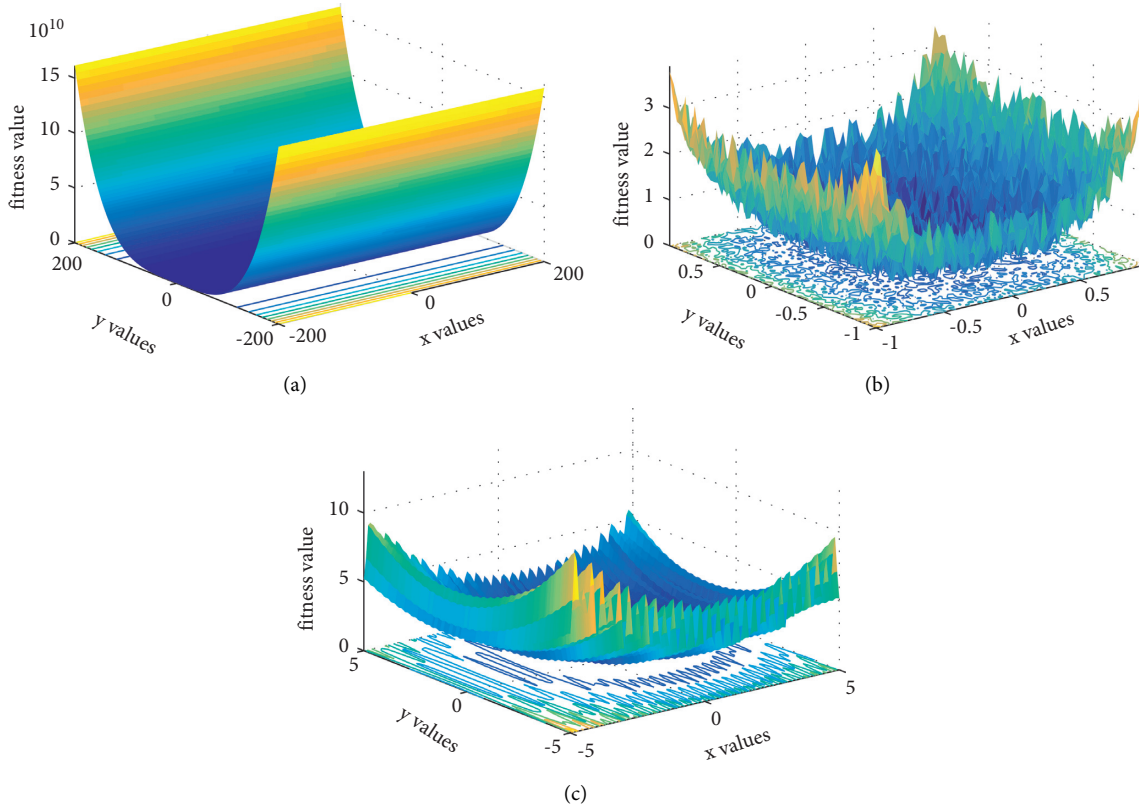


FIGURE 7: Demonstration of benchmark functions.

center. This verifies the exploitation ability of the DLSGWO. By contrast, LFGWO cannot guarantee strong exploitation in further iterations; it shows a lower concentration of solutions around the optimum due to the randomness from Levy flight. Furthermore, DLGWO is able to obtain a fine balance: it exhibits a strong exploration capacity in the initial iterations and a good exploitation capacity in further iterations. This is because of the incorporation of the two strategies. As for the diversity curves in Figure 8, LFGWO obtains the highest diversity, and DLGWO is ranked second. From the results of diversity, it can be observed that the DLSGWO maintains the smallest diversity and, consequently, has high convergence speed. As expected, the LFGWO maintains the highest diversity during the initial stage, and then Levy flight can significantly improve the exploration of the search space which provides a wide range of variety in solutions. Finally, the diversity of the DLGWO is lower than that of the LFGWO, but higher than that of the standard GWO and DLSGWO. This is because of the appropriate balance between exploitation and exploration provided by the interaction and cooperation of the DLS and Levy flight. Therefore, the diversity comparison results also verify our expectation that the DLS is mainly responsible for local exploitation while Levy flight for global exploration. According to the results and analysis above, the incorporation of the DLS and Levy flight can effectively improve the balance between the exploratory and exploitative performance of the DLGWO, while the DLSGWO and LFGWO with only one

of those strategies cannot obtain a better balance, individually.

To further investigate the utility and efficacy of the DLS and Levy flight, we compare the optimization results on  $f_1 - f_{13}$  of the DLGWO with both DLS and Levy flight, DLSGWO with the DLS alone, and LFGWO with Levy flight alone. The maximum number of fitness evaluations is set to 300000. Table 2 presents the statistical results of unimodal functions ( $f_1 - f_7$ ) and multimodal functions ( $f_8 - f_{13}$ ), respectively. From the numerical results shown in Table 2, DLSGWO ranks first, followed by DLGWO and LFGWO, for unimodal functions; LFGWO ranks first, followed by DLGWO and DLSGWO, for multimodal functions. It can be drawn from the experimental results that DLS guarantees, in each generation, the outstanding information of the three dominant wolves to be inherited by each wolf in the swarm through the exemplar wolf, which strongly enhances the convergence performance. However, the diversity of the DLSGWO is relatively low and is easily trapped in the local optimum; thus, it is challenging for the DLS to deal with multimodal functions. Levy flight improves the diversity of the population and exhibits better performance for solving multimodal functions with many local optima. Our design expectation is verified by the experimental results that the DLS concentrates on local exploitation, while Levy flight focuses on global exploration. It is the interaction and cooperation of DLS and Levy flight that enables the competitive performance of the DLGWO for both unimodal and multimodal functions.

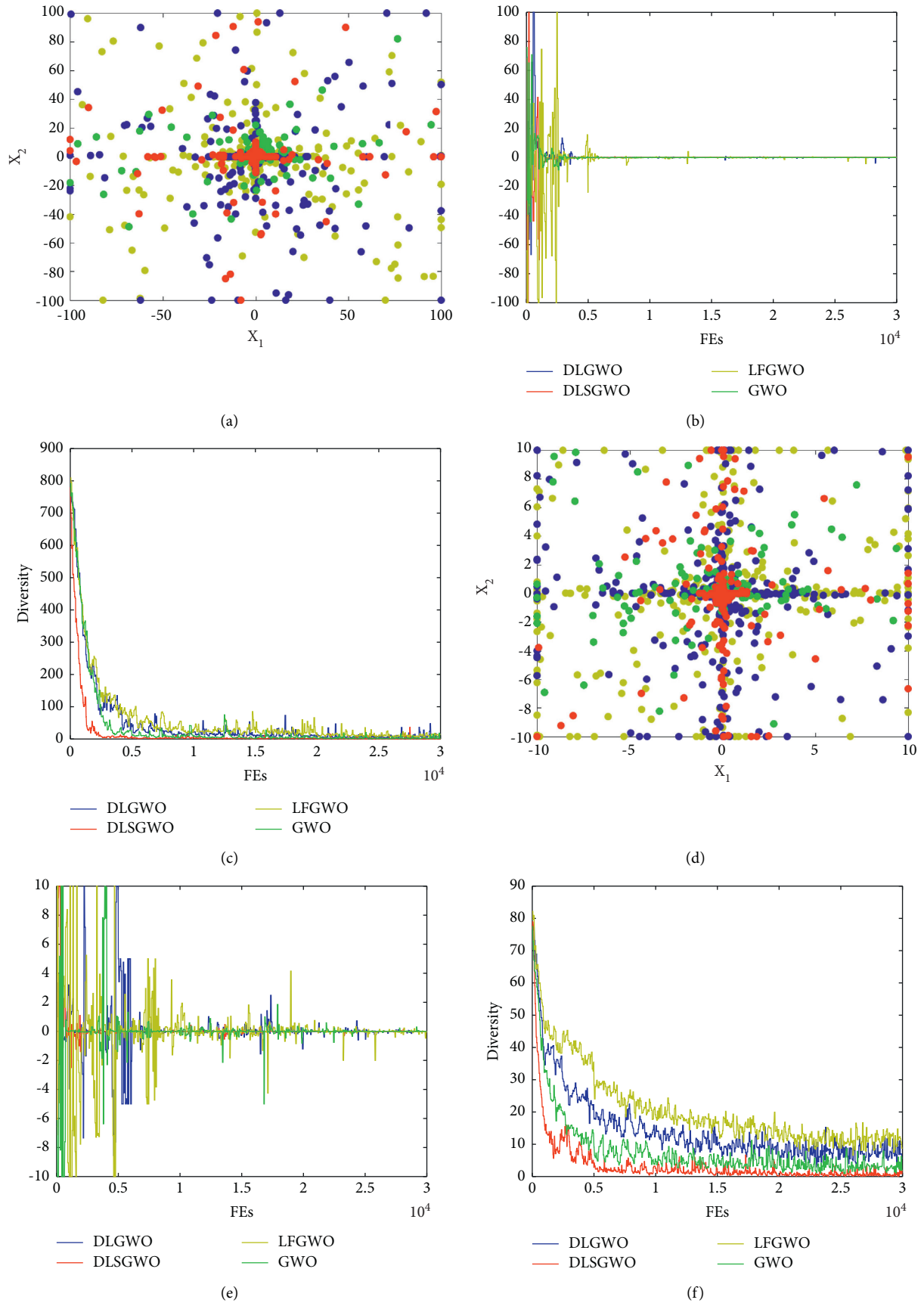


FIGURE 8: Continued.

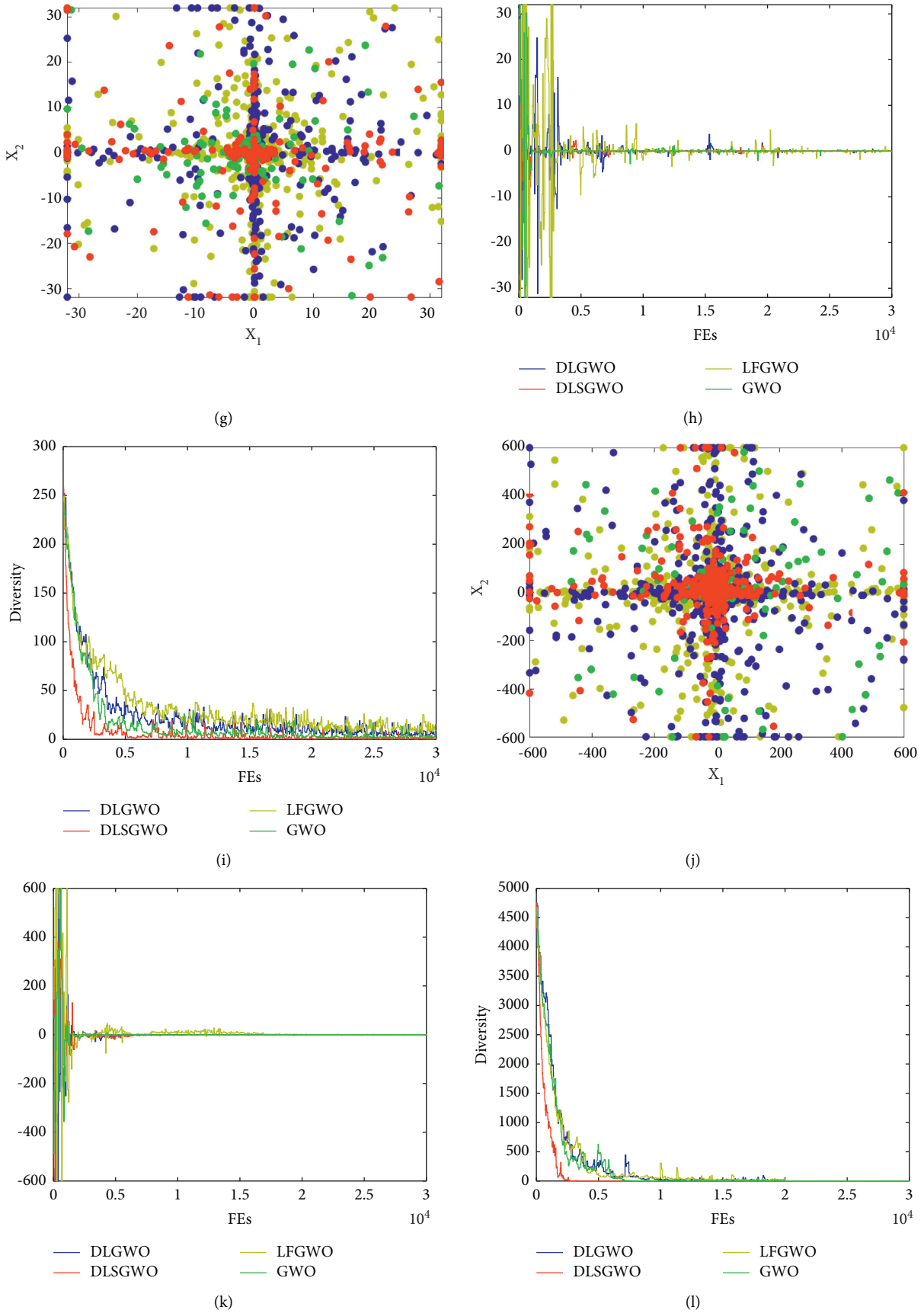


FIGURE 8: Comparison of search history, trajectories, and diversity between solutions of the DLGWO, DLSGWO, and LFGWO.

