IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 6, 2020, accepted June 16, 2020, date of publication June 26, 2020, date of current version July 13, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3005182*

# Grey Wolf Optimizer With a Novel Weighted Distance for Global Optimization

**FU YAN[1], XINLIANG XU[2], AND JIANZHONG XU[1]**
[1]School of Economics and Management, Harbin Engineering University, Harbin 150001, China
[2]College of Economics and Management, Northeast Agricultural University, Harbin 150030, China

Corresponding author: Xinliang Xu (xuxinliang86@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71841054, in part by the National Natural Science Foundation of Heilongjiang Province under Grant LH2019G014, in part by the 2019 Annual Basic Project of the Party's Political Construction Research Center of the Ministry of Industry and Information Technology of the People's Republic of China under Grant 19GZY411, in part by the Postdoctoral Foundation of Heilongjiang Province under Grant LBH-Z18015, in part by the Young Innovative Talents Training Program for Universities in Heilongjiang Province under Grant UNPYSCT-2018151, in part by the Project of Philosophy and Social Science of Heilongjiang Province under Grant 18GLC209, and in part by the Project of Philosophy and Social Science of Heilongjiang Province under Grant 18JYC257.

**ABSTRACT** In this paper, a new grey wolf optimizer (GWO) variant based on a novel weighted distance (WD) called the GWO-WD algorithm is presented to solve global optimization problems. First, a modified position-updating equation formulated using the proposed strategy is employed to obtain additional information and improved global solutions. Then, several of the worst individuals are eliminated and repositioned using an elimination and repositioning strategy to improve the capability of the algorithm and avoid falling into local optima. The performance of the algorithm is verified by utilizing 23 widely used benchmark test functions, the IEEE CEC2014 test suite and three well-known engineering design problems. The simulation results of the proposed algorithm are compared with those of the standard GWO algorithm, three GWO variants and several existing methods, and the proposed algorithm is revealed to be very competitive and, in many cases, superior.

**INDEX TERMS** Grey wolf optimizer, global optimization, weight distance strategy, elimination and repositioning strategy, engineering design problem.

## I. INTRODUCTION

Global optimization (GO), defined as the process of finding the best solution from all feasible solutions [1], is necessary, challenging and inevitable in most optimization problems [2]. Complex multimodal optimization problems are common in science and engineering, and most classic deterministic methods (or gradient-based methods) generally fail or produce infeasible solutions [3]. Among the most successful and competitive GO methods are swarm intelligence (SI) algorithms [2], which were developed by mimicking the survival behaviors of bird swarms, fish schools, insect colonies, bacterial growth and other animal herds and are used to solve complex problems.

With its strong optimization performance, SI has attracted considerable interest from researchers in recent years. Several well-known SI algorithms, such as particle swarm optimization (PSO) [4], artificial bee colony (ABC) optimization [5], ant colony optimization (ACO) [6], the cuckoo search (CS) method [7], the firefly algorithm (FA) [8], the multiverse optimizer (MVO) [9], the dragonfly algorithm (DA) [10], the ant lion optimizer (ALO) [11], the whale optimization algorithm (WOA) [12], moth-flame optimization (MFO) [13], the bat algorithm (BA) [14], the squirrel search algorithm (SSA) [15], the naked mole-rat (NMR) algorithm [16], the grey wolf optimizer (GWO) [17]– [19], the binary spotted hyena optimizer [20] and the sine cosine algorithm [21], [22], have been developed in the past few decades to solve GO problems. Among the current SI algorithms, GWO is a representative algorithm that has been widely employed for GO purposes. The GWO algorithm, which is inspired by the hunting behavior and leadership hierarchy of grey wolves, was first developed by Mirjalilili et. al. in 2014 [17]. According to the hunting process of wolves, four main steps are implemented in the GWO algorithm: hunting, searching, encircling, and attacking. In addition,

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

based on the leadership hierarchy of grey wolf packs, pack members in GWO are divided into four groups based on their hunting ability, namely, alpha ($\alpha$), beta ($\beta$), delta ($\delta$), and omega ($\omega$) groups, where the alpha wolf has the best hunting ability (denoted by the fitness value), the beta wolf has the second-best hunting ability, and the delta wolf has the third-best hunting ability. In GWO, the best three wolves (i.e., $\alpha$, $\beta$ and $\delta$ wolves) are responsible for guiding the $\omega$ wolves and hunting for prey; i.e., the position-updating equation of $\omega$ wolves is decided by the $\alpha$, $\beta$ and $\delta$ wolves. This mechanism of location updating creates a simple algorithm framework for GWO, is easy to implement on a computer, and requires few parameters to be adjusted. Therefore, this method has been successfully utilized in the fields of economic load dispatch (ELD) problems [23], [24], automatic control [25], image processing [26], strategic bidding in the energy market [27], machine learning [28], and aerial vehicle path planning in unmanned combat [29], among others.

However, similar to other SI algorithms, the GWO algorithm suffers from several drawbacks, such as a low solution accuracy, slow convergence speed and tendency to converge to local optima. These drawbacks are especially obvious when solving high-dimensional, complex optimization problems. To alleviate these shortcomings, researchers have developed several modified GWO algorithms, which can be divided into four categories [30], including methods that modify the updating mechanism, operators, encoding scheme for individuals, and population structure and hierarchy.

The methods that modify the updating mechanism follow two different approaches. One approach is to dynamically change the GWO control parameters, while the other involves proposing a new position-updating equation for GWO. To enhance the global exploration of GWO, Mittal *et al.* [31] designed a new control parameter $a$ by using an exponential decay function. Xu *et al.* [32] proposed a nonlinear control parameter $a$ to achieve a balance between exploration and exploitation and to accelerate the GWO convergence speed. In [33], two control parameters were redesigned, namely, $a$ and $C$, whose values were dynamically and iteratively modified. In addition, to prevent the GWO algorithm from converging to local optima, researchers have also modified the updating equation. Jaiswal *et al.* [34] introduced random weight coefficients into the GWO algorithm to modify the position-updating equation and avoid local optima. In [1], [2], the authors adopted weighted distance coefficients to update the positions of individuals instead of using a simple average of the first three best individuals. These two modified position-updating equation techniques are particularly effective for solving complex multimodal problems. To increase the diversity of potential individuals in GWO, it is efficient to adopt new operators. In [35], the authors developed a modified GWO variant based on a simple crossover operator generated by randomly selecting two different individuals. In another work, Saremi *et al.* [36] employed evolutionary population dynamics (EPD) in GWO to eliminate the worst individuals and reposition them around the three best wolves. Furthermore, modifying the encoding scheme of individuals can enhance the information capacity of individuals and increase the diversity of the population. In [37], a novel complex-valued encoding GWO algorithm (CGWO) was developed. In CGWO, the real-valued encoding method is replaced with a complex-valued method, which consists of two main parts: an imaginary part and a real part. As mentioned above, the original GWO algorithm was proposed based on a unique hierarchical population structure described in [17], [30]. Therefore, an interesting research direction is to modify the population structure and hierarchy. In a notable work, Yang *et al.* [38] proposed a GWO variant based on different leadership hierarchies. In this GWO variant, the population is divided into two independent groups: a cooperative hunting group and a random scouting group: the cooperative hunting group conducts deep exploitation, while the scouting group performs extensive exploration.

Although the performance of most modified GWO algorithms was improved after adding several additional operators or new mechanisms, two problems remained: premature convergence and an imbalance between global exploration and local exploitation. These phenomena can be explained by four aspects [39]: a) lack of sufficient diversity among individuals; b) a lack of experimental analysis and statistical results to justify the applicability and validity of the findings of previous studies; c) a lack of algorithm parameters, such as the population size and dimension size, and a poor understanding of their impacts on the optimization performance; and d) a poor balance between exploration and exploitation, which should be considered in future works.

Inspired by these factors, this work proposes a novel modified GWO variant called the grey wolf optimizer based on the weighted distance (GWO-WD). The main contributions of this paper are listed as follows.

(1) A novel weighted distance is proposed for the GWO algorithm considering both the fitness value of each leader and the corresponding control parameter $C$.

(2) The position-updating equation for the GWO algorithm is redesigned based on the proposed weighted distance to effectively accelerate the convergence speed and enhance the accuracy of the solution.

(3) A strategy for eliminating the worst individuals and repositioning them in the search space by using the three leader wolves is proposed; this approach can help the GWO algorithm avoid local optima.