TABLE 2: Comparison results for the DLGWO, LFGWO, and DLSGWO.

	Metrics	DLSGWO	LFGWO	DLGWO
$f_1$	Mean	0.00E+00	9.25E-273	0.00E+00
	SD	0.00E+00	1.51E-272	0.00E+00
	Rank	1	3	1
$f_2$	Mean	0.00E+00	1.13E-282	0.00E+00
	SD	0.00E+00	8.46E-282	0.00E+00
	Rank	1	3	1
$f_3$	Mean	6.03E-192	8.19E-28	6.27E-182
	SD	7.68E-191	1.60E-27	1.01E-182
	Rank	1	3	2
$f_4$	Mean	7.11E-188	1.33E-115	9.07E-150
	SD	1.58E-189	1.06E-116	1.26E-151
	Rank	1	3	2
$f_5$	Mean	2.47E+01	2.71E+01	2.35E+01
	SD	1.23E+00	1.04E+00	3.08E-01
	Rank	2	3	1
$f_6$	Mean	1.24E-06	1.86E-01	1.08E-06
	SD	3.61E-07	1.06E-01	3.93E-07
	Rank	2	3	1
$f_7$	Mean	1.01E-05	6.12E-04	1.44E-05
	SD	8.51E-06	2.73E-04	5.18E-05
	Rank	1	3	2
Average rank for $f_1-f_7$		1.28	3.00	1.42
$f_8$	Mean	-7.21E+03	-7.92E+03	-8.33E+03
	SD	1.07E+03	5.46E+02	2.25E+02
	Rank	3	2	1
$f_9$	Mean	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1
$f_{10}$	Mean	8.97E-15	4.44E-15	4.95E-15
	SD	8.89E-15	1.58E-15	3.17E-15
	Rank	3	1	2
$f_{11}$	Mean	1.25E-03	0.00E+00	0.00E+00
	SD	2.81E-03	0.00E+00	0.00E+00
	Rank	3	1	1
$f_{12}$	Mean	6.54E-03	1.74E-08	1.68E-08
	SD	4.62E-03	5.73E-09	9.08E-09
	Rank	3	2	1
$f_{13}$	Mean	3.09E-05	9.63E-06	2.88E-07
	SD	1.31E-06	7.04E-07	1.82E-07
	Rank	3	2	1
Average rank for $f_8-f_{13}$		2.66	1.50	1.16
Average rank for $f_1-f_{13}$		1.92	2.30	1.31

5.2. *The Impact of Problem Dimension.* Any efficient optimizer should be able to make a fine tradeoff between minimizing cost and maximizing accuracy. From the analysis of computational complexity, the computational cost of utilizing DLS increases as the dimension of the problem increases. To better evaluate the impact of dimension on the performance of the proposed method, DLGWO and the standard GWO are implemented on  $f_1 - f_{13}$  from Table 1 with dimensions of 10, 30, 100, and 200. All conditions are the same, and the maximum number of fitness evaluations is 300000. By increasing the dimension, the statistical results are recorded and shown in Table 3. The boldface in Table 3 indicates the best experimental results. It can be seen from Table 3 that DLGWO outperforms the standard GWO in all dimensions with a promising and

stable performance. When dimension increases, the exemplar wolf has a more comprehensive knowledge of population information at different dimensions; thus, the exemplar wolf can guide the swarm with more efficiency. However, it can also be observed that, compared with the proposed method, the standard GWO can obtain optimization results with higher accuracy on  $f_3$ ,  $f_5$ , and  $f_{13}$  when the dimension increases to 200. The reason is that the extra fitness evaluations for DLGWO algorithm will also increase as the dimension rises, which can limit the performance of the DLGWO. Therefore, it is still an interesting and challenging future work to decrease the extra cost of constructing the exemplar wolf and obtain a better tradeoff between algorithm performance and computational cost.

TABLE 3: Comparison results on benchmark functions with different dimensions.

Dim	Metrics	10		30		100		200	
		GWO	DLGWO	GWO	DLGWO	GWO	DLGWO	GWO	DLGWO
$f_1$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	3.92E-262	<b>2.48E-286</b>	2.95E-129	<b>1.67E-162</b>
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.88E-263	0.00E+00	6.44E-128	9.88E-163
$f_2$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	2.23E-218	<b>8.43E-237</b>
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.05E-217	5.75E-237
$f_3$	Mean	5.69E-188	<b>3.86E-230</b>	1.05E-156	<b>6.10E-184</b>	5.61E-105	<b>1.52E-110</b>	<b>1.94E+00</b>	3.16E+00
	SD	2.76E-187	8.22E-231	9.01E-157	5.41E-184	9.65E-105	7.52E-111	8.08E-01	1.22E+E+00
$f_4$	Mean	1.09E-293	<b>1.33E-322</b>	5.45E-121	<b>9.36E-150</b>	3.16E-26	<b>2.79E-38</b>	3.91E-01	<b>1.99E-03</b>
	SD	2.24E-294	3.69E-323	6.56E-121	2.51E-151	2.35E-26	6.75E-38	1.52E-01	1.47E-03
$f_5$	Mean	6.22E+00	<b>2.76E+00</b>	2.77E+01	<b>2.36E+01</b>	9.61E+01	<b>9.41E+01</b>	<b>1.92E+02</b>	1.98E+02
	SD	2.62E-01	7.84E-01	9.34E-01	4.22E-01	3.16E-01	5.02E-01	2.38E-01	1.09E-01
$f_6$	Mean	7.06E-01	<b>2.53E-08</b>	5.96E-01	<b>4.55E-07</b>	1.49E+00	<b>2.50E-03</b>	2.74E+00	<b>7.29E-01</b>
	SD	5.26E-02	3.12E-08	3.45E-01	2.46E-07	7.94E-01	2.41E-04	1.74E+00	2.24E-01
$f_7$	Mean	9.55E-05	<b>2.27E-05</b>	4.28E-05	<b>1.09E-05</b>	8.26E-03	<b>6.04E-04</b>	1.44E-02	<b>2.56E-04</b>
	SD	5.79E-04	1.37E-05	3.07E-05	8.99E-06	3.22E-04	9.90E-05	8.88E-03	2.11E-04
$f_8$	Mean	-3.40E+03	<b>-4.35E+03</b>	-6.57E+03	<b>-8.29E+03</b>	-1.61E+04	<b>-2.02E+04</b>	-2.85E+04	<b>-3.48E+04</b>
	SD	2.34E+02	5.75E+01	5.25E+02	8.77E+02	4.31E+02	1.74E+02	5.54E+04	6.39E+02
$f_9$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	1.14E-13	<b>0.00E+00</b>
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.84E-14	0.00E+00
$f_{10}$	Mean	7.99E-15	<b>4.44E-15</b>	9.02E-15	<b>4.84E-15</b>	2.22E-14	<b>1.51E-14</b>	2.93E-14	<b>2.22E-14</b>
	SD	2.15E-15	1.17E-15	1.73E-15	1.34E-15	5.31E-15	3.33E-15	7.63E-14	4.75E-15
$f_{11}$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$f_{12}$	Mean	1.01E-01	<b>5.48E-08</b>	2.74E-02	<b>1.72E-08</b>	2.18E-01	<b>8.26E-04</b>	4.12E-01	<b>3.92E-03</b>
	SD	1.82E-02	1.04E-09	8.62E-03	4.69E-09	3.44E-01	1.36E-03	4.17E-01	8.10E-02
$f_{13}$	Mean	9.67E-01	<b>4.96E-08</b>	7.02E-01	<b>2.38E-07</b>	4.72E+00	<b>4.34E-01</b>	<b>1.51E+01</b>	2.06E+01
	SD	2.01E-01	5.90E-08	2.18E-01	5.91E-08	2.16E+01	9.59E-01	8.10E+00	6.55E+00

5.3. *Comparison of the DLGWO with the GWO and Its Variants.* In this section, the proposed DLGWO is compared with the GWO and its several promising variants. The explanations and parameter settings for these GWO variants are given in Table 4. The test functions utilized in this part are all classical benchmark problems from Table 1. Each problem is independently executed 30 times, where the number of search agents and the maximum number of fitness evaluations are 40 and 300000, respectively. The statistical results are provided in Tables 5–9.

5.4. *Algorithm Accuracy Analysis.* The numerical results on unimodal functions ( $f_1$ – $f_7$ ) are presented in Table 5. As for unimodal functions  $f_1$ – $f_7$ , these problems are suitable for testing the exploitation ability of algorithms. From the obtained results, DLGWO achieves the best rank for  $f_1$ ,  $f_2$ ,  $f_5$ , and  $f_7$ . On  $f_3$ , IGWO is the best optimizer with a competitive accuracy of  $8.42E-192$ , whereas DLGWO takes the second place with an accuracy of  $6.53E-182$ . On  $f_4$ , DLGWO is not as competitive as GWOCS, MGWO, and IGWO and comes in the fourth place. The global optimum of  $f_5$  is in a narrow area, which is quite challenging to obtain for most optimizers; thus, the results of the DLGWO and the other algorithms for  $f_5$  are satisfactory. On  $f_6$ , RWGWO exhibits slightly higher search accuracy compared with the DLGWO, but DLGWO also provides an outstanding accuracy of  $4.28E-07$ . In the final

average rank, DLGWO is the best optimizer with the best exploitation ability, followed by IGWO, GWOCS, MGWO, learnGWO, SOGWO, RWGWO, and GWO.

The numerical results on multimodal functions ( $f_8$ – $f_{13}$ ) are presented in Table 6. Contrast to unimodal functions, multimodal functions are suitable for examining the exploration ability since most of them contain numerous local optima, which may lead to premature convergence of optimizers. On  $f_8$ , it is difficult for the standard GWO to locate the global optimum. The reason is that this problem has many deep local optima far away from the global optima; once alpha, beta, and delta in the standard GWO are all trapped into a deep local optimum, they can hardly escape and may lead more search agents into the local optimum. From Table 6, DLGWO exhibits better ability of avoiding the local optimum and hence obtains a higher search accuracy of  $-8.35E+03$ . This can be ascribed to the Levy flight strategy in the DLGWO which improves the population diversity of the algorithm. All the methods can locate the global optimum for  $f_9$  and  $f_{11}$ . On  $f_{10}$ , IGWO obtains the highest accuracy of  $4.44E-15$ , followed by DLGWO with an accuracy of  $4.94E-15$ . On  $f_{11}$ , DLGWO has an apparent advantage compared with the other GWO variants, with an excellent accuracy of  $1.75E-08$ . On  $f_{13}$ , DLGWO takes the first place, and RWGWO also obtains a competitive accuracy of  $3.57E-07$ . On multimodal problems, DLGWO also exhibits the best performance, and RWGWO is the second



TABLE 4: Parameter settings and explanations for all versions of the GWO.

Name	Explanation	Parameters
GWO		$a = [2,0]$
RWGWO [39]	GWO with random walk	$a = [2,0]$
learnGWO [50]	GWO with improved hierarchy	$a = [2,0], \theta_\alpha = 0.004715, \theta_\beta = \theta_\delta = 0.00647$
GWOCS [44]	GWO hybridized with cuckoo search	$a = [2,0]$
IGWO [56]	GWO with individual memory	$a = [2,0], b_1 = 0.6, b_2 = 0.4, p = 0.5$
SOGWO [43]	GWO with opposition-based learning	$a = [2,0]$
MGWO [34]	GWO with modified parameter $C$	$a = [2,0]$

TABLE 5: Results on benchmark functions  $f_1-f_7$ .

	Metrics	GWO	RWGWO	learnGWO	GWOCS	IGWO	SOGWO	MGWO	DLGWO
$f_1$	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1	1	1	1	1	1
$f_2$	Mean	1.80E-284	5.40E-223	4.98E-291	0.00E+00	0.00E+00	1.33E-287	0.00E+00	0.00E+00
	SD	1.07E-284	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.90E-290	0.00E+00	0.00E+00
	Rank	7	8	5	1	1	6	1	1
$f_3$	Mean	6.82E-151	1.30E-153	1.80E-162	4.98E-180	8.42E-192	5.40E-153	2.10E-176	6.53E-182
	SD	2.32E-152	1.66E-153	6.06E-161	2.49E-180	7.62E-191	2.67E-153	5.27E-175	3.34E-182
	Rank	8	7	5	3	1	6	4	2
$f_4$	Mean	5.45E-121	7.53E-116	6.72E-126	7.23E-213	7.54E-155	1.83E-122	5.38E-190	9.36E-150
	SD	6.56E-121	2.86E-117	1.78E-127	0.00E+00	4.83E-155	8.25E-123	3.85E-191	2.51E-151
	Rank	7	8	5	1	3	6	2	4
$f_5$	Mean	2.73E+01	2.66E+01	2.63E+01	2.64E+01	2.61E+01	2.49E+01	2.77E+01	2.33E+01
	SD	9.47E-01	4.96E-01	3.61E-01	8.39E-01	7.79E-01	7.26E-02	7.53E-01	4.19E-01
	Rank	7	6	4	5	3	2	8	1
$f_6$	Mean	6.06E-01	1.88E-07	6.01E-01	1.21E+00	9.59E-02	9.98E-01	1.60E+00	4.28E-07
	SD	3.78E-01	3.88E-08	2.43E-01	3.63E-01	2.18E-02	9.64E-02	4.31E-02	1.06E-07
	Rank	5	1	4	7	3	6	8	2
$f_7$	Mean	5.18E-05	3.76E-04	5.16E-04	6.97E-05	1.93E-05	7.73E-05	4.22E-05	1.21E-05
	SD	2.68E-05	4.27E-05	1.39E-05	3.31E-05	9.21E-04	2.93E-05	2.05E-05	8.99E-05
	Rank	4	7	8	5	2	6	3	1
Average rank for $f_1-f_7$		5.57	5.42	4.57	3.28	2.00	4.71	3.85	1.71
Final rank for $f_1-f_7$		8	7	5	3	2	6	4	1

competitive method, followed by IGWO, learnGWO, SOGWO, GWOCS, MGWO, and GWO. As observed in the results of unimodal and multimodal functions, DLGWO obtains the best rank for 9 out of 13 problems and can achieve the global optima for 4 problems, i.e.,  $f_1, f_2, f_9$ , and  $f_{11}$ . It ranks first statistically among all the algorithms on both unimodal and multimodal functions, which demonstrates the efficiency of the DLS and Levy flight and the strength of their interaction and cooperation.

Table 7 presents the numerical results on the fixed-dimension multimodal functions ( $f_{14}-f_{23}$ ). These functions can evaluate the efficiency of the algorithms when dealing with functions with different characteristics such as rotation and shift. From the results provided in Table 7, DLGWO performs best on 4 out of 10 functions, i.e.,  $f_{15}, f_{16}, f_{18}$ , and  $f_{23}$ . On  $f_{14}, f_{17}, f_{19}, f_{21}$ , and  $f_{22}$ , DLGWO has the same search accuracy compared with the algorithms ranking first (RWGWO for  $f_{14}, f_{17}$ , and  $f_{21}$ , SOGWO for  $f_{19}$ , and MGWO for  $f_{22}$ ), but DLGWO is not as robust as these algorithms and has a higher standard deviation. On  $f_{20}$ ,

SOGWO obtains the global optimum, followed by DLGWO with a search accuracy of  $-3.28E+00$ . On  $f_{23}$ , DLGWO has the most robust performance with the smallest standard deviation of  $5.04E-07$ . In general, DLGWO shows competitive and robust performance on the fixed-dimensional functions, which proves that the two strategies implemented in the DLGWO show different effectiveness when dealing with functions with different features.

Tables 8 and 9 give the overall Wilcoxon rank-sum test results of the DLGWO and each algorithm. From Table 9, DLGWO outperforms the other comparative methods in different problems and obtains results of 20/3/0 vs. GWO, 14/6/3 vs. RWGWO, 19/4/0 vs. learnGWO, 18/4/1 vs. GWOCS, 14/6/3 vs. IGWO, 18/4/1 vs. SOGWO, and 16/5/2 vs. MGWO.

**5.5. Convergence Behavior Analysis.** To compare the convergence behavior, Figures 9 and 10 present the convergence curves of the DLGWO and other algorithms on  $f_1-f_{13}$ . It can be detected that the DLGWO has the fastest convergence speed on  $f_1$  and  $f_2$ , and it obtains the global optimum within 100000

TABLE 6: Results on benchmark functions  $f_8-f_{13}$ .

	Metrics	GWO	RWGW0	learnGWO	GW0CS	IGWO	S0GWO	MGWO	DLGWO
$f_8$	Mean	$-5.99E+03$	$-7.53E+03$	$-5.27E+03$	$-4.81E+03$	$-7.29E+03$	$-4.94E+03$	$-6.51E+03$	$-8.35E+03$
	SD	$1.16E+02$	$4.83E+02$	$1.25E+03$	$1.54E+03$	$4.27E+02$	$1.12E+03$	$5.33E+02$	$8.65E+02$
	Rank	5	2	6	8	3	7	4	1
$f_9$	Mean	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	SD	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	Rank	1	1	1	1	1	1	1	1
$f_{10}$	Mean	$8.97E-15$	$7.99E-15$	$7.99E-15$	$7.99E-15$	$4.44E-15$	$8.22E-15$	$8.05E-15$	$4.94E-15$
	SD	$1.73E-15$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$1.34E-15$	$4.42E-16$	$1.34E-15$
	Rank	8	3	3	3	1	7	6	2
$f_{11}$	Mean	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	SD	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	Rank	1	1	1	1	1	1	1	1
$f_{12}$	Mean	$2.76E-02$	$4.70E-05$	$4.57E-02$	$6.67E-02$	$3.46E-02$	$2.07E-02$	$1.12E-01$	$1.75E-08$
	SD	$8.68E-03$	$1.13E-04$	$2.35E-02$	$2.27E-02$	$1.29E-02$	$6.17E-09$	$5.33E-02$	$4.70E-09$
	Rank	5	2	3	6	8	4	7	1
$f_{13}$	Mean	$7.48E-01$	$3.57E-07$	$5.27E-01$	$6.66E-01$	$5.76E-01$	$3.08E-07$	$7.21E-01$	$2.42E-07$
	SD	$2.38E-01$	$5.29E-08$	$2.67E-01$	$1.99E-01$	$2.06E-01$	$6.80E-08$	$2.68E-01$	$6.31E-08$
	Rank	8	2	5	6	4	3	7	1
Average rank for $f_8-f_{13}$		4.67	1.83	3.17	4.17	3.00	3.83	4.33	1.17
Final rank for $f_8-f_{13}$		8	2	4	6	3	5	7	1

TABLE 7: Results on benchmark functions  $f_{14}-f_{23}$ .

	Metrics	GWO	RWGW0	learnGWO	GW0CS	IGWO	S0GWO	MGWO	DLGWO
$f_{14}$	Mean	$4.92E+00$	$9.98E-01$	$7.41E+00$	$4.35E+00$	$6.29E+00$	$9.98E-01$	$8.81E+00$	$9.98E-01$
	SD	$4.05E+00$	$8.27E-14$	$4.86E+00$	$4.46E+00$	$5.46E+00$	$3.83E-13$	$4.04E+00$	$3.83E-13$
	Rank	5	1	7	4	6	2	8	2
$f_{15}$	Mean	$6.03E-03$	$6.69E-04$	$3.17E-03$	$3.08E-04$	$3.43E-03$	$3.10E-04$	$4.38E-04$	$3.07E-04$
	SD	$9.78E-03$	$4.57E-04$	$7.58E-03$	$4.86E-07$	$7.47E-03$	$1.73E-08$	$3.46E-04$	$3.33E-10$
	Rank	8	5	6	2	7	3	4	1
$f_{16}$	Mean	$-1.03E+00$	$-1.03E+00$	$-1.03E+00$	$-1.03E+00$	$-1.03E+00$	$-1.03E+00$	$-1.03E+00$	$-1.03E+00$
	SD	$7.66E-11$	$4.59E-11$	$5.77E-11$	$1.79E-07$	$6.64E-10$	$7.49E-11$	$6.30E-11$	$1.03E-11$
	Rank	6	2	3	8	7	5	4	1
$f_{17}$	Mean	$3.97E-01$	$3.97E-01$	$3.97E-01$	$3.97E-01$	$3.97E-01$	$3.97E-01$	$3.97E-01$	$3.97E-01$
	SD	$2.24E-09$	$2.62E-10$	$7.88E-10$	$9.84E-06$	$3.77E-08$	$2.05E-09$	$4.86E-10$	$8.44E-10$
	Rank	6	1	3	8	7	5	2	4
$f_{18}$	Mean	$3.00E+00$	$3.00E+00$	$3.00E+00$	$3.00E+00$	$3.00E+00$	$3.00E+00$	$3.00E+00$	$3.00E+00$
	SD	$7.73E-07$	$9.59E-08$	$5.36E-08$	$1.74E-07$	$7.55E-08$	$2.09E-09$	$4.91E-08$	$5.68E-10$
	Rank	8	6	4	7	5	2	3	1
$f_{19}$	Mean	$-3.86E+00$	$-3.86E+00$	$-3.86E+00$	$-3.86E+00$	$-3.86E+00$	$-3.86E+00$	$-3.86E+00$	$-3.86E+00$
	SD	$2.49E-07$	$7.67E-05$	$1.94E-07$	$3.18E-07$	$3.04E+07$	$1.31E-08$	$1.32E-07$	$9.86E-08$
	Rank	5	8	4	7	6	1	3	2
$f_{20}$	Mean	$-3.24E+00$	$-3.27E+00$	$-3.24E+00$	$-3.27E+00$	$-3.26E+00$	$-3.32E+00$	$-3.18E+00$	$-3.28E+00$
	SD	$9.84E-02$	$6.46E-02$	$8.99E-02$	$5.81E-02$	$7.91E-02$	$2.41E-08$	$1.73E-01$	$5.80E-02$
	Rank	7	4	6	3	5	1	8	2
$f_{21}$	Mean	$-8.72E+00$	$-1.01E+01$	$-8.69E+00$	$-6.91E+00$	$-9.42E+00$	$-1.01E+01$	$-9.42E+00$	$-1.01E+01$
	SD	$2.43E-02$	$3.81E-07$	$2.48E+00$	$2.34E+00$	$1.92E+00$	$2.39E-06$	$1.92E+00$	$6.35E-07$
	Rank	6	1	7	8	4	3	4	2
$f_{22}$	Mean	$-9.89E+00$	$-1.04E+01$	$-9.87E+00$	$-9.91E+00$	$-9.92E+01$	$-1.04E+01$	$-1.04E+01$	$-1.04E+01$
	SD	$1.94E+00$	$6.43E-07$	$1.61E+00$	$2.20E+00$	$1.58E+00$	$2.51E-06$	$5.14E-07$	$1.59E-06$
	Rank	7	2	8	6	5	4	1	3
$f_{23}$	Mean	$-9.95E+00$	$-1.01E+01$	$-9.40E+00$	$-8.45E+00$	$-1.01E+01$	$-1.05E+01$	$-9.76E+00$	$-1.05E+01$
	SD	$8.75E-07$	$4.69E-06$	$1.33E-06$	$2.29E+00$	$2.45E-05$	$2.28E-06$	$2.04E+00$	$5.04E-07$
	Rank	5	3	7	8	4	2	6	1
Average rank for $f_{14}-f_{23}$		6.30	3.30	5.50	6.10	5.60	2.80	4.30	1.90
Final rank for $f_{14}-f_{23}$		8	3	5	7	6	2	4	1

TABLE 8: Wilcoxon rank-sum test on benchmark functions.

DLGWO vs.	GWO $p$	RWGWO $p$	learnGWO $p$	GWOCS $p$	IGWO $p$	SOGWO $p$	MGWO $p$	
$f_1$	NaN	≈	NaN	≈	NaN	≈	NaN	≈
$f_2$	$1.21E-12$	+	$1.21E-12$	+	$1.21E-12$	+	NaN	≈
$f_3$	$3.02E-11$	+	$3.02E-11$	+	$3.02E-11$	+	$7.77E-09$	+
$f_4$	$1.21E-12$	+	$1.21E-12$	+	$1.21E-12$	+	$3.02E-11$	-
$f_5$	$3.02E-11$	+	$3.02E-11$	+	$3.02E-11$	+	$3.02E-11$	+
$f_6$	$3.02E-11$	+	$5.09E-06$	-	$3.02E-11$	+	$3.02E-11$	+
$f_7$	$2.95E-06$	+	$3.02E-11$	+	$3.02E-11$	+	$7.04E-07$	+
$f_8$	$3.02E-11$	+	$3.02E-11$	+	$3.02E-11$	+	$3.02E-11$	+
$f_9$	NaN	≈	NaN	≈	NaN	≈	NaN	≈
$f_{10}$	$4.50E-11$	+	$3.24E-07$	+	$3.24E-07$	+	$4.72E-08$	-
$f_{11}$	NaN	≈	NaN	≈	NaN	≈	NaN	≈
$f_{12}$	$1.21E-12$	+	$1.09E-10$	+	$1.21E-12$	+	$1.21E-12$	+
$f_{13}$	$1.21E-12$	+	$6.54E-09$	+	$1.21E-12$	+	$1.21E-12$	+
$f_{14}$	$2.09E-09$	+	$2.82E-06$	-	$1.99E-11$	+	$5.12E-09$	+
$f_{15}$	$4.05E-09$	+	$8.31E-08$	+	$4.82E-08$	+	$5.24E-06$	+
$f_{16}$	$3.22E-06$	+	$1.92E-04$	+	$1.49E-05$	+	$1.05E-08$	+
$f_{17}$	$2.11E-04$	+	$5.06E-06$	-	$4.32E-01$	≈	$4.05E-11$	+
$f_{18}$	$3.31E-09$	+	$2.02E-06$	+	$8.52E-05$	+	$1.41E-09$	+
$f_{19}$	$4.33E-05$	+	$2.39E-11$	+	$7.15E-05$	+	$9.18E-10$	+
$f_{20}$	$2.17E-04$	+	$4.11E-03$	+	$5.51E-04$	+	$5.82E-03$	+
$f_{21}$	$6.55E-05$	+	$1.89E-02$	≈	$4.37E-05$	+	$1.22E-05$	+
$f_{22}$	$6.44E-12$	+	$8.74E-01$	≈	$6.44E-12$	+	$6.44E-12$	+
$f_{23}$	$8.24E-06$	+	$4.52E-01$	≈	$1.52E-06$	+	$2.33E-08$	+

TABLE 9: Statistical results of the Wilcoxon rank-sum test.