The remainder of this paper is arranged as follows. Section 2 briefly explains the standard GWO algorithm. In Section 3, the proposed GWO-WD algorithm is described based on the novel weighted distance, the redesigned position-updating equation and the strategy for eliminating and repositioning the worst individuals. Section 4 describes how the algorithm parameters used in the experiment were set and presents the experimental results. Applications of the proposed GWO-WD algorithm to three real-world engineering design problems are investigated in Section 5. Finally, the conclusions are drawn in Section 6.

## II. GREY WOLF OPTIMIZER AND WEIGHTED DISTANCE

### A. STANDARD GREY WOLF OPTIMIZER

GWO, proposed by Mirjalili *et al.* [17], is a recently developed SI algorithm inspired by *Canis -lupus*. The GWO algorithm mimics the special hunting strategy of grey wolves in searching for and capturing prey with a strict division of responsibilities and mutual cooperation. In the GWO algorithm, the global best search agent in the population is called the alpha ($\alpha$) wolf. The global second- and third-best search agents are named the beta ($\beta$) and delta ($\delta$) wolves, respectively. The other search agents in the population are considered omega ($\omega$) wolves. The hunting strategy of grey wolf packs involves three steps: encircling, hunting and attacking.

The grey wolves begin to encircle the prey after determining its location. To mathematically describe the encircling process, Eq. (1) is established [16] as follows.

$$\vec{X}(t+1)=\vec{X}_p(t)-\vec{A}\cdot\left|\vec{C}\cdot\vec{X}_p(t)-\vec{X}(t)\right| \tag{1}$$

where $t$ is the current iteration, $\vec{X}(t)$ is the current position vector of a grey wolf, $\vec{X}(t+1)$ is the next position vector of the wolf, $\vec{X}_p(t)$ is the current position vector of the prey, and $\vec{A}$ and $\vec{C}$ are coefficient vectors. These coefficients can be described as follows.

$$\vec{A}=2\vec{a}\cdot\vec{r}_1-\vec{a} \tag{2}$$

$$\vec{C}=2\vec{r}_2 \tag{3}$$

where $\vec{r}_1$ and $\vec{r}_2$ are random vectors in [0, 1] and $\vec{a}$ linearly decreases from 2 to 0 with increasing number of iterations.

$$\vec{a}=2-\frac{2t}{MaxIter} \tag{4}$$

where $t$ is the number of iterations and *MaxIter* is the total number of iterations.

After the first step is completed, the predation process begins. The other wolves update their positions under the guidance of the $\alpha$, $\beta$, and $\delta$ wolves as follows [16].

$$\begin{cases}\vec{X}_1(t)=\vec{X}_\alpha(t)-\vec{A}_1\cdot\left|\vec{C}_1\cdot\vec{X}_\alpha(t)-\vec{X}(t)\right|\\ \vec{X}_2(t)=\vec{X}_\beta(t)-\vec{A}_2\cdot\left|\vec{C}_2\cdot\vec{X}_\beta(t)-\vec{X}(t)\right|\\ \vec{X}_3(t)=\vec{X}_\delta(t)-\vec{A}_3\cdot\left|\vec{C}_3\cdot\vec{X}_\delta(t)-\vec{X}(t)\right|\end{cases} \tag{5}$$

$$\vec{X}(t+1)=\frac{\vec{X}_1(t)+\vec{X}_2(t)+\vec{X}_3(t)}{3} \tag{6}$$

The third step is to attack the prey. In the GWO algorithm, the attack behavior of grey wolves is controlled by the coefficient vector $\vec{A}$, where $\vec{A}\in\left[-2\vec{a},2\vec{a}\right]$. When $\left|\vec{A}\right|\geq 1$, the grey wolves diverge from the prey and conduct a global search to find better prey; when $\left|\vec{A}\right|<1$, the grey wolves attack the prey.

### B. TWO DIFFERENT WEIGHTED DISTANCES

Jitkongchuen *et al.* presented a notable GWO variant, namely, the weighted distance GWO (WDGWO) [1]. In the WDGWO algorithm, the $\omega$ wolves update their positions by using the unique weight values of the three leaders, and these weight values are determined by the fitness value of each leader. The $\alpha$ wolf has the best weight value, and the $\beta$ and $\delta$ wolves have the second-best and third-best weight values, respectively. These weight values are calculated as follows [1].

$$W_1=\frac{f_\alpha}{f_\alpha+f_\beta+f_\delta} \tag{7}$$

$$W_2=\frac{f_\beta}{f_\alpha+f_\beta+f_\delta} \tag{8}$$

$$W_3=\frac{f_\delta}{f_\alpha+f_\beta+f_\delta} \tag{9}$$

where $f_\alpha$, $f_\beta$ and $f_\delta$ are the fitness values of the $\alpha$, $\beta$ and $\delta$ wolves, respectively.

Therefore, $\omega$ wolves will update their positions as follows [1].

$$\vec{X}(t+1)=\frac{W_1\vec{X}_1(t)+W_2\vec{X}_2(t)+W_3\vec{X}_3(t)}{3} \tag{10}$$

where $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ are calculated from Eq. (5).

In another interesting study, Malik *et al.* proposed another GWO variant, the weighted distance GWO (WdGWO), which is used mainly to improve the performance of the standard GWO algorithm in complex multimodal GO problems [2]. In the WdGWO algorithm, the weight values of the three leaders are calculated by adopting the coefficient vectors $\vec{A}_i$ and $\vec{C}_i$, as in Eq. (2) and Eq. (3), respectively, where $i=1,2,3$.

$$w_1=\vec{A}_1\cdot\vec{C}_1 \tag{11}$$

$$w_2=\vec{A}_2\cdot\vec{C}_2 \tag{12}$$

$$w_3=\vec{A}_3\cdot\vec{C}_3 \tag{13}$$

Therefore, the $\omega$ wolves update their positions as follows [2].

$$\vec{X}(t+1)=\frac{w_1\vec{X}_1(t)+w_2\vec{X}_2(t)+w_3\vec{X}_3(t)}{w_1+w_2+w_3} \tag{14}$$

## III. THE PROPOSED GWO-WD ALGORITHM

Although GWO has been improved by researchers from different institutions around the world, a better balance between exploration and exploitation is required to improve the optimization capability of the algorithm. To improve performance, a new GWO variant named GWO-WD is proposed and described in this section. The proposed GWO-WD algorithm is based on the following three basic concepts:

- A novel weighted distance is proposed;
- Based on the proposed weighted distance, a modified position-updating equation is established to improve the exploitation of the GWO algorithm; and
- A new elimination and repositioning strategy is proposed to enhance the exploration of the GWO algorithm.

### A. POSITION-UPDATING EQUATION BASED ON THE NOVEL WEIGHTED DISTANCE

As described in Eq. (3), the global exploration capacity of the GWO algorithm is controlled by the control parameter $\vec{C}$ [17], [19]. As shown in Eq. 5, the influence of $\vec{C}$ on the global exploration ability of the GWO algorithm is achieved mainly by controlling the global best individual (the $\alpha$ wolf), the global second-best individual (the $\beta$ wolf) and the global third-best individual (the $\delta$ wolf). The parameter $\vec{A}$ controls mainly the extent to which the $\omega$ wolves approximate the $\alpha$, $\beta$ and $\delta$ wolves. Therefore, in Eqs. (11)-(13), $\vec{A}$ is related to the weights of $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$, which introduces redundant information and diminishes the GWO performance. However, the weights of $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ in the position-updating equation of the GWO algorithm should be based only on the relevant coefficients of the three leader wolves. Based on this approach and the relevant works described in above subsection, this paper proposes a novel weighted distance as follows.

$$\vec{\varphi}_1 = \vec{C}_1 \frac{f_\alpha}{f_\alpha + f_\beta + f_\delta} \tag{15}$$

$$\vec{\varphi}_2 = \vec{C}_2 \frac{f_\beta}{f_\alpha + f_\beta + f_\delta} \tag{16}$$

$$\vec{\varphi}_3 = \vec{C}_3 \frac{f_\delta}{f_\alpha + f_\beta + f_\delta} \tag{17}$$

The position-updating equation based on the proposed weighted distance is modeled as follows.

$$\vec{X}_{11}(t+1) = \frac{\vec{\varphi}_1 \vec{X}_1(t) + \vec{\varphi}_2 \vec{X}_2(t) + \vec{\varphi}_3 \vec{X}_3(t)}{\vec{\varphi}_1 + \vec{\varphi}_2 + \vec{\varphi}_3} \tag{18}$$

where $\vec{\varphi}_1$, $\vec{\varphi}_2$ and $\vec{\varphi}_3$ are weight parameters and reflect the extents of influence of $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ on the position update, respectively.