DLGWO vs.		$f_1-f_7$	$f_8-f_{13}$	$f_{14}-f_{23}$	Sum
Wilcoxon's rank-sum test (+/≈/-)	GWO	6/1/0	4/2/0	10/0/0	20/3/0
	RWGWO	5/1/1	4/2/0	5/3/2	14/6/3
	learnGWO	6/1/0	4/2/0	9/1/0	19/4/0
	GWOCS	4/2/1	4/2/0	10/0/0	18/4/1
	IGWO	2/3/2	3/2/1	9/1/0	14/6/3
	SOGWO	6/1/0	4/2/0	8/1/1	18/4/1
	MGWO	4/2/1	4/2/0	8/1/1	16/5/2

function evaluations. On  $f_3$ , all the methods have similar convergence trends, but IGWO has the highest search accuracy, followed by DLGWO. On  $f_4$ , GWOCS, MGWO, and IGWO can exploit the search space more successfully than the DLGWO, whereas the DLGWO takes the fourth place. The convergence rate for  $f_5$  is similar for all the algorithms, and they converge fast during a few iterations, but then reveal stagnation behaviors. On  $f_6$  and  $f_7$ , DLGWO exhibits the fastest convergence speed and the highest convergence accuracy. The same trend can be observed on  $f_8$ . On  $f_9$  and  $f_{11}$ , DLGWO and other GWO variants can all locate the global optimum. On  $f_{10}$ , DLGWO exhibits slightly higher convergence rate compared with other optimizers. On  $f_{12}$  and  $f_{13}$ , DLGWO and RWGWO are top 2 optimizers, and they can explore the search space with more efficiency when other methods reach stagnation prematurely. According to these results, the utilized DLS and Levy flight in the proposed DLGWO algorithm can generally improve its convergence performance.

**5.6. Comparison of the DLGWO with Other Metaheuristics.** In this section, the performance of the DLGWO is examined with other well-established metaheuristics which can be

generally divided into two categories: (1) recently developed algorithms, including TLBO [19], SSA [20], SCA [21], WOA [22], and BOA [23]; (2) high-performance algorithms, including LSHADE [57] (champion optimizer in the CEC 2014 test) and LSHADE-cnEpSin [58] (champion optimizer in the CEC 2017 test). The parameter settings for these algorithms are given in Table 10. The experimental conditions are the same as in the previous section: all algorithms are implemented over 30 independent runs with 40 search agents and the maximum number of function evaluations of 300000.

**5.7. Algorithm Accuracy Analysis.** Table 11 provides the comparison results on unimodal functions. For the unimodal functions ( $f_1-f_7$ ), DLGWO attains the best rank for 4 functions, i.e.,  $f_1$ ,  $f_2$ ,  $f_4$ , and  $f_7$ , and obtains the global optimum for  $f_1$  and  $f_2$ . LSHADE achieves the best rank for 5 functions, one more than the DLGWO, and finds the global optimum for  $f_1$ ,  $f_2$ ,  $f_3$ , and  $f_6$ , two more than the DLGWO. LSHADE-cnEpSin achieves the best rank for 4 functions, the same as the DLGWO, but also achieves the global optimum for  $f_1$ ,  $f_2$ ,  $f_3$ , and  $f_6$ , two more than the DLGWO. On  $f_4$ , DLGWO obtains the highest accuracy of

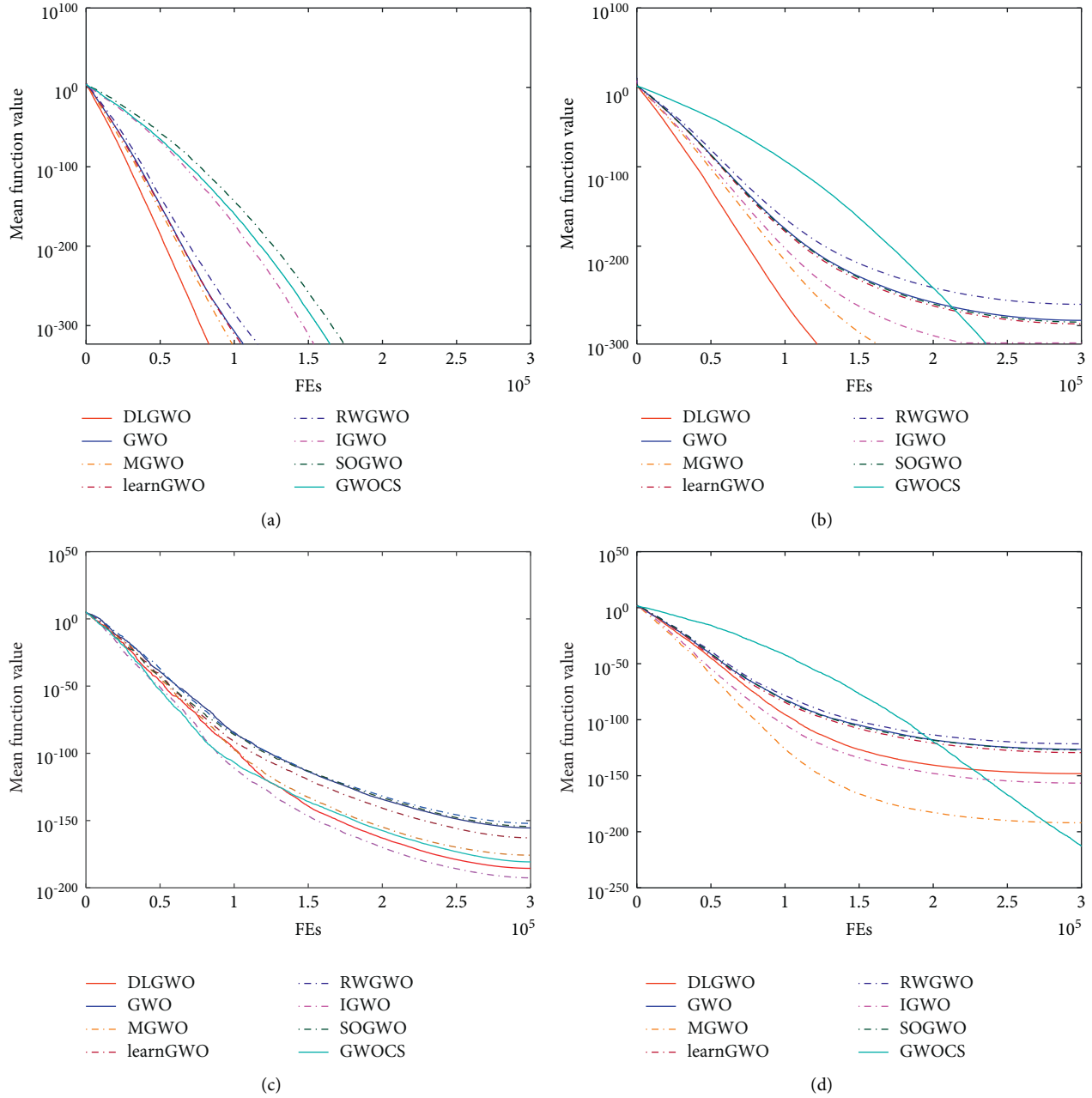


FIGURE 9: Convergence behavior on test functions  $f_1-f_4$ .

$9.57E-148$ , which is a great superiority to LSHADE and LSHADE-cnEpSin. LSHADE and LSHADE-cnEpSin overtake the other methods on  $f_5$  with apparently higher search accuracy, and DLGWO comes in the third place. Compared with the rest metaheuristics, i.e., TLBO, SSA, SCA, WOA, and BOA, DLGWO exhibits distinguished performance since its average rank is smaller than these algorithms. To sum up, the proposed DLGWO shows strong performance on unimodal functions, but its exploitation ability is worse than CEC 2014 champion algorithm LSHADE and CEC 2017 champion algorithm LSHADE-cnEpSin. In the final rank, LSHADE-cnEpSin and LSHADE are the top 2 optimizers, with DLGWO ranking third, followed by TLBO,

WOA, BOA and SSA (tied for the sixth place), and SCA. With respect to the multimodal functions ( $f_8-f_{13}$ ) shown in Table 12, WOA outperforms the other methods on  $f_8$  with the highest accuracy of  $-1.23E+04$ , followed by the proposed DLGWO with an accuracy of  $-8.29E+03$ . Among 8 algorithms, 5 algorithms can achieve the global optimum on  $f_9$ , i.e., SCA, WOA, LSHADE, LSHADE-cnEpSin, and DLGWO. On  $f_{10}$ , LSHADE-cnEpSin has the best result, but DLGWO also obtains competitive result with an accuracy of  $4.64E-15$ . The global optimum of  $f_{11}$  can be achieved by 6 methods, i.e., TLBO, SCA, WOA, LSHADE, LSHADE-cnEpSin, and DLGWO. TLBO takes the first place on  $f_{12}$  with the highest search precision of  $2.45E-24$ , followed by