### B. MODIFIED POSITION-UPDATING EQUATION

The position-updating equation of the GWO algorithm is determined simply based on the average location of the three leader wolves. Although this approach is effective for most normal problems, it has limited efficacy in high-dimensional, complex multimodal problems [2]. A well-designed position-updating equation with different weights in different iterations can better reflect the complex search process of the GWO algorithm. Therefore, the position-updating equation of the GWO algorithm should be modified as follows by adding useful information and combining the standard position-updating equation with weighted distances.

$$\vec{X}_{22}(t+1) = \frac{\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t)}{3} \tag{19}$$

$$\vec{X}(t+1) = \vec{X}_{22} + c_1 \mu_1 \left( \rho_1 \vec{X}_{11} - (1-\rho_1) \vec{X}_{22} \right) \tag{20}$$

$$\rho_1 = 1 - \frac{t}{MaxIter} \tag{21}$$

where $c_1$ is a constant between 0 and 2 (its value is 1.49445 in this paper); $\mu_1$ is a random number in the range of (0, 1); and $\rho_1$ is a weight used to reflect the influences of $\vec{X}_{11}$ and $\vec{X}_{22}$ at different iterations. In addition, the weight $\rho_1$ indicates that in the initial stage of the search, the roles of the three fittest wolves are almost the same, while in the later stage, the abilities of the three wolves vary.

### C. ELIMINATION AND REPOSITIONING STRATEGY

The worst individuals in stochastic population-based algorithms contribute little to the algorithm performance and may even weaken the performance because of their limited search capacity. This phenomenon also exists in the GWO algorithm. To improve the optimization performance of the GWO algorithm, an elimination and repositioning strategy, which is employed to eliminate several of the worst search agents and reinitialize them after a certain number of iterations, is proposed in this work.

In the GWO algorithm, the worst search agents and the elite search agents are obtained with the help of the three best search agents. If continual repositioning is adopted, positioning the worst search agents around the positions of the three best search agents may increase the probability of falling into local optima. Therefore, the worst search agents can be repositioned in promising parts of the search space far from the positions of the three best search agents. The expression used to reposition the worst candidates in promising areas is modeled as follows.

$$\vec{X}(t+1) = \mu_2 (ub - lb) - \mu_3 \rho_2 \frac{\vec{X}_\alpha(t) + \vec{X}_\beta(t) + \vec{X}_\delta(t)}{3} \tag{22}$$

$$\rho_2 = 1 - \left( \frac{t}{MaxIter} \right)^2 \tag{23}$$

where $\mu_2$ and $\mu_3$ are random numbers in the range of (0, 1), *ub* is the upper bound of the solution space, and *lb* is the lower bound of the solution space.

Note that in this work, we eliminated one-half of the worst individuals every five iterations and reinitialized them with Eqs. (22) -(23).

### D. COMPUTATIONAL COMPLEXITY OF GWO-WD

The computational complexity of the GWO, WDGWO, WdGWO, and GWO-WD algorithms is given as follows.

(1) In the initialization stage, GWO, WDGWO, WdGWO and GWO-WD require $O(N \times D)$ time, where $N$ denotes the population size and $D$ represents the size (dimension) of the problem.

(2) Calculating the control parameters of GWO, WDGWO, WdGWO and GWO-WD requires $O(N \times D)$ time.

(3) Updating the individuals in GWO, WDGWO, WdGWO and GWO-WD with the position-updating equation requires $O(N \times D)$ time.

(4) Evaluating the objective function fitness value of each search agent requires $O(N \times D)$ time.

(5) In GWO-WD, the elimination and repositioning strategy requires $O(N/2 \times D)$ time.

Based on this analysis, for each cycle of calculations, the total time (complexity) is $O(N \times D)$. After reaching the maximum number of iterations, the total time (complexity) of GWO, WDGWO, WdGWO and GWO-WD is O($N \times D \times MaxIter$), where $MaxIter$ denotes the maximum number of iterations.

## IV. RESULTS AND COMPARISON

### A. BENCHMARK TEST PROBLEMS AND PARAMETER SETTINGS

Three test series from several references [4], [19] were selected to evaluate the optimization performance of GWO-WD in comparison with that of the other algorithms. The first test series includes seven common unimodal functions ($f_1$-$f_7$) that have one global optimum and no local optima; therefore, this series is suitable for testing the exploitation ability of the chosen algorithms. The information for this series is listed in **Table 1**. The second test series consists of six classic multimodal functions ($f_8$-$f_{13}$) that have many local optima and are usually selected to verify the exploration and local minima avoidance capabilities of algorithms [32]. The information for this series is listed in **Table 2**.

The third test series includes ten fixed-dimension multimodal benchmark functions ($f_{14}$-$f_{23}$) that have fewer local optima than most multimodal problems and are thus useful for benchmarking both the exploration and the exploitation abilities of the algorithms. The information for this series is listed in **Table 3**. Note that in **Tables 1**, **2** and **3**, $f_{\min}$ indicates the global minimum value of the function, $D$ represents the dimension of the function, and *Range* represents the boundaries of the solution space. To perform an experiment with a fair comparison, the common parameters of the algorithms are listed in **Table 4**, and the other parameters are detailed in **Table 5**, where $N$ represents the population size and $R$ indicates the number of independent runs. The experiment was performed in MATLAB R2015a (MathWorks).

### B. COMPARISON WITH THE STANDARD GWO ALGORITHM

To investigate the optimization performance of the GWO-WD algorithm on the three types of benchmark test problems presented in **Tables 1** through **3**, the problems in **Tables 1** and **2** are tested for 30, 100 and 1000 dimensions. The 30D case is used to investigate the performance of the algorithm in solving low-dimensional problems, the 100D case is adopted to test the performance of the algorithm in solving medium-dimensional problems, and the 1000D case is employed to verify the optimization performance of the algorithm in solving challenging, large-scale problems. We compared the best (Best), average (Mean), worst (Worst) and standard deviation (St. dev.) results of the GWO-WD algorithm and the standard GWO algorithm after executing 30 independent experiments. The experimental results for the unimodal and multimodal problems are shown in **Table 6**, and those for the fixed-dimension multimodal problems are listed in **Table 7**.

As shown in **Table 6**, the GWO-WD algorithm yielded the best results in 6 out of 7 unimodal test problems from the low -dimension to the large-scale problems. For the test function $f_6$, a low- to medium-dimension problem, the GWO algorithm achieved the best results, but the GWO-WD algorithm provided better results for the high-dimensional problems. Among the 6 multimodal test functions, the GWO-WD algorithm provided better results than the standard GWO algorithm for 5 functions ($f_8$-$f_{11}$ and $f_{13}$). However, for test problem $f_{12}$, the standard GWO algorithm produced better results than the GWO-WD algorithm only in the low-dimensional case. From **Table 7**, the GWO-WD algorithm achieved better results for 7 test functions ($f_{14}$-$f_{17}$ and $f_{21}$-$f_{23}$) and similar results for one test function ($f_{18}$). Moreover, compared with the GWO-WD algorithm, GWO yielded better results for two test functions ($f_{19}$ and $f_{20}$). From the optimization results of the GWO-WD and standard GWO algorithms for 13 low-, medium- and high-dimensional test functions, the GWO-WD algorithm yielded better results than the standard GWO algorithm in most cases. Overall, the optimization results obtained by the GWO-WD algorithm for low-, medium- and high-dimensional test functions indicate that the GWO-WD performance deteriorates less as the problem dimension increases drastically. In other words, the GWO-WD algorithm displays excellent scalability considering the search dimension of complex problems.

To compare the performance of the GWO and GWO-WD algorithms on the basis of the statistical results, **Table 8** presents the results of a Wilcoxon rank -sum test with a significance level of 0.05. According to **Table 8**, the GWO-WD algorithm yielded the best results in 39 out of 49 cases, equal results in 6 out of 49 cases, and worse results in 4 out of 49 cases, compared to the standard GWO algorithm. These statistical results verify that the performance of the standard GWO algorithm has been considered improved by the GWO-WD algorithm for unimodal and multimodal test problems and that the two algorithms achieved similar performance for fixed-dimension test problems.

For an intuitive illustration, the convergence curves of the standard GWO and GWO-WD algorithms for three typical unimodal and three representative multimodal test functions with D=30, 100 and 1000 and four classic fixed-dimension multimodal benchmark functions are plotted in **Figure 1**. As displayed in **Figure 1**, GWO-WD yielded a faster convergence speed than the standard GWO algorithm in all test problems regardless of dimension, including for the fixed-dimension multimodal problems. In addition, the GWO-WD algorithm provided relatively similar convergence rates for the problems as the dimension changed. Furthermore, for function $f_9$, the GWO-WD algorithm exhibited a faster convergence rate in the high-dimensional case than in the low-dimensional case. This phenomenon indicates that the GWO-WD algorithm is a robust algorithm.