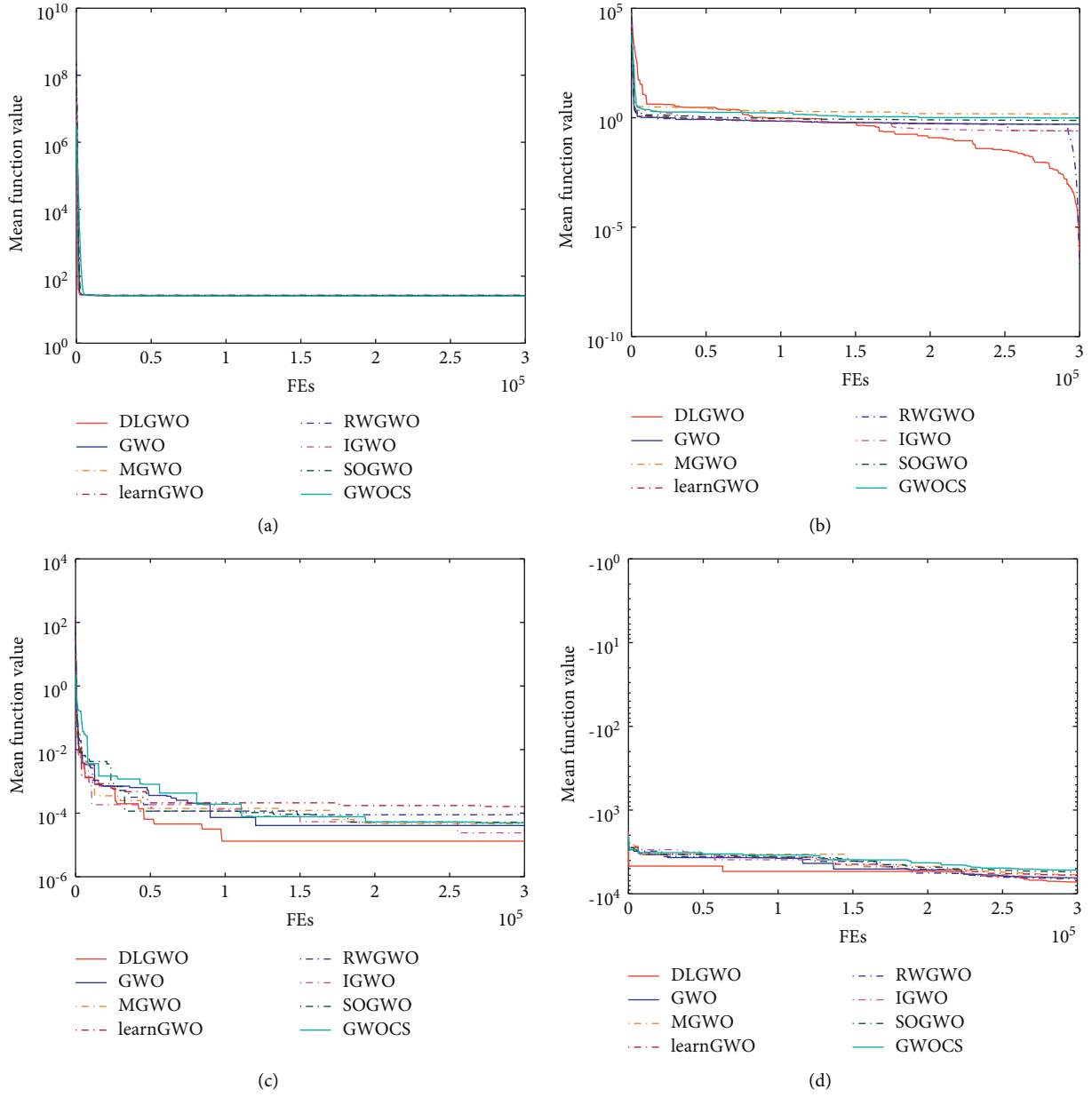
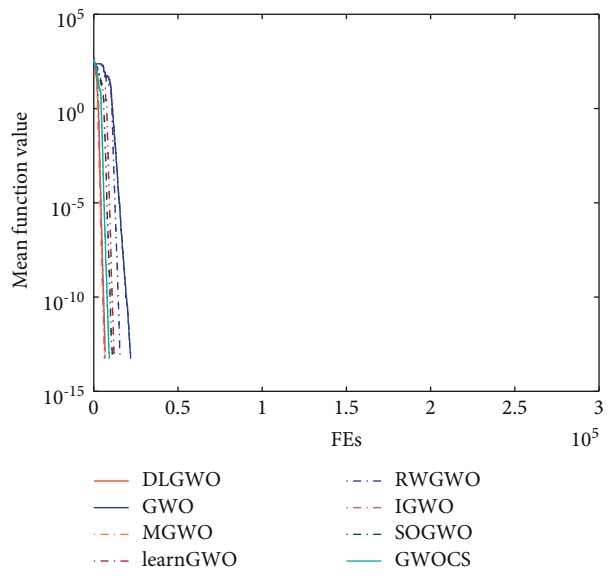
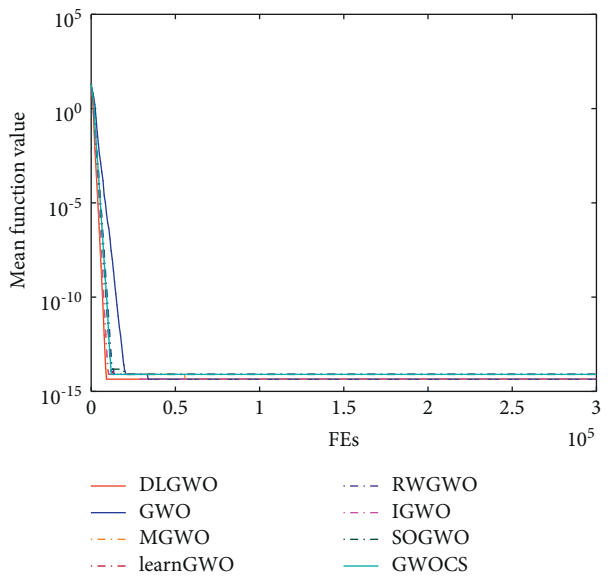


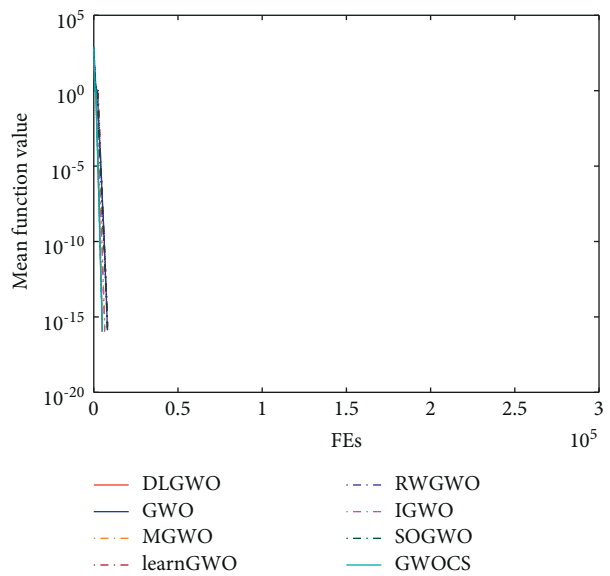
FIGURE 10: Continued.



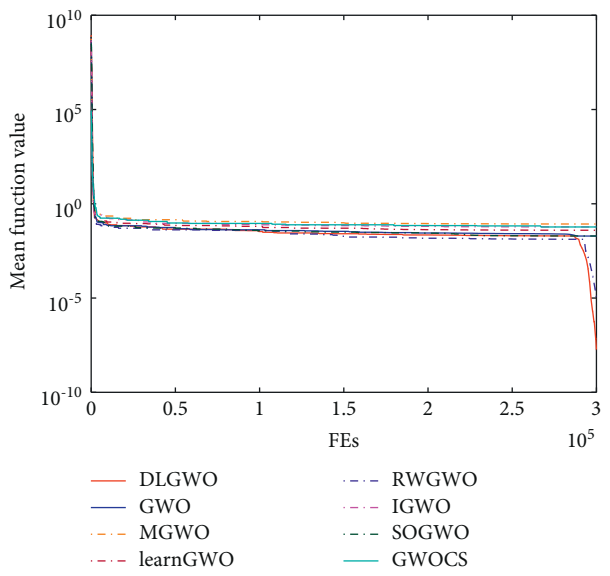
(e)



(f)



(g)



(h)

FIGURE 10: Continued.

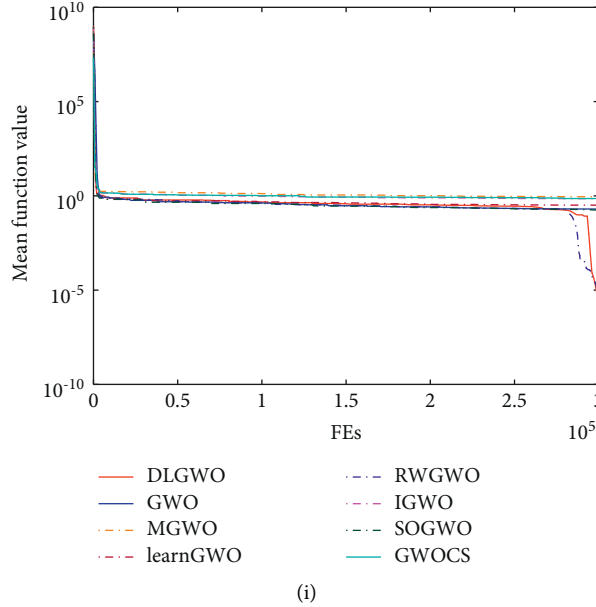


FIGURE 10: Convergence behavior on test functions  $f_5$ - $f_{13}$ .

TABLE 10: Parameter settings of the tested algorithm.

Algorithms	Parameters
TLBO [19]	$TF$ is randomly selected with 1 or 2
SSA [20]	$c_1 = [2, 0]$
SCA [21]	$a = 2, r1 = [a, 0]$
WOA [22]	$a_1 = [2, 0], a_2 = [-1, -2]$
BOA [23]	$p = 0.6, a = 0.1, c = 0.01$
LSHADE [57]	$p = 0.11, r^{arc} = 2.6, H = 6$
LSHADE-cnEpSin [58]	$P_b = 0.4, P_s = 0.5, p = 0.11,$ $r^{arc} = 1.4, \text{freq-inti} = 0.5, H = 5$

LSHADE-cnEpSin and DLGWO. On  $f_{13}$ , SSA is the most efficient optimizer. According to the average rank and final rank of these algorithms on multimodal problems, DLGWO is inferior to CEC 2017 champion algorithm LSHADE-cnEpSin, but outperforms CEC 2014 champion algorithm LSHADE with slight advantage, and is superior to the other algorithms. Hence, the experimental results demonstrate the exploration capacity of the proposed DLGWO.

Table 13 gives the statistical results on the fixed-dimension multimodal functions ( $f_{14}$ - $f_{23}$ ). According to the obtained results in Table 13, SSA, SCA, LSHADE-cnEpSin, and DLGWO obtain the highest accuracy on  $f_{14}$ , but SSA has the smallest standard deviation. On  $f_{15}$ , LSHADE-cnEpSin achieves the best result with the highest search accuracy of  $2.11E-07$ , followed by LSHADE and BOA, with DLGWO ranking fourth. On  $f_{16}$ - $f_{19}$ , all algorithms can obtain or locate close to the global optimum ( $-1.0316$  for  $f_{16}$ ,  $0.398$  for  $f_{17}$ ,  $3.00$  for  $f_{18}$ , and  $-3.86$  for  $f_{19}$ ), but they have different standard deviations, which indicate different performance stability of algorithms. TLBO attains the smallest standard deviations for  $f_{16}$ ,  $f_{17}$ , and  $f_{19}$ . On  $f_{18}$ , LSHADE and LSHADE-cnEpSin have the most robust performance and achieve the smallest standard deviations.

LSHADE obtains the best result on  $f_{20}$ , while DLGWO also provides competitive result with an accuracy of  $-3.28E+00$ . On  $f_{21}$ , WOA and DLGWO are the best performed algorithms, both methods achieve the highest search precision of  $-1.01E+01$ , but the standard deviation of WOA is slightly smaller. On  $f_{22}$  and  $f_{23}$ , DLGWO exhibits a noticeable advantage in terms of search precision and stability. From the average rank and final rank of these methods, DLGWO is the best optimizer in solving the fixed-dimension multimodal functions, with LSHADE-cnEpSin and LSHADE ranking second and third, followed by WOA and TLBO (tied for the fourth place), SSA, BOA, and SCA. Moreover, from Table 13, it can also be observed that DLGWO exhibits larger standard deviation on  $f_{16}$ - $f_{19}$  compared with several non-GWO competitors. The unstable performance on these functions can be attributed to the defect in the original GWO design that has a search tendency towards the origin of the coordinate system. When the global optimum of functions is shifted from the origin, this defect can affect the GWO and its modified variants [59].

Tables 14 and 15 provide the overall Wilcoxon rank-sum test results between DLGWO and each metaheuristic. From Table 14, DLGWO generally shows strong performance in comparison with other non-GWOs. Even compared with CEC 2017 champion algorithm LSHADE-cnEpSin and CEC 2014 champion algorithm LSHADE, DLGWO can obtain competitive results of 7/8/8 and 7/10/6, respectively. With regard to the rest optimizers, it shows superior performance and achieves results of 13/4/6 vs. TLBO, 15/1/7 vs. SSA, 19/4/0 vs. SCA, 12/7/4 vs. WOA, and 18/4/1 vs. BOA.

**5.8. Convergence Behavior Analysis.** The convergence curves of the DLGWO and the other metaheuristics based on 4 test



TABLE 11: Results on benchmark functions  $f_1-f_7$ .

	Metrics	TLBO	SSA	SCA	WOA	BOA	LSHADE	LSHADE-c	DLGWO
$f_1$	Mean	0.00E+00	4.66E-09	3.46E-43	0.00E+00	1.42E-10	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	1.19E-09	6.93E-43	0.00E+00	5.65E-12	0.00E+00	0.00E+00	0.00E+00
	Rank	1	6	5	1	7	1	1	1
$f_2$	Mean	0.00E+00	3.81E-03	4.93E-46	0.00E+00	4.77E-08	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	6.26E-03	9.66E-46	0.00E+00	3.81E-09	0.00E+00	0.00E+00	0.00E+00
	Rank	1	7	6	1	8	1	1	1
$f_3$	Mean	3.52E-145	6.23E-08	1.73E-01	1.81E+00	1.15E-10	0.00E+00	0.00E+00	6.22E-181
	SD	4.05E-145	1.75E-08	2.04E-01	3.30E-01	5.43E-12	0.00E+00	0.00E+00	4.26E-182
	Rank	4	6	7	8	5	1	1	3
$f_4$	Mean	4.19E-122	1.92E-02	2.07E-04	1.94E-02	6.43E-08	2.39E-55	2.46E-72	9.57E-148
	SD	2.06E-122	4.36E-03	3.67E-04	3.88E-03	2.25E-09	7.58E-56	1.22E-72	1.04E-149
	Rank	2	7	6	8	5	4	3	1
$f_5$	Mean	2.39E+01	9.34E+01	2.75E+01	2.46E+01	2.38E+01	1.74E+00	5.52E+00	2.36E+01
	SD	1.84E-01	7.11E+01	7.45E-01	3.79E-01	1.57E+00	9.29E-01	1.05E+00	3.88E-01
	Rank	5	8	7	6	4	1	2	3
$f_6$	Mean	4.19E-22	3.55E-09	4.70E-01	3.74E-06	3.74E-01	0.00E+00	0.00E+00	5.05E-07
	SD	5.65E-22	1.02E-09	2.62E-02	1.67E-06	2.82E-01	0.00E+00	0.00E+00	2.36E-07
	Rank	3	4	8	6	7	1	1	5
$f_7$	Mean	1.66E-04	2.13E-03	9.19E-03	1.92E-04	2.90E-02	1.75E-03	1.33E-04	2.20E-05
	SD	3.71E-05	1.23E-04	3.52E-03	2.92E-04	2.13E-03	8.52E-04	2.21E-04	7.65E-05
	Rank	3	6	7	5	8	4	2	1
Average rank for $f_1-f_7$		2.71	6.28	6.57	5.00	6.28	1.85	1.57	2.14
Final rank for $f_1-f_7$		4	6	8	5	6	2	1	3

TABLE 12: Results on benchmark functions  $f_8-f_{13}$ .

	Metrics	TLBO	SSA	SCA	WOA	BOA	LSHADE	LSHADE-c	DLGWO
$f_8$	Mean	-7.24E+03	-7.80E+03	-6.40E+03	-1.23E+04	-7.06E+03	-7.95E+03	-7.22E+03	-8.29E+03
	SD	6.11E+02	1.01E+03	1.74E+02	3.41E+02	3.29E+02	6.78E+02	5.22E+02	9.02E+02
	Rank	5	4	8	1	7	3	6	2
$f_9$	Mean	5.55E+00	5.17E+01	0.00E+00	0.00E+00	1.80E+02	0.00E+00	0.00E+00	0.00E+00
	SD	2.26E-01	2.15E+01	0.00E+00	0.00E+00	1.09E+02	0.00E+00	0.00E+00	0.00E+00
	Rank	6	7	1	1	8	1	1	1
$f_{10}$	Mean	5.62E-15	1.69E+00	7.99E-15	2.66E-15	1.96E+01	8.88E-16	8.88E-16	4.64E-15
	SD	2.05E-15	7.71E-01	7.20E-15	2.05E-15	6.23E-01	3.49E-16	2.02E-16	1.58E-15
	Rank	5	7	6	3	8	2	1	4
$f_{11}$	Mean	0.00E+00	1.14E-02	0.00E+00	0.00E+00	8.57E-11	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	2.83E-03	0.00E+00	0.00E+00	2.84E-11	0.00E+00	0.00E+00	0.00E+00
	Rank	1	8	1	1	7	1	1	1
$f_{12}$	Mean	2.45E-24	1.73E-01	3.24E-01	1.12E-06	1.10E-01	2.21E-08	9.26E-09	1.79E-08
	SD	3.77E-24	2.99E-01	1.87E-01	1.03E-07	6.29E-02	7.55E-08	8.24E-10	4.81E-09
	Rank	1	7	8	5	6	4	2	3
$f_{13}$	Mean	8.37E-02	1.09E-10	2.06E+00	1.23E-05	1.85E+00	8.89E-07	1.62E-10	2.40E-07
	SD	7.45E-02	3.82E-11	1.05E-01	5.28E-06	4.43E-01	4.55E-08	2.34E-10	7.22E-08
	Rank	6	1	8	5	7	4	2	3
Average rank for $f_8-f_{13}$		4.00	5.67	5.33	2.66	7.17	2.50	2.17	2.33
Final rank for $f_8-f_{13}$		5	7	6	4	8	3	1	2

functions are presented in Figure 11. As shown in Figure 11, DLGWO has the fastest convergence speed on  $f_2$ , followed by LSHADE-cnEpSin and LSHADE. WOA does not converge fast in the initial stage of iterations, but it then accelerates and converges fast in further iterations. As for the other methods SSA, SCA, and BOA, they converge

prematurely and cannot find solutions with high accuracy. On  $f_3$ , LSHADE-cnEpSin and LSHADE have the highest convergence accuracy, although they cannot converge fast in the first half of iterations. DLGWO is inferior to these two algorithms in terms of search accuracy, but it shows an agreeable convergence behavior during the whole iterative

TABLE 13: Results on benchmark functions  $f_{14}-f_{23}$ .

Metrics		TLBO	SSA	SCA	WOA	BOA	LSHADE	LSHADE-c	DLGWO
$f_{14}$	Mean	1.02E+00	9.98E-01	9.98E-01	3.43E+00	1.24E+00	1.01E+00	9.98E-01	9.98E-01
	SD	6.21E-01	1.35E-16	4.30E-09	4.88E+00	4.97E-01	4.12E-01	2.24E-15	3.78E-13
	Rank	6	1	4	8	7	5	2	3
$f_{15}$	Mean	7.42E-04	1.22E-03	3.19E-04	3.14E-04	3.09E-04	4.85E-05	2.11E-07	3.10E-04
	SD	3.81E-04	2.64E-16	9.75E-06	1.08E-05	1.30E-06	2.21E-05	6.44E-07	3.41E-10
	Rank	7	8	6	5	3	2	1	4
$f_{16}$	Mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	SD	0.00E+00	2.71E-16	1.09E-06	2.87E-15	3.32E-15	2.11E-09	9.84E-10	1.03E-11
	Rank	1	2	8	3	4	7	6	5
$f_{17}$	Mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	SD	0.00E+00	0.00E+00	1.16E-05	2.27E-10	5.06E-05	2.44E-09	7.11E-10	8.29E-10
	Rank	1	1	7	3	8	6	4	5
$f_{18}$	Mean	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	SD	1.33E-15	7.14E-15	2.62E-07	4.09E-09	2.67E-05	0.00E+00	0.00E+00	5.72E-10
	Rank	3	4	7	6	8	1	1	5
$f_{19}$	Mean	-3.86E+00	-3.86E+00	-3.85E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	SD	0.00E+00	4.44E-16	6.99E-05	3.94E-03	6.82E-03	5.45E-08	2.72E-06	9.86E-08
	Rank	1	2	6	7	8	3	5	4
$f_{20}$	Mean	-3.22E+00	-3.20E+00	-2.98E+00	-3.29E+00	-3.27E+00	-3.30E+00	-3.27E+00	-3.28E+00
	SD	1.55E-02	3.87E-12	3.97E-01	5.98E-02	1.55E-10	1.04E-10	2.55E-08	5.81E-02
	Rank	6	7	8	2	4	1	5	3
$f_{21}$	Mean	-9.35E+00	-9.66E+00	-8.92E+00	-1.01E+01	-7.49E+00	-9.94E+00	-1.01E+01	-1.01E+01
	SD	4.32E+00	4.06E+00	2.03E+00	4.55E-07	1.26E+00	3.11E-06	4.15E-05	6.27E-07
	Rank	6	5	7	1	8	4	3	2
$f_{22}$	Mean	-9.64E+00	-9.53E+00	-7.03E+00	-1.02E+01	-8.97E+00	-1.02E+00	-9.99E+00	-1.04E+01
	SD	1.77E+00	1.09E+00	1.71E+00	4.80E-07	2.04E+00	2.85E-07	5.62E-05	7.71E-07
	Rank	5	6	8	3	7	2	4	1
$f_{23}$	Mean	-1.02E+00	-9.56E+00	-9.98E+00	-1.02E+01	-7.28E+00	-1.02E+00	-1.02E+00	-1.05E+01
	SD	1.85E+00	2.62E+00	1.56E+00	6.45E-07	1.43E+00	1.92E+00	1.15E+00	2.02E-07
	Rank	4	7	6	2	8	5	3	1
Average rank for $f_{14}-f_{23}$		4.00	4.30	6.70	4.00	6.50	3.60	3.40	3.30
Final rank for $f_{14}-f_{23}$		4	6	8	4	7	3	2	1

TABLE 14: Wilcoxon rank-sum test on benchmark functions.

DLGWO vs.	TLBO		SSA		SCA		WOA		BOA		LSHADE		LSHADE-c
	$p$		$p$		$p$		$p$		$p$		$p$		$p$
$f_1$	NaN	≈	NaN	≈	NaN	≈	NaN	≈	NaN	≈	NaN	≈	NaN
$f_2$	NaN	≈	NaN	≈	NaN	≈	NaN	≈	NaN	≈	NaN	≈	NaN
$f_3$	4.62E-10	+	1.21E-12	+	1.21E-12	+	1.21E-12	+	1.21E-12	+	2.24E-11	+	2.24E-11
$f_4$	2.20E-09	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11
$f_5$	1.05E-04	+	3.02E-11	+	4.01E-09	+	7.44E-09	+	2.52E-03	+	3.02E-11	-	3.02E-11
	7.39E-09	-	5.44E-07	-	3.02E-11	+	1.89E-03	+	3.02E-11	+	3.02E-11	-	3.02E-11
$f_7$	4.52E-02	≈	3.02E-11	+	3.02E-11	+	1.09E-03	+	3.02E-11	+	1.43E-03	+	7.01E-02
$f_8$	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	-	3.02E-11	+	3.02E-11	+	3.02E-11
$f_9$	1.21E-12	+	1.21E-12	+	NaN	≈	NaN	≈	1.21E-12	+	NaN	≈	NaN
$f_{10}$	3.02E-11	+	3.02E-11	+	3.02E-11	+	8.22E-07	-	3.02E-11	+	9.22E-09	-	9.22E-09
$f_{11}$	NaN	≈	1.21E-12	+	NaN	≈	NaN	≈	1.62E-02	≈	NaN	≈	NaN
$f_{12}$	3.02E-11	-	3.02E-11	+	3.02E-11	+	2.26E-04	+	3.02E-11	+	4.89E-02	≈	2.72E-02
$f_{13}$	5.79E-06	+	8.56E-04	-	2.42E-09	+	3.48E-04	+	3.57E-09	+	1.37E-01	≈	8.79E-04
$f_{14}$	2.05E-07	+	6.55E-06	-	4.72E-04	+	4.64E-09	+	1.35E-08	+	3.18E-07	+	2.14E-04
$f_{15}$	1.55E-03	+	3.04E-05	+	6.85E-02	≈	5.36E-02	≈	2.90E-01	≈	2.83E-02	≈	1.28E-03
$f_{16}$	3.02E-11	-	6.55E-10	-	3.75E-07	+	7.20E-10	-	8.11E-10	-	2.61E-04	+	1.22E-03
$f_{17}$	1.21E-12	-	1.21E-12	-	5.07E-06	+	4.29E-02	≈	4.10E-06	+	1.22E-02	≈	5.74E-02
$f_{18}$	4.24E-07	-	2.27E-06	-	7.02E-04	+	1.74E-02	≈	6.62E-06	+	3.02E-11	-	3.02E-11
$f_{19}$	3.02E-11	-	4.40E-08	-	6.76E-05	+	1.11E-06	+	6.87E-07	+	4.50E-01	≈	4.08E-03
$f_{20}$	1.49E-04	+	5.84E-05	+	4.05E-05	+	2.87E-04	-	4.63E-03	+	1.32E-06	-	1.12E-03
$f_{21}$	6.00E-07	+	4.77E-06	+	3.02E-11	+	2.13E-01	≈	3.02E-11	+	5.91E-03	+	5.24E-01
$f_{22}$	6.84E-05	+	1.70E-05	+	1.63E-07	+	1.66E-03	+	6.77E-06	+	2.90E-02	≈	7.40E-04
$f_{23}$	3.36E-04	+	2.58E-06	+	4.22E-06	+	8.02E-03	+	1.09E-10	+	2.60E-04	+	5.72E-04

TABLE 15: Statistical results of the Wilcoxon rank-sum test.