In addition to investigating the solution quality and convergence speed of the standard GWO and GWO-WD algorithms, the computational times should also be compared [41].

**TABLE 1.** Descriptions of the seven unimodal benchmark test functions.

| Function | D | Range | $f_{min}$ |
|---|---|---|---|
| $f_1(x)=\sum_{i=1}^{n} x_i^2$ | 30, 100, 1000 | [-100,100] | 0 |
| $f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$ | 30, 100, 1000 | [-10,10] | 0 |
| $f_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$ | 30, 100, 1000 | [-100,100] | 0 |
| $f_4(x)=\max\left\{\lvert x_i\rvert, 1\le x_i\le n\right\}$ | 30, 100, 1000 | [-100,100] | 0 |
| $f_5(x)=\sum_{i=1}^{n}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$ | 30, 100, 1000 | [-30,30] | 0 |
| $f_6(x)=\sum_{i=1}^{n}\left(\lvert x_i+0.5\rvert\right)^2$ | 30, 100, 1000 | [-100,100] | 0 |
| $f_7(x)=\sum_{i=1}^{n} ix_i^4+random\left[0,1\right)$ | 30, 100, 1000 | [-1.28,1.28] | 0 |

**TABLE 2.** Descriptions of the six multimodal benchmark test functions.

| Function | D | Range | $f_{min}$ |
|---|---|---|---|
| $f_8(x)=\sum_{i=1}^{n} -x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$ | 30, 100, 1000 | [-500, 500] | 0 |
| $f_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$ | 30, 100, 1000 | [-5.12,5.12] | 0 |
| $f_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$ | 30, 100, 1000 | [-32, 32] | 0 |
| $f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$ | 30, 100, 1000 | [-600, 600] | 0 |
| $f_{12}(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})+(y_n-1)^2\right]\right\}$ $+\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{x_i+1}{4},\ u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\ 0, & -a<x_i<a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$ | 30, 100, 1000 | [-50, 50] | 0 |
| $f_{13}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]\right.$ $\left.+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}+\sum_{i=1}^{n}u(x_i,5,100,4)$ | 30, 100, 1000 | [-50, 50] | 0 |

Therefore, the CPU run times of GWO-WD and GWO were compared on the same machine under the same conditions. **Table 9** lists the results (in seconds) of the GWO-WD and GWO algorithms for 13 test functions with D=30, 100 and 1000 and for 14 fixed-dimension multimodal test functions. As shown in this table, for functions $f_1$, $f_{14}$-$f_{15}$, $f_{18}$ and $f_{21}$-$f_{23}$, the GWO-WD algorithm yielded shorter run times than the GWO algorithm. For function $f_2$, the GWO algorithm had shorter run times than the GWO-WD algorithm for medium- and high- dimension problems but longer run times for low-dimension cases. For the remaining test problems, the GWO-WD and GWO algorithms provided very similar computational run times. In summary, the GWO-WD algorithm improves the optimization performance of the GWO algorithm but does not increase the CPU run time.

### C. COMPARISON WITH THE GWO VARIANTS

To further investigate the excellent performance of GWO-WD, we compared its optimization results with those of three GWO variants (mGWO [31], WDGWO [1] and

**TABLE 3.** Descriptions of the ten fixed-dimension multimodal benchmark test functions.

| Function | D | Range | $f_{\min}$ |
|---|---|---|---|
| $f_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$ | 2 | [-65,65] | 1 |
| $f_{15}(x)=\sum_{i=1}^{n}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$ | 4 | [-5,5] | 0.00030 |
| $f_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$ | 2 | [-5,5] | -1.0316 |
| $f_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$ | 2 | [-5,5] | 0.398 |
| $f_{18}(x)=\left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right]\times \left[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\right]$ | 2 | [-2,2] | 3 |
| $f_{19}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right)$ | 3 | [1,3] | -3.86 |
| $f_{20}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right)$ | 6 | [0,1] | -3.32 |
| $f_{21}(x)=-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$ | 4 | [0,10] | -10.1532 |
| $f_{22}(x)=-\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$ | 4 | [0,10] | -10.4028 |
| $f_{23}(x)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$ | 4 | [0,10] | -10.5363 |

**TABLE 4.** Experimental parameter settings for GWO-WD and the other selected algorithms.

| Algorithm | Parameter | | | | | |
|---|---|---|---|---|---|---|
| | *D* | | | *N* | *MaxIter* | *R* |
| GWO | 30 | 100 | 1000 | 30 | 500 | 30 |
| mGWO | 30 | 100 | 1000 | 30 | 500 | 30 |
| WDGWO | 30 | 100 | 1000 | 30 | 500 | 30 |
| WdGWO | 30 | 100 | 1000 | 30 | 500 | 30 |
| WOA | 30 | 100 | 1000 | 30 | 500 | 30 |
| MFO | 30 | 100 | 1000 | 30 | 500 | 30 |
| GSA | 30 | 100 | 1000 | 30 | 500 | 30 |
| PSO | 30 | 100 | 1000 | 30 | 500 | 30 |
| CLSPO | 30 | 100 | 1000 | 30 | 500 | 30 |
| GWO-WD | 30 | 100 | 1000 | 30 | 500 | 30 |

WdGWO [2]) on the 23 benchmark test problems described in **Tables 1-3**. The parameters of mGWO, WDGWO and WdGWO were set as shown in **Tables 4** and **5**.

Each algorithm was executed 30 times independently during the experiments for each test problem. To evaluate the performance, the mean (Mean) and standard deviation (St. dev) of the fitness values were employed as criteria for experimental validation. The experimental results are reported in **Tables 10-11**.

From **Table 10**, for the 7 unimodal test functions, WDGWO obtained the best results for 5 functions ($f_1$-$f_4$ and $f_7$) with D=30, and GWO-WD yielded the second-best

**TABLE 5.** Experimental parameter settings for GWO-WD and the other selected algorithms.

| Algorithm | Parameters | Values |
|---|---|---|
| GWO, mGWO | $\vec{a}$ | Linearly decreased from 2 to 0 |
| WDGWO, WdGWO | Stopping criteria | Maximum iteration |
| WOA | $\vec{a}_1$ | Linearly decreased from 2 to 0 |
| | $\vec{a}_2$ | Linearly increased from -1 to 1 |
| | The shape of the logarithmic spiral (b) | 1 |
| | Stopping criteria | Maximum iteration |
| MFO | $\vec{a}$ | Linearly decreased from -1 to -2 |
| | The shape of the logarithmic spiral (b) | 1 |
| | Stopping criteria | Maximum iteration |
| GSA | Descending coefficient ($\alpha$) | 20 |
| | The gravitational constant initial value ( $G_0$ ) | 100 |
| | The constant value ( $\varepsilon$ ) | $10^{-100}$ |
| | Stopping criteria | Maximum iteration |
| PSO | Inertia weight ( $w$ ) | Linearly decreased from 0.9 to 0.4 |
| | Stopping criteria | Maximum iteration |
| CLPSO | Inertia weight ( $w$ ) | Linearly decreased from 0.9 to 0.2 |
| | Acceleration coefficient ( $c$ ) | 1.49445 |
| | Refreshing gap ( $m$ ) | 5 |

results for the same functions. In addition, GWO-WD achieved the best result for function $f_5$ with D=30, and the third-best result for function $f_6$ with D=30. For the 6 multimodal test functions with D=30, GWO-WD provided the best results for all functions except $f_{10}$, and GWO-WD obtained the second-best Mean and St. dev results for this function. WDGWO produced results similar to GWO-WD for two functions, namely, $f_9$ and $f_{11}$. For the 13 unimodal and multimodal test functions with D=100, WDGWO achieved the best results for all functions except $f_5$ and $f_6$. GWO-WD obtained the best results for two functions ($f_5$ and $f_6$) and the second-best results for 5 functions ($f_1$-$f_4$ and $f_7$). GWO-WD provided the best results for 5 out of 6 multimodal test functions, and the results were comparable to those of WDGWO for function $f_{10}$. For the 13 large-scale unimodal and multimodal test problems with D=1000, WDGWO obtained the best results for 5 functions ($f_1$, $f_3$-$f_4$, $f_7$ and $f_{10}$), and GWO-WD achieved the best and second-best results for 8 functions ($f_2$, $f_5$-$f_6$, $f_8$-$f_9$ and $f_{11}$-$f_{13}$) and 6 functions ($f_1$, $f_3$-$f_4$, $f_7$ and $f_{10}$), respectively. The results presented above indicate that the WDGWO algorithm performs better than the GWO-WD algorithm for unimodal problems, but the GWO-WD algorithm is superior on multimodal problems. It should be noted that the WDGWO algorithm experienced a dimensional disaster and failed for function $f_2$ with D=1000; this phenomenon indicates that the WDGWO algorithm may suffer from the same issue as the dimension of unimodal problems continues to increase. Therefore, the GWO-WD algorithm can obtain results that are better than or very similar to those of the WDGWO algorithm for large-scale complex unimodal problems.