DLGWO vs.		$f_1-f_7$	$f_8-f_{13}$	$f_{14}-f_{23}$	Sum
Wilcoxon's rank-sum test (+/ ≈ /-)	TLBO	3/3/1	4/1/1	6/0/4	13/4/6
	SSA	5/1/1	5/0/1	5/0/5	15/1/7
	SCA	6/1/0	4/2/0	9/1/0	19/4/0
	WOA	6/1/0	2/2/2	4/4/2	12/7/4
	BOA	5/2/0	5/1/0	8/1/1	18/4/1
	LSHADE	2/2/3	1/4/1	4/4/2	7/10/6
	LSHADE-cnEpSin	1/3/3	1/3/2	5/2/3	7/8/8

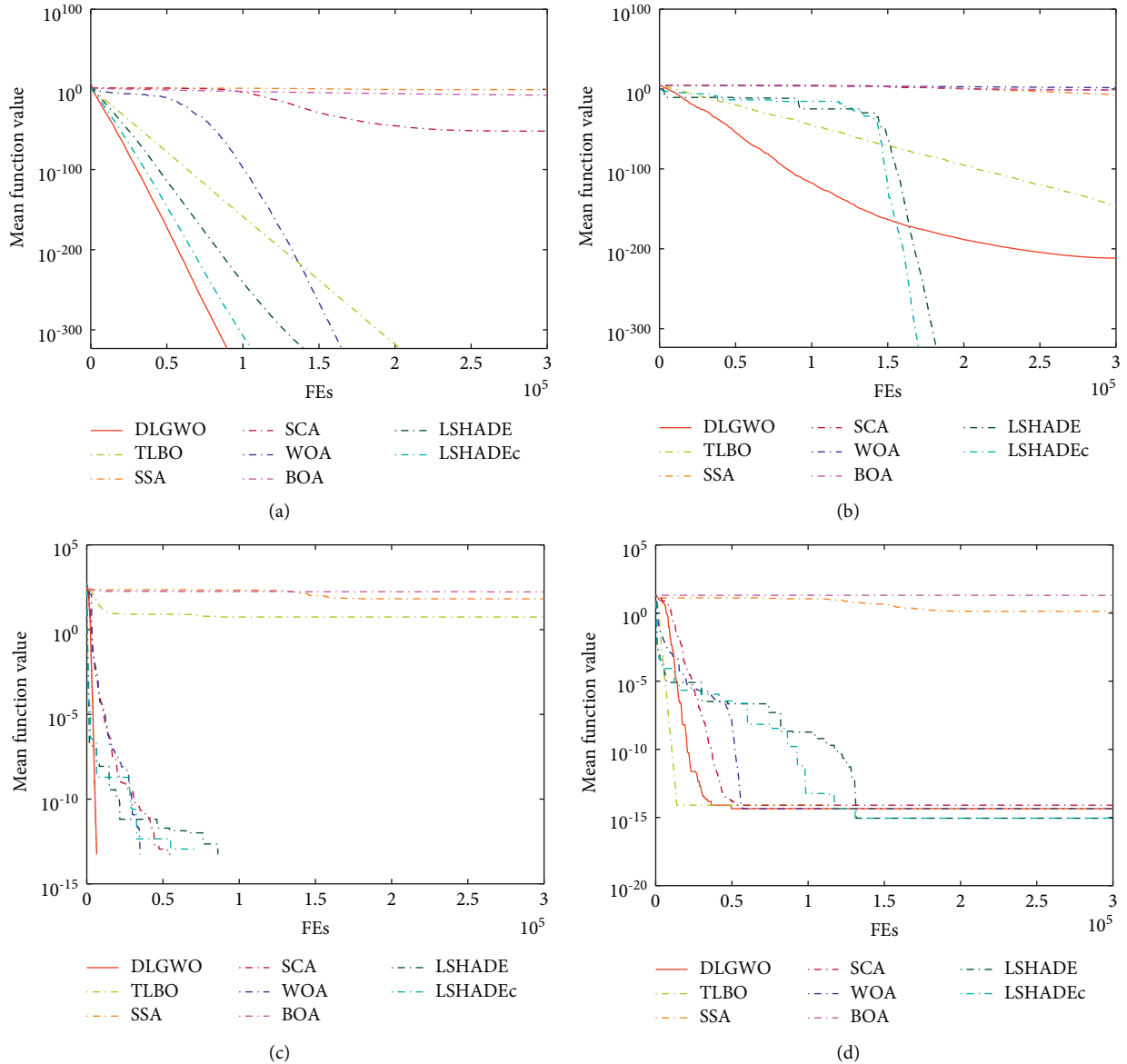


FIGURE 11: Convergence behavior on test functions.

course and has a strong ability to resist the trapping of local optima. By contrast, WOA, SSA, SCA, and BOA suffer from the immature stagnation. With regard to  $f_9$ , DLGWO has the fastest convergence speed and achieves the global optimum at a very early stage of the iterations. On  $f_{10}$ , the convergence

accuracy of the DLGWO is slightly lower than that of LSHADE-cnEpSin and LSHADE, but DLGWO has the second fastest convergence rate, only inferior to TLBO. From the above analysis, DLGWO is effective with regard to convergence accuracy and speed compared with other metaheuristics.

## 6. DLGWO for Real-World Optimization Problems

In this section, the proposed DLGWO is implemented on three classical engineering optimization problems, tension/compression spring design, welded beam design, and pressure vessel design, to verify its performance in solving real-world problems. These three problems are often employed as classical constrained optimization problems [60, 61]. For a fair comparison, the experiments on the DLGWO and the other optimizers mentioned above are conducted on the same platform based on 30 independent executions with 40 search agents and 100000 function evaluations.

*6.1. Tension/Compression Spring Design.* This is a well-known optimization problem. The structure of a tension/compression spring is shown in Figure 12. The objective is to find the minimum weight of a spring with constraints on

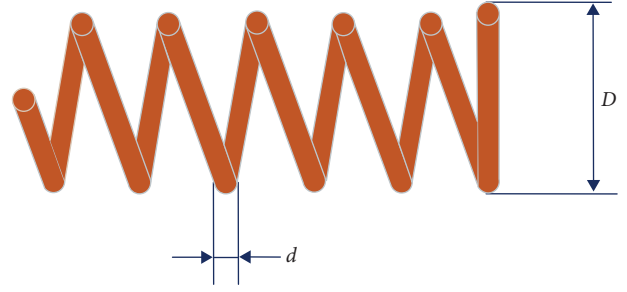


FIGURE 12: Tension-compression spring.

shear stress, surge frequency, and minimum deflection. Three decision variables are involved in the problem: wire diameter ( $d$ ), mean coil diameter ( $D$ ), and active coil number ( $N$ ).

The mathematical formulation of this problem is as follows:

Consider

$$[x_1, x_2, x_3] = [d, D, N]$$

Minimize

$$f(x) = x_1^2 x_2 (x_3 + 2)$$

subject to

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$$

$$g_2(x) = \frac{1}{5108 x_1^2} + \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} - 1 \leq 0$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

Range of variables :

$$0.05 \leq x_1 \leq 2$$

$$0.25 \leq x_2 \leq 1.3$$

$$2 \leq x_3 \leq 15$$

(17)

The numerical results obtained by the DLGWO are compared with other optimization algorithms such as GWO, learnGWO, GSA, SCA, MVO, HHO, and LSHADE, and the comparison results are listed in Table 16. As shown in Table 16, the minimum weight for the compression/tension spring is achieved by LSHADE. Although the DLGWO is inferior to LSHADE, it is the second best optimizer for the compression/tension spring problem with a promising optimization performance.

*6.2. Welded Beam Design.* The goal of this problem is to obtain the minimum cost of the welded beam subject to constraints including bending stress ( $\sigma$ ), shear stress ( $\tau$ ), buckling load ( $P_c$ ), beam deflection ( $\delta$ ), and other side constraints. The four design variables related to the problem are weld thickness ( $h$ ), length of the beam attached to the weld ( $l$ ), height of the beam ( $t$ ), and thickness of the beam ( $b$ ). The schematic of this problem is illustrated in Figure 13.

TABLE 16: Comparison results on the spring design problem.

Algorithms	$x_1$	$x_2$	$x_3$	Optimal cost
DLGWO	0.051260	0.346480	11.916330	0.012670
GWO [25]	0.050149	0.320762	13.757132	0.012711
learnGWO [50]	0.050365	0.325661	13.376170	0.012702
GSA [62]	0.050005	0.317518	14.023225	0.012722
SCA [21]	0.055355	0.451498	7.323204	0.012898
MVO [63]	0.050001	0.317402	14.032308	0.012721
WOA [22]	0.054637	0.431879	7.941518	0.012817
HHO [24]	0.052201	0.369155	10.595956	0.012671
LSHADE [57]	0.051700	0.356979	11.274151	0.012667

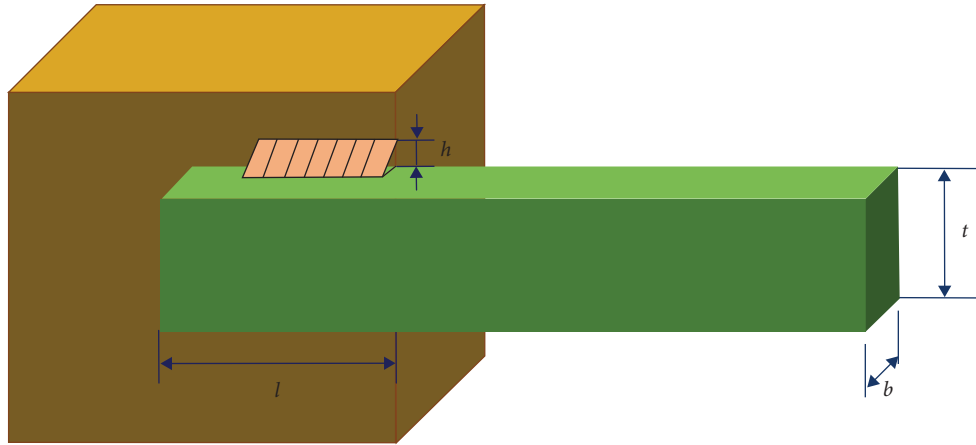


FIGURE 13: Welded beam design.

This problem can be mathematically expressed as follows:

Consider

Minimize

subject to

$$g_2(y) = \sigma(y) - \sigma_{\max} \leq 0$$

$$g_3(y) = y_1 - y_4 \leq 0$$

$$g_4(y) = \delta(y) - \delta_{\max} \leq 0$$

$$g_5(y) = P - P_c(y) \leq 0$$

$$g_6(y) = 0.125 - y_1 \leq 0$$

$$g_7(y) = 1.10471y_1^2 + 0.04811y_3y_4(14.0 + y_2) - 5.0 \leq 0$$

Range of variables :

$$0.1 \leq y_2 \leq 10$$

$$0.1 \leq y_3 \leq 10$$

$$0.1 \leq y_4 \leq 2$$

$$[y_1, y_2, y_3, y_4] = [h, l, t, b]$$

$$f(y) = 1.10471y_1^2y_2 + 0.0481y_3y_4(14.0 + y_2)$$

$$g_1(y) = \tau(y) - \tau_{\max} \leq 0$$

(18)

Table 17 gives the numerical results obtained by the DLGWO and the other methods such as GWO, RWGWO, MFO, KH, BOA, SSA, MPA, and JADE. As

demonstrated by Table 17, the minimum weight for the welded beam design problem is obtained by the DLGWO.

TABLE 17: Comparison results on the welded beam problem.

Algorithms	$y_1$	$y_2$	$y_3$	$y_4$	Optimal cost
DLGWO	0.205729	3.470489	9.036624	0.205729	1.724853
GWO [25]	0.205709	3.469307	9.040968	0.205712	1.725276
RWGWO [39]	0.205519	3.475426	9.036760	0.205753	1.725179
MFO [64]	0.205727	3.470552	9.036614	0.205731	1.724861
KH [65]	0.203720	3.530723	9.036809	0.205838	1.730702
BOA [23]	0.205674	3.474845	9.036501	0.205744	1.725452
SSA [20]	0.204800	3.491382	9.039175	0.205724	1.726626
MPA [66]	0.205666	3.473276	9.041017	0.205726	1.725866
JADE [67]	0.205728	3.470518	9.036617	0.205730	1.724858

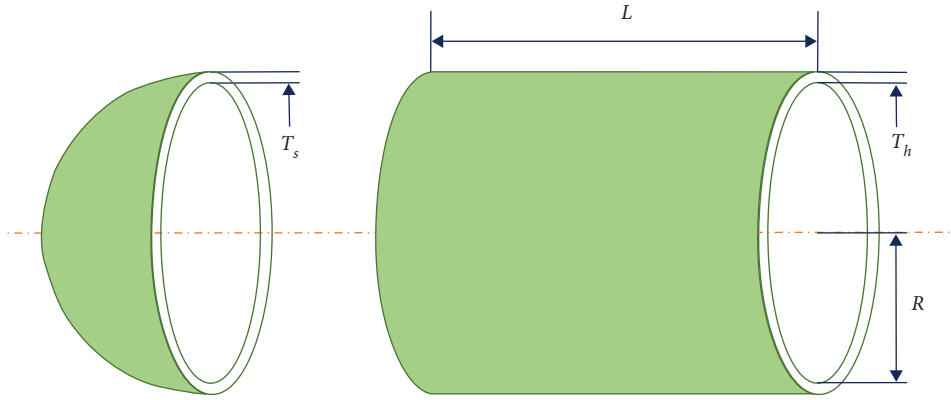


FIGURE 14: Pressure vessel design problem.

6.3. *Pressure Vessel Design Problem.* The pressure vessel design problem was proposed by Kannan and Kramer [68], and the goal of this problem is to obtain the minimum cost of a columnar vessel, which is shown in Figure 14.