**Table 11** presents the results for the 10 fixed-dimension multimodal benchmark functions. As shown in this table, the GWO-WD algorithm obtained the best results for 5 functions ($f_{15}$, $f_{17}$ and $f_{21}$-$f_{23}$) and similar results to the mGWO and WdGWO algorithms for two functions: $f_{16}$ and $f_{18}$. The mGWO algorithm achieved the best results for functions $f_{19}$ and $f_{20}$, and the GWO-WD algorithm provided the third-best results for the same two functions. In addition, the WDGWO algorithm achieved the worst results for all 10 fixed-dimension multimodal benchmark functions.

**Figures 2** to **5** plot the convergence curves of mGWO, WdGWO, WDGWO and GWO-WD. **Figures 2** through **4** demonstrate that the WDGWO algorithm displayed the fastest convergence speed for 6 functions ($f_1$, $f_3$-$f_4$, $f_7$, $f_9$ and $f_{11}$), while the GWO-WD algorithm exhibited a similar convergence speed among the same 6 test functions and was faster than mGWO and WdGWO. GWO-WD displayed the fastest convergence speed among the remaining 4 functions ($f_5$, $f_8$ and $f_{12}$-$f_{13}$). As shown in **Figure 5**, GWO-WD exhibited the fastest convergence speed for 9 out of 10 fixed-dimension multimodal benchmark functions and the third-fastest convergence speed for function $f_{20}$. The above analysis, confirms that the GWO-WD algorithm yielded good convergence speeds for 23 classic benchmark test functions.

To compare the performance between GWO-WD and the three GWO variants from the statistical results, **Table 12** summarizes the Wilcoxon rank -sum test results with a significance level of 0.05. The GWO-WD algorithm provided better results in 42 out of 49, 28 out of 49, and 41 out of 49 cases compared to the mGWO, WDGWO, and WdGWO algorithms, respectively.

The GWO-WI algorithm, which was presented by Jitkongchuen *et al.* [1], is another GWO variant. To compare the performance of the GWO-WD algorithm with the GWO-WI and GWO algorithms, nine functions were selected from the literature [1] as the benchmark set; detailed

**TABLE 6.** Comparisons between GWO-WD and GWO based on the 13 unimodal and multimodal benchmark test functions.

| Function | D | GWO | | | | GWO-WD | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Best | Mean | Worst | St.dev | Best | Mean | Worst | St.dev |
| $f_1$ | 30 | 1.81E-29 | 1.17E-27 | 9.32E-27 | 1.95E-27 | 6.50E-145 | 1.34E-140 | 1.24E-139 | 2.71E-140 |
| | 100 | 3.81E-13 | 1.52E-12 | 4.04E-12 | 1.05E-12 | 3.18E-133 | 5.59E-130 | 4.42E-129 | 1.00E-129 |
| | 1000 | 1.32E-01 | 2.42E-01 | 4.49E-01 | 7.61E-02 | 3.30E-121 | 2.13E-119 | 1.56E-118 | 3.53E-119 |
| $f_2$ | 30 | 1.56E-17 | 7.77E-17 | 2.45E-16 | 5.05E-17 | 9.68E-75 | 1.64E-73 | 1.10E-72 | 2.15E-73 |
| | 100 | 1.95E-08 | 4.52E-08 | 8.79E-08 | 1.78E-08 | 1.26E-68 | 3.49E-67 | 2.42E-66 | 4.36E-67 |
| | 1000 | 2.49E-01 | 6.08E-01 | 1.50E+00 | 2.49E-01 | 8.39E-63 | 2.42E-61 | 7.87E-61 | 2.26E-61 |
| $f_3$ | 30 | 8.14E-09 | 6.39E-06 | 8.99E-05 | 1.70E-05 | 7.66E-130 | 6.67E-123 | 7.23E-122 | 1.83E-122 |
| | 100 | 3.56E+01 | 5.97E+02 | 1.81E+03 | 4.60E+02 | 1.77E-120 | 1.02E-115 | 1.80E-114 | 3.39E-115 |
| | 1000 | 1.02E+06 | 1.49E+06 | 2.11E+06 | 2.91E+05 | 2.40E-111 | 5.85E-107 | 6.69E-106 | 1.65E-106 |
| $f_4$ | 30 | 4.71E-08 | 9.46E-07 | 4.50E-06 | 9.43E-07 | 1.08E-69 | 3.96E-67 | 3.59E-66 | 7.29E-67 |
| | 100 | 1.69E-01 | 9.16E-01 | 4.33E+00 | 8.56E-01 | 9.03E-65 | 3.43E-62 | 6.86E-61 | 1.25E-61 |
| | 1000 | 7.04E+01 | 7.83E+01 | 8.71E+01 | 3.45E+00 | 9.57E-59 | 5.23E-54 | 1.30E-52 | 2.38E-53 |
| $f_5$ | 30 | 2.60E+01 | 2.70E+01 | 2.88E+01 | 7.85E-01 | 4.35E-04 | 2.74E+00 | 2.89E+01 | 5.64E+00 |
| | 100 | 9.60E+01 | 9.78E+01 | 9.85E+01 | 7.41E-01 | 2.01E-02 | 1.02E+01 | 8.18E+01 | 2.13E+01 |
| | 1000 | 1.02E+03 | 1.06E+03 | 1.12E+03 | 2.47E+01 | 1.37E-01 | 7.70E+01 | 2.82E+02 | 8.74E+01 |
| $f_6$ | 30 | 2.56E-01 | 8.25E-01 | 1.77E+00 | 3.76E-01 | 2.22E-01 | 1.75E+00 | 4.40E+00 | 1.35E+00 |
| | 100 | 7.78E+00 | 9.87E+00 | 1.20E+01 | 1.02E+00 | 2.54E-01 | 1.03E+01 | 2.18E+01 | 8.17E+00 |
| | 1000 | 1.99E+02 | 2.03E+02 | 2.09E+02 | 2.31E+00 | 3.67E-01 | 9.85E+01 | 2.48E+02 | 8.10E+01 |
| $f_7$ | 30 | 6.94E-04 | 2.14E-03 | 5.99E-03 | 1.24E-03 | 1.68E-05 | 1.86E-04 | 6.93E-04 | 1.86E-04 |
| | 100 | 3.04E-03 | 6.78E-03 | 1.67E-02 | 3.20E-03 | 1.88E-05 | 3.42E-04 | 9.78E-04 | 2.59E-04 |
| | 1000 | 1.12E-01 | 1.51E-01 | 2.20E-01 | 2.82E-02 | 1.98E-05 | 3.06E-04 | 9.73E-04 | 2.79E-04 |
| $f_8$ | 30 | -7.40E+03 | -6.03E+03 | -2.96E+03 | 8.96E+02 | -2.99E+04 | -2.17E+04 | -2.14E+04 | 1.53E+03 |
| | 100 | -2.04E+04 | -1.61E+04 | -5.75E+03 | 2.99E+03 | -7.15E+04 | -7.15E+04 | -7.15E+04 | 4.26E+00 |
| | 1000 | -1.10E+05 | -8.82E+04 | -2.17E+04 | 1.45E+04 | -7.41E+05 | -7.16E+05 | -7.15E+05 | 4.77E+03 |
| $f_9$ | 30 | 0.00E+00 | 1.95E+00 | 9.80E+00 | 3.24E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 100 | 3.43E-09 | 9.33E+00 | 2.54E+01 | 6.21E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 1000 | 1.26E+02 | 1.85E+02 | 2.55E+02 | 3.62E+01 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| $f_{10}$ | 30 | 7.90E-14 | 1.05E-13 | 1.71E-13 | 2.00E-14 | 4.44E-15 | 4.44E-15 | 4.44E-15 | 0.00E+00 |
| | 100 | 6.09E-08 | 1.22E-07 | 2.46E-07 | 4.58E-08 | 4.44E-15 | 4.56E-15 | 7.99E-15 | 6.49E-16 |
| | 1000 | 1.54E-02 | 1.82E-02 | 2.45E-02 | 2.42E-03 | 4.44E-15 | 5.39E-15 | 7.99E-15 | 1.60E-15 |
| $f_{11}$ | 30 | 0.00E+00 | 3.95E-03 | 2.98E-02 | 8.33E-03 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 100 | 2.26E-13 | 4.88E-03 | 3.15E-02 | 1.03E-02 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 1000 | 9.18E-03 | 5.73E-02 | 2.44E-01 | 8.23E-02 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| $f_{12}$ | 30 | 6.62E-03 | 4.24E-02 | 1.32E-01 | 2.25E-02 | 4.74E-05 | 8.93E-02 | 4.76E-01 | 1.06E-01 |
| | 100 | 1.96E-01 | 3.04E-01 | 4.58E-01 | 7.80E-02 | 4.24E-03 | 2.16E-01 | 1.87E+00 | 3.84E-01 |
| | 1000 | 8.91E-01 | 1.23E+00 | 2.10E+00 | 2.94E-01 | 4.00E-04 | 2.81E-01 | 1.14E+00 | 3.16E-01 |
| $f_{13}$ | 30 | 1.07E-01 | 6.10E-01 | 1.14E+00 | 2.34E-01 | 2.55E-05 | 2.63E-02 | 1.52E-01 | 4.24E-02 |
| | 100 | 5.85E+00 | 6.63E+00 | 7.79E+00 | 4.89E-01 | 3.21E-03 | 3.56E-02 | 1.82E-01 | 4.78E-02 |
| | 1000 | 1.11E+02 | 1.22E+02 | 1.43E+02 | 8.26E+00 | 9.10E-04 | 6.75E-01 | 7.42E+00 | 1.56E+00 |

**TABLE 7.** Comparisons between GWO-WD and GWO based on the 10 fixed-dimension multimodal benchmark test functions.

| Function | D | GWO | | | | GWO-WD | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Best | Mean | Worst | St.dev | Best | Mean | Worst | St.dev |
| $f_{14}$ | 2 | 9.98E-01 | 2.45E+00 | 1.27E+01 | 2.18E+00 | 9.98E-01 | 1.26E+00 | 2.98E+00 | 6.86E-01 |
| $f_{15}$ | 4 | 3.08E-04 | 3.15E-03 | 2.04E-02 | 6.87E-03 | 3.37E-04 | 6.73E-04 | 1.38E-03 | 3.40E-04 |
| $f_{16}$ | 2 | -1.0316 | -1.0316 | -1.0316 | 3.25E-08 | -1.0316 | -1.0316 | -1.0316 | 1.38E-05 |
| $f_{17}$ | 2 | 0.3979 | 0.3979 | 0.3979 | 3.98E-06 | 0.3979 | 0.3989 | 0.4029 | 1.15E-03 |
| $f_{18}$ | 2 | 3 | 3 | 3 | 2.68E-05 | 3 | 3 | 3 | 4.78E-05 |
| $f_{19}$ | 3 | -3.8628 | -3.8615 | -3.8550 | 1.92E-03 | -3.8625 | -3.8605 | -3.8509 | 2.35E-03 |
| $f_{20}$ | 6 | -3.3220 | -3.2759 | -3.1416 | 6.26E-02 | -3.3052 | -3.1795 | -3.1145 | 3.71E-02 |
| $f_{21}$ | 4 | -10.1531 | -9.5637 | -2.6303 | 1.83E+00 | -10.1532 | -10.1488 | -10.1235 | 6.70E-03 |
| $f_{22}$ | 4 | -10.4024 | -10.2237 | -5.0876 | 9.70E-01 | -10.4028 | -10.3946 | -10.3153 | 1.67E-02 |
| $f_{23}$ | 4 | -10.5361 | -10.3544 | -5.1284 | 9.87E-01 | -10.5363 | -10.5292 | -10.4869 | 1.24E-02 |

descriptions of these functions can be found in Ref. [1]. The population size and maximum number of iterations for each test function were set as in Ref. [1], and each experiment was run independently 10 times. The experimental results are shown in Table 13, where the GWO-WI results were obtained directly from the results reported in [1].

The experimental results reveal that both the proposed GWO-WD algorithm and the GWO-WI algorithm achieved a

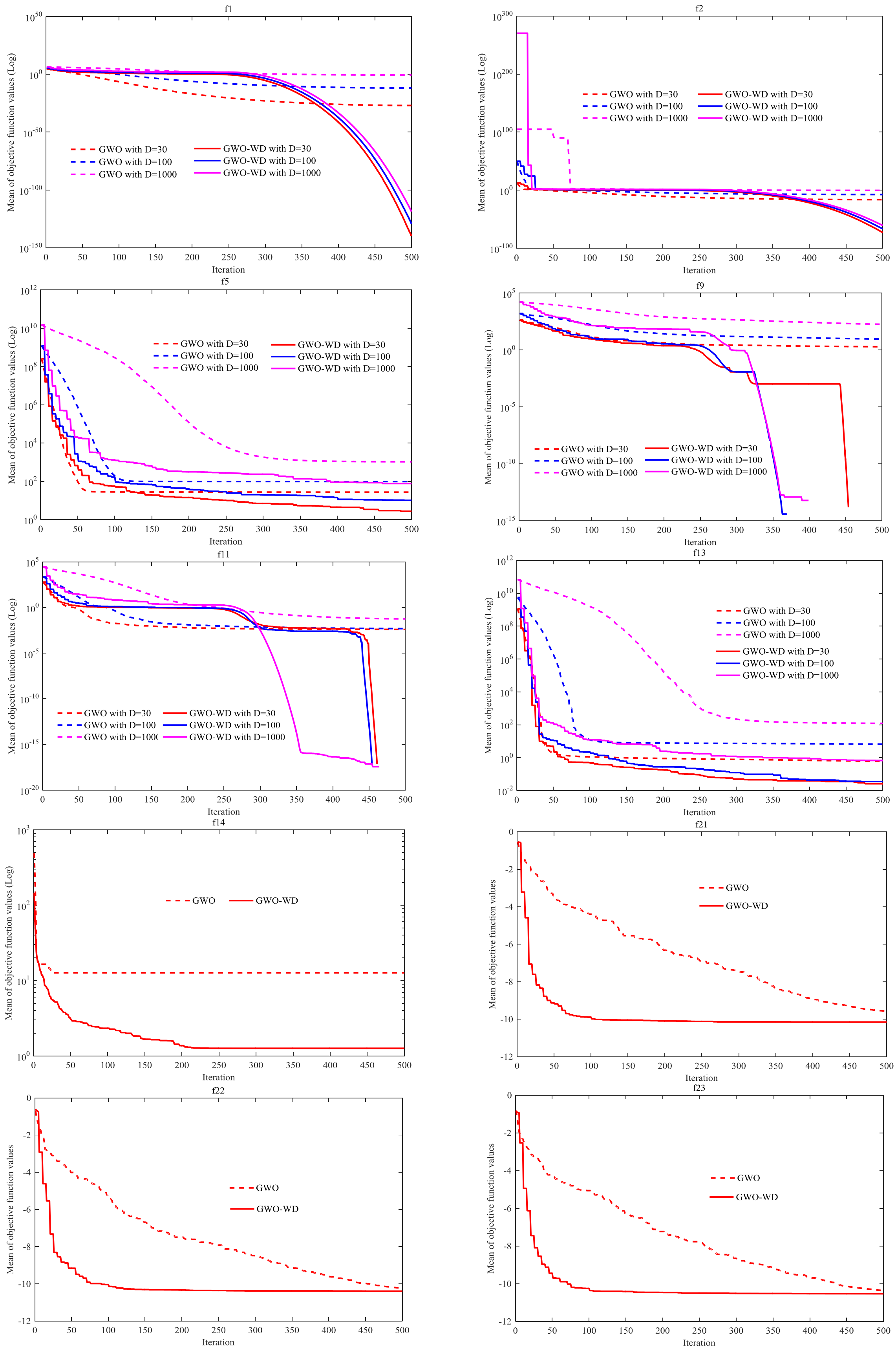



**FIGURE 1.** **Convergence curves of GWO and GWO-WD based on several typical functions.**

**TABLE 8.** Summarized Wilcoxon rank -sum test results between GWO and GWO-WD.

| GWO vs | GWO-WD | | |
|---|---|---|---|
| | Worse | Equal | Better |
| D=30 | 1 | 1 | 11 |
| D=100 | 0 | 1 | 12 |
| D=1000 | 0 | 0 | 13 |
| Fixed-dimension | 3 | 4 | 3 |
| Total | 4 | 6 | 39 |

minimum of zero for functions $F_1$-$F_4$ and $F_9$. For function $F_6$, GWO-WI obtained the best value, whereas GWO-WD achieved the second best value. GWO-WI and GWO-WD acquired almost the same result on function $F_7$. In addition, GWO-WD obtained the best results on functions $F_5$ and $F_8$. From the above optimization results, the optimization performance of the proposed GWO-WD algorithm is better than that of GWO-WI and GWO.