The design variables are the thickness of the ball ( $T_h$ ), shell thickness ( $T_s$ ), shell length ( $l$ ), and inside radius ( $R$ ). The mathematical model can be expressed as follows:

$$\begin{aligned}
 &\text{Consider} && [z_1, z_2, z_3, z_4] = [T_s, T_h, R, l] \\
 &\text{Minimize} && f(z) = 0.6224z_1z_3z_4 + 1.7881z_2z_3^2 + 3.1661z_1^2z_4 + 19.84z_1^2z_3 \\
 &\text{subject to} && g_1(z) = -z_1 + 0.0193z_3 \leq 0 \\
 &&& g_2(z) = -z_2 + 0.00954z_3 \leq 0 \\
 &&& g_3(z) = -\pi z_3^2 z_4 - \frac{4}{3}\pi z_3^3 + 1296000 \leq 0 \\
 &&& g_4(z) = z_4 - 240 \leq 0 \\
 &\text{Range of variables :} && 0 \leq z_1 \leq 100 \\
 &&& 0 \leq z_2 \leq 100 \\
 &&& 10 \leq z_3 \leq 200 \\
 &&& 10 \leq z_4 \leq 200
 \end{aligned} \tag{19}$$

TABLE 18: Comparison results on the pressure vessel design problem.

Algorithms	$z_1$	$z_2$	$z_3$	$z_4$	Optimal cost
DLGWO	0.7785	0.3848	40.3380	199.7643	5886.5293
GWO [25]	0.7791	0.3852	40.3651	199.3867	5887.6935
EEGWO [29]	0.7921	0.3964	40.7491	194.8122	5978.3454
CMA-ES [69]	0.7805	0.3869	40.4375	198.3741	5892.6520
TSO [70]	0.7952	0.4122	41.0635	190.6280	6006.5711
CSA [71]	0.8235	0.4372	42.6094	170.7793	6081.1492
EO [72]	0.7827	0.3878	40.5559	196.7654	5896.5698
TLBO [19]	0.7963	0.4226	41.2309	188.0681	6016.6774
DA [73]	0.7784	0.3855	40.3279	199.9235	5888.7013

The optimization results for the pressure vessel design problem are listed in Table 18. The compared algorithms are GWO, EEGWO, CMA-ES, TSO, CSA, EO, TLBO, and DA. From Table 18, DLGWO can provide an excellent parameter design plan with lower cost compared to other algorithms.

## 7. Conclusion

This paper introduces DLS into the leading hierarchy of the GWO algorithm. During the process of the DLS, the delta wolf learns from the corresponding dimensions of both alpha and beta wolves to construct an exemplar wolf. Instead of learning from the three dominant wolves in each iteration, the search agents learn from the exemplar wolf which is composed of the excellent information learned from the three dominant wolves. This improves the efficiency of utilizing the population information of the GWO. Moreover, the Levy flight is adopted to reinforce the global exploration ability. Based on these two strategies, the DLGWO algorithm is proposed. To verify the validity of the proposed method for solving global optimization problems, DLGWO is compared with some variants of the GWO and some other metaheuristics based on 23 widely utilized benchmark functions. The experimental results of the benchmark functions manifest that the DLGWO exhibits a promising search performance and has a good balance between exploitation and exploration. Furthermore, DLGWO is implemented into three engineering problems, and the statistical results verify that the proposed DLGWO is an efficient and stable optimization method.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant no. 51675228) and Program of Science and Technology Development Plan of Jilin Province, China (Grant nos. 20180101052JC and 20190303020SF).

## References

- [1] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: squirrel search algorithm," *Swarm and Evolutionary Computation*, vol. 44, pp. 148–175, 2019.
- [2] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization," *Artificial Intelligence Review*, vol. 2021, Article ID 9210050, 2021.
- [3] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2233, 2019.
- [4] S. Mandavi, M. Ebrahim Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Information Sciences*, vol. 295, pp. 407–428, 2015.
- [5] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers and Industrial Engineering*, vol. 137, Article ID 106040, 2019.
- [6] S. Gupta and K. Deep, "A memory-based grey wolf optimizer for global optimization tasks," *Applied Soft Computing*, vol. 93, Article ID 106367, 2020.
- [7] M. Crepinsek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *Acm Computing Surveys*, vol. 45, no. 3, pp. 1–33, 2013.
- [8] E. Russell and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
- [9] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2007.
- [10] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [11] H. Gao, Z. Fu, C. M. Pun, J. Zhang, and S. Kwong, "An efficient artificial bee colony algorithm with an improved linkage identification method," *IEEE Transactions on Cybernetics*, pp. 1–15, 2020.
- [12] J.-Q. Li, M.-X. Song, L. Wang et al., "Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2425–2439, 2020.
- [13] W. Xi, Y. Wang, B. Chen, and H. Wu, "Iterative learning control of robot based on artificial bee colony algorithm," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 233, no. 9, pp. 1221–1238, 2019.



- [14] Y. Yu, C. Zhang, Y. Wang, and M. Zhou, "Neural network-based iterative learning control for hysteresis in magnetic shape memory alloy actuator," *IEEE/ASME Transactions on Mechatronics*, p. 1, 2021.
- [15] X. S. Yang and S. Deb, "Cuckoo search via levey flights," in *Proceedings of the 2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pp. 210–214, Coimbatore, India, December 2009.
- [16] S. I. Boushaki, N. Kamel, and B. Omar, "A new quantum chaotic cuckoo search algorithm for data clustering," *Expert Systems with Applications*, vol. 96, pp. 358–372, 2018.
- [17] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74, 2010.
- [18] C. Gan, W.-H. Cao, K.-Z. Liu, M. Wu, F.-W. Wang, and S.-B. Zhang, "A new hybrid bat algorithm and its application to the rop optimization in drilling processes," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7338–7348, 2020.
- [19] R. Venkata Rao, V. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences*, vol. 183, pp. 1–15, 2012.
- [20] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [21] S. Mirjalili, "Sca: A sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.
- [22] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [23] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.
- [24] A. Asghar Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [26] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications*, vol. 30, no. 2, pp. 413–435, 2018.
- [27] N. M. Hatta, A. M. Zain, R. Sallehuddin, Z. Shayfull, and Y. Yusoff, "Recent studies on optimisation method of grey wolf optimiser (gwo): a review (2014-2017)," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2651–2683, 2019.
- [28] C. Lu, L. Gao, and J. Yi, "Grey wolf optimizer with cellular topological structure," *Expert Systems with Applications*, vol. 107, pp. 89–114, 2018.
- [29] L. Wen, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 63–80, 2018.
- [30] X. Liu, Y. Tian, X. Lei, H. Wang, Z. Liu, and J. Wang, "An improved self-adaptive grey wolf optimizer for the daily optimal operation of cascade pumping stations," *Applied Soft Computing*, vol. 75, pp. 473–493, 2019.
- [31] S. Khalilpourazari and S. Khalilpourazary, "Optimization of production time in the multi-pass milling process via a robust grey wolf optimizer," *Neural Computing and Applications*, vol. 29, no. 12, pp. 1321–1336, 2018.
- [32] L. Rodriguez, O. Castillo, and J. Soria, "Grey Wolf Optimizer with Dynamic Adaptation of Parameters Using Fuzzy Logic," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3116–3123, Vancouver, Canada, November 2016.
- [33] A. Saxena, R. Kumar, and S. Das, " $\beta$ -Chaotic map enabled grey wolf optimizer," *Applied Soft Computing*, vol. 75, pp. 84–105, 2019.
- [34] V. Kumar and D. Kumar, "An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems," *Advances in Engineering Software*, vol. 112, pp. 231–254, 2017.
- [35] H. Xing, X. Zhou, X. Wang et al., "An integer encoding grey wolf optimizer for virtual network function placement," *Applied Soft Computing*, vol. 76, pp. 575–594, 2019.
- [36] Z. Cai, J. Gu, J. Luo et al., "Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy," *Expert Systems with Applications*, vol. 138, 2019.
- [37] Y. Yang, B. Yang, S. Wang, T. Jin, and L. Shi, "An enhanced multi-objective grey wolf optimizer for service composition in cloud manufacturing," *Applied Soft Computing*, vol. 87, Article ID 106003, 2020.
- [38] D. Miao, W. Chen, W. Zhao, and T. Demas, "Parameter estimation of pem fuel cells employing the hybrid grey wolf optimization method," *Energy*, vol. 193, Article ID 116616, 2020.
- [39] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm and Evolutionary Computation*, vol. 44, pp. 101–112, 2019.
- [40] M. A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, "Natural selection methods for grey wolf optimizer," *Expert Systems with Applications*, vol. 113, pp. 481–498, 2018.
- [41] J. Liu, Z. Yang, and D. Li, "A multiple search strategies based grey wolf optimizer for solving multi-objective optimization problems," *Expert Systems with Applications*, vol. 145, Article ID 113134, 2020.
- [42] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Applied Soft Computing*, vol. 76, pp. 16–30, 2019.
- [43] S. Dhargupta, M. Ghosh, S. Mirjalili, and R. Sarkar, "Selective opposition based grey wolf optimization," *Expert Systems with Applications*, vol. 151, Article ID 113389, 2020.
- [44] L. Wen, S. Cai, J. Jiao, M. Xu, and T. Wu, "A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models," *Energy Conversion and Management*, vol. 203, Article ID 112243, 2020.
- [45] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, A. K. Abasi, and Z. A. Alkareem Alyasseri, "A novel hybrid grey wolf optimizer with min-conflict algorithm for power scheduling problem in a smart home," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100793, 2021.
- [46] C. Qu, W. Gai, J. Zhang, and M. Zhong, "A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (uav) path planning," *Knowledge-Based Systems*, vol. 194, Article ID 105530, 2020.
- [47] B. Martin, J. Marot, and S. Bourennane, "Mixed grey wolf optimizer for the joint denoising and unmixing of multi-spectral images," *Applied Soft Computing*, vol. 74, pp. 385–410, 2019.

- [48] Z. Miao, X. Yuan, F. Zhou, X. Qiu, Y. Song, and K. Chen, "Grey wolf optimizer with an enhanced hierarchy and its application to the wireless sensor network coverage optimization problem," *Applied Soft Computing*, vol. 96, Article ID 106602, 2020.
- [49] B. Yang, X. Zhang, T. Yu, H. Shu, and Z. Fang, "Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine," *Energy Conversion and Management*, vol. 133, pp. 427–443, 2017.
- [50] Fehmi Burcin Ozsoydan, "Effects of dominant wolves in grey wolf optimization algorithm," *Applied Soft Computing*, vol. 83, Article ID 105658, 2019.
- [51] Y. Wang, T. Zhang, Z. Cai, Z. Jiang, and K. Wu, "Multi-uav coordination control by chaotic grey wolf optimization based distributed mpc with event-triggered strategy," *Chinese Journal of Aeronautics*, vol. 33, pp. 2877–2897, 2020.
- [52] G. Xu, Q. Cui, X. Shi et al., "Particle swarm optimization based on dimensional learning strategy," *Swarm and Evolutionary Computation*, vol. 45, pp. 33–51, 2019.
- [53] F. Van Den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2019.
- [54] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes," *Physical Review E*, vol. 49, no. 5, pp. 4677–4683, 1994.
- [55] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with levy flight for optimization tasks," *Applied Soft Computing*, vol. 60, pp. 115–134, 2017.
- [56] P. Yao, H. Wang, and H. Ji, "Multi-uavs tracking target in urban environment by model predictive control and improved grey wolf optimizer," *Aerospace Science and Technology*, vol. 55, pp. 131–143, 2016.
- [57] R. Tanabe and S. F. Alex, "Improving the search performance of shade using linear population size reduction," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1658–1665, Beijing, China, July 2014.
- [58] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood for Solving Cec2017 Benchmark Problems," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 372–379, Donostia, Spain, June 2017.
- [59] P. Niu, S. Niu, N. Liu, and L. Chang, "The defect of the grey wolf optimization algorithm and its verification method," *Knowledge-Based Systems*, vol. 171, pp. 37–43, 2019.
- [60] F. Z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340–356, 2007.
- [61] M. Jaberipour and E. Khorram, "Two improved harmony search algorithms for solving engineering optimization problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3316–3331, 2010.
- [62] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "Gsa: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [63] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.
- [64] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [65] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [66] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and H. G. Amir, "Marine predators algorithm: a nature-inspired meta-heuristic," *Expert Systems with Applications*, vol. 152, Article ID 113377, 2020.
- [67] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [68] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *Journal of Mechanical Design*, vol. 116, no. 2, pp. 405–411, 1994.
- [69] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [70] L. Xie, T. Han, H. Zhou, Z. Ran Zhang, B. Han, and A. Tang, "Tuna Swarm Optimization: A Novel Swarm-Based Meta-heuristic Algorithm for Global Optimization," *Computational intelligence and Neuroscience*, vol. 2021, Article ID 9210050, 2021.
- [71] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," *Computers and Structures*, vol. 169, no. 1–12, 2016.
- [72] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: a novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, Article ID 105190, 2020.
- [73] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.