### D. COMPARISON WITH OTHER STATE-OF-THE-ART ALGORITHMS

In this subsection, we compared the GWO-WD algorithm to several state-of-the-art algorithms, such as the PSO algorithm [4], the comprehensive learning particle swarm optimizer (CLPSO) [42], the WOA [12], the MFO algorithm [13] and the gravitational search algorithm (GSA) [43]. All of the common parameters of PSO, CLPSO, WOA, MFO, GSA and GWO-WD were set as in **Tables 4** and **5**. The other algorithm parameter values in the five state-of-the-art algorithms were taken directly from the original papers in which they were presented. The common parameters were set as follows: the population size was 30, the maximum number of iterations was 500, and the dimension of each problem was 30. Each algorithm was independently run 30 times for each benchmark test problem. The experimental results are summarized in **Table 14**, where "Mean" and "St. dev." indicate the mean and standard deviation of the fitness values, respectively.

As shown in **Table 14**, for the seven unimodal problems, the GWO-WD algorithm achieved the best results for all test functions except $f_6$. For function $f_6$, the GSA algorithm obtained the best results the WOA obtained the second-best results, and the GWO-WD algorithm provided the third-best results, outperforming the CLPSO, MFO and PSO algorithms. For the six multimodal problems, GWO-WD provided the best results for four functions ($f_7$-$f_9$ and $f_{11}$). For function $f_{10}$, the WOA obtained the best results, and the GWO-WD algorithm ranked second and very close to the WOA. For functions $f_{12}$ and $f_{13}$, the WOA achieved the best results, with PSO ranking second and GWO-WD ranking third. In addition, the results of GWO-WD were very similar to those of WOA and PSO for the same two functions. For the ten fixed-dimension multimodal problems, the GWO-WD algorithm achieved the best results for five functions ($f_{15}$, $f_{17}$ and $f_{21}$-$f_{23}$). For two functions, namely, $f_{16}$ and $f_{18}$, GWO-WD obtained "Mean" results similar to those of the other five algorithms and the worst "St.dev.". For function $f_{14}$, the CLPSO algorithm provided the best results, and GWO-WD obtained the second-best results. For function $f_{19}$, the CLPSO algorithm, GSA and PSO algorithm produced the best results, and the GWO-WD algorithm performed better than the WOA. The GSA achieved the best results for function $f_{20}$, and the GWO-WD algorithm provided the worst results for these functions. From the optimization results presented above, we can deduce that the comprehensive optimization performance of the GWO-WD algorithm is the best among the algorithms compared.

The Wilcoxon rank -sum test results with a significance level of 0.05 are listed in **Table 15**. This table indicates that the GWO-WD algorithm yielded better results in 17 out of 23, 19 out of 23, 20 out of 23, 17 out of 23, and 13 out of 23 cases compared to the CLPSO algorithm, GSA, MFO algorithm, PSO algorithm, and WOA, respectively.

The convergence curves of the mean fitness values derived from GWO-WD and the other five comparison algorithms are plotted in **Figure 6** for 10 typical test functions with a dimension of 30. As shown in **Figure 6**, the GWO-WD algorithm has a faster convergence rate than the other algorithms for all test functions except function $f_9$.

To further investigate the performance of GWO-WD, the IEEE CEC2014 benchmark series, which includes 30 test functions that are more challenging than the 23 classic test functions recorded in **Tables 1**through **3**, was adopted to execute several independent experiments. The CEC2014 benchmark functions can be divided into four categories: a) unimodal test functions (F1-F3); b) multimodal test functions (F4-F16); c) hybrid test functions (F17-F22); and composite test functions (F23-F30). Detailed information on this benchmark data can be found in [44]. The search range of these 30 test functions is [−100,100], and the dimension was set to 30. We compared the GWO-WD algorithm to the improved PSO with time-varying accelerator coefficients (IPSO) [45], the modified PSO with adaptive acceleration coefficients (TACPSO) [46], the dynamically dimensioned search grey wolf optimizer (DGWO) [47], the GWO algorithm [17] and the differential evolution (DE) algorithm [48]. The population size was set to 30, and the maximum number of iterations was set to 5000. Each test function was executed independently 30 times. The mean (Mean) and standard deviation (St. dev) of the function values are recorded in **Table 16**. Note that the experimental results of IPSO, TACPSO, DGWO and GWO for the CEC2014 benchmark data are taken directly from the literature [47].

As shown in **Table 16**, the comprehensive performance of GWO-WD is very competitive with that of DGWO. Compared with IPSO, GWO-WD obtained better results for 16 test functions and similar results for 3 test functions (F5 and F12-F13). Compared with TACPSO, GWO-WD achieved better results for 20 test functions and similar and worse results for 2 functions (F13 and F16) and 8 test functions (F5-F6, F8-F11, F17 and F21), respectively. With respect to DGWO, GWO-WD produced better and similar results for 8 test functions (F2, F15, F20 and F25-F29) and 6 test

**TABLE 9.** Mean CPU run time comparison between GWO-WD and GWO based on the 23 functions.

| Function | GWO (D=30) | GWO-WD (D=30) | GWO (D=100) | GWO-WD (D=100) | GWO (D=1000) | GWO-WD (D=1000) |
|---|---|---|---|---|---|---|
| $f_1$ | 1.794 | 1.307 | 2.178 | 2.095 | 12.486 | 12.474 |
| $f_2$ | 1.440 | 1.287 | 1.957 | 2.203 | 10.813 | 12.781 |
| $f_3$ | 3.477 | 3.994 | 10.334 | 10.037 | 137.337 | 139.237 |
| $f_4$ | 1.118 | 1.191 | 1.916 | 2.088 | 10.657 | 12.334 |
| $f_5$ | 1.213 | 1.316 | 2.131 | 2.187 | 11.002 | 12.596 |
| $f_6$ | 1.228 | 1.269 | 1.829 | 2.113 | 10.654 | 12.609 |
| $f_7$ | 1.465 | 1.527 | 2.800 | 2.798 | 16.644 | 17.068 |
| $f_8$ | 1.236 | 1.413 | 2.187 | 2.364 | 14.114 | 13.953 |
| $f_9$ | 1.142 | 1.259 | 1.958 | 2.217 | 11.426 | 13.286 |
| $f_{10}$ | 1.200 | 1.327 | 2.046 | 2.358 | 11.654 | 13.460 |
| $f_{11}$ | 1.299 | 1.423 | 2.154 | 2.430 | 11.996 | 14.091 |
| $f_{12}$ | 2.378 | 2.282 | 3.756 | 3.992 | 20.310 | 21.551 |
| $f_{13}$ | 2.163 | 2.235 | 3.710 | 3.935 | 20.344 | 21.704 |
| Function | GWO | | | GWO-WD | | |
| $f_{14}$ | 5.082 | | | 4.934 | | |
| $f_{15}$ | 0.904 | | | 0.882 | | |
| $f_{16}$ | 0.690 | | | 0.692 | | |
| $f_{17}$ | 0.656 | | | 0.711 | | |
| $f_{18}$ | 0.707 | | | 0.670 | | |
| $f_{19}$ | 1.095 | | | 1.098 | | |
| $f_{20}$ | 1.108 | | | 1.144 | | |
| $f_{21}$ | 1.688 | | | 1.490 | | |
| $f_{22}$ | 1.846 | | | 1.806 | | |
| $f_{23}$ | 2.281 | | | 2.254 | | |

functions (F5, F12-F14, F16 and F24), respectively, and for the remaining functions, GWO-WD provided results that were very similar to those obtained by DGWO. Compared with the GWO algorithm, GWO-WD achieved better and similar results for 14 and 5 test functions (F12-F14, F16, and F24), respectively. When compared with the DE algorithm, GWO-WD obtained better results for 14 functions, worse results for 13 functions and equal results for 3 functions.

The Wilcoxon rank -sum test results with a significance level of 0.05 are recorded in **Table 17**. From this table, the GWO-WD algorithm obtained better results in 13 out of 30, 20 out of 30, 11 out of 30, 17 out of 30, and 14 out of 30 cases compared to the IPSO, TACPSO, DGWO, GWO, and DE algorithms, respectively.

### E. PERFORMANCE INVESTIGATION FOR THE TWO COMPONENTS IN GWO-WD

As noted above, the GWO-WD algorithm consists of two main components: the modified position-updating equation based on the novel weighted distance and the elimination and repositioning strategy. The objective of this subsection is to analyze the effects of these two components to improve the performance of the GWO-WD algorithm. In this experiment, two additional experiments were performed for 7 unimodal and 6 multimodal benchmark test functions with a dimension of 30 and 10 fixed-dimension multimodal benchmark test functions. In the first experiment, GWO is changed only by using the modified position-updating equation based on the novel weighted distance (i.e., Eq. (20)), and the elimination and repositioning strategy is ignored (referred to as GWO-WD1). In another experiment, GWO-WD adopts only the elimination and repositioning strategy (i.e., Eq. (22)), and the modified position-updating equation based on the novel weighted distance is not used (referred to as GWO-WD2). In the two experiments, 30 independent runs were conducted for each test function, and the maximum number of objective function evaluations was set to 15000 (i.e., the population size and the maximum number of iterations were set to 30 and 500, respectively). The experimental results of GWO-WD1, GWO-WD2, and GWO-WD are summarized in **Table 18**, in which the Wilcoxon rank-sum test results with a significance level of 0.05 are also recorded.

As shown in **Table 18**, for the 7 unimodal test functions, GWO-WD1 produced equal results in 5 cases compared with the proposed algorithm; additionally, GWO-WD2 provided worse results in 5 cases, better results in 1 case and equal results in 1 case. For the 16 multimodal and fixed-dimension multimodal test functions, GWO-WD1 yielded equal results in 8 cases and worse results in 8 cases, and GWO-WD2 provided better results in 4 cases, equal results in 8 cases, and worse results in 4 cases. From this analysis, we can conclude that GWO-WD1 has a better exploitation performance than GWO-WD2, and the exploitation performance is similar to that of GWO-WD. However, GWO-WD2 has a better exploration ability than GWO-WD1 and a similar ability to GWO-WD. Therefore, the modified position-updating equation based on a novel weighted distance is effective for improving the exploitation ability of the GWO algorithm, and the elimination and repositioning strategy is useful for enhancing the exploration performance of the

**TABLE 10. Comparisons between GWO-WD and three GWO variants based on 13 benchmark test functions with D=30, 100 and 1000.**

| Function | mGWO (D=30) (Mean±St. dev) | WdGWO (D=30) (Mean±St. dev) | WDGWO (D=30) (Mean±St. dev) | GWO-WD (D=30) (Mean±St. dev) |
|---|---|---|---|---|
| $f_1$ | 2.45E-36±3.77E-36 | 2.89E-21±6.91E-21 | 0.00E+00±0.00E+00 | 1.34E-140±2.71E-140 |
| $f_2$ | 1.04E-21±1.06E-21 | 2.50E-13±1.92E-13 | 6.59E-254±0.00E+00 | 1.64E-73±2.15E-73 |
| $f_3$ | 8.88E-08±2.26E-07 | 3.63E-03±5.71E-03 | 0.00E+00±0.00E+00 | 6.67E-123±1.83E-122 |
| $f_4$ | 9.35E-10±7.11E-10 | 2.68E-05±2.44E-05 | 3.93E-245±0.00E+00 | 3.96E-67±7.29E-67 |
| $f_5$ | 2.68E+01±7.02E-01 | 2.71E+01±7.59E-01 | 2.89E+01±3.91E-02 | 2.74E+00±5.64E+00 |
| $f_6$ | 5.76E-01±3.69E-01 | 6.98E-01±3.66E-01 | 6.09E+00±5.38E-01 | 1.75E+00±1.35E+00 |
| $f_7$ | 1.37E-03±6.26E-04 | 2.25E-03±1.19E-03 | 6.66E-05±7.17E-05 | 1.86E-04±1.86E-04 |
| $f_8$ | -5.31E+03±1.24E+03 | -4.94E+03±1.48E+03 | -2.18E+03±4.54E+02 | -2.17E+04±1.53E+03 |
| $f_9$ | 4.76E-01±1.24E+00 | 8.81E-01±2.50E+00 | 0.00E+00±0.00E+00 | 0.00E+00±0.00E+00 |
| $f_{10}$ | 2.14E-14±4.82E-15 | 1.36E+01±9.77E+00 | 3.61E-15±1.53E-15 | 4.44E-15±0.00E+00 |
| $f_{11}$ | 3.61E-03±8.75E-03 | 2.28E-03±5.38E-03 | 0.00E+00±0.00E+00 | 0.00E+00±0.00E+00 |
| $f_{12}$ | 3.84E-02±2.07E-02 | 4.12E-02±1.92E-02 | 1.10E+00±2.11E-01 | 8.93E-02±1.06E-01 |
| $f_{13}$ | 5.25E-01±2.38E-01 | 4.58E-01±1.88E-01 | 2.99E+00±1.37E-02 | 2.63E-02±4.24E-02 |
| Function | mGWO (D=100) (Mean±St. dev) | WdGWO (D=100) (Mean±St. dev) | WDGWO (D=100) (Mean±St. dev) | GWO-WD (D=100) (Mean±St. dev) |
| $f_1$ | 3.70E-16±4.85E-16 | 2.20E-08±1.97E-08 | 0.00E+00±0.00E+00 | 5.59E-130±1.00E-129 |
| $f_2$ | 1.70E-10±6.50E-11 | 6.86E-06±2.44E-06 | 1.55E-245±0.00E+00 | 3.49E-67±4.36E-67 |
| $f_3$ | 4.53E+02±5.78E+02 | 3.65E+03±3.57E+03 | 0.00E+00±0.00E+00 | 1.02E-115±3.39E-115 |
| $f_4$ | 1.85E+00±3.99E+00 | 2.74E+00±2.19E+00 | 1.01E-240±0.00E+00 | 3.43E-62±1.25E-61 |
| $f_5$ | 9.75E+01±8.09E-01 | 9.80E+01±7.29E-01 | 9.89E+01±3.14E-02 | 1.02E+01±2.13E+01 |
| $f_6$ | 9.65E+00±8.42E-01 | 1.06E+01±1.03E+00 | 2.34E+01±8.56E-01 | 1.03E+01±8.17E+00 |
| $f_7$ | 4.07E-03±2.20E-03 | 9.04E-03±3.40E-03 | 7.49E-05±7.46E-05 | 3.42E-04±2.59E-04 |
| $f_8$ | -1.46E+04±3.50E+03 | -9.04E+03±4.24E+03 | -3.96E+03±8.65E+02 | -7.15E+04±4.26E+00 |
| $f_9$ | 7.11E-01±2.44E+00 | 1.24E+01±7.76E+00 | 0.00E+00±0.00E+00 | 0.00E+00±0.00E+00 |
| $f_{10}$ | 1.82E-09±8.80E-10 | 1.75E+01±7.45E+00 | 4.32E-15±6.49E-16 | 4.56E-15±6.49E-16 |
| $f_{11}$ | 1.28E-03±4.88E-03 | 3.62E-03±9.79E-03 | 0.00E+00±0.00E+00 | 0.00E+00±0.00E+00 |
| $f_{12}$ | 2.52E-01±4.85E-02 | 3.49E-01±7.39E-02 | 1.16E+00±5.27E-02 | 2.16E-01±3.84E-01 |
| $f_{13}$ | 6.29E+00±4.47E-01 | 7.05E+00±5.36E-01 | 9.99E+00±2.98E-03 | 3.56E-02±4.78E-02 |
| Function | mGWO (D=1000) (Mean±St. dev) | WdGWO (D=1000) (Mean±St. dev) | WDGWO (D=1000) (Mean±St. dev) | GWO-WD (D=1000) (Mean±St. dev) |
| $f_1$ | 2.06E-02±7.30E-03 | 1.06E+01±3.87E+00 | 0.00E+00±0.00E+00 | 2.13E-119±3.53E-119 |
| $f_2$ | 7.06E-01±7.45E-01 | 4.50E+01±2.37E+01 | Inf±NaN | 8.39E-63±2.26E-61 |
| $f_3$ | 1.63E+06±3.24E+05 | 2.13E+06±5.42E+05 | 0.00E+00±0.00E+00 | 5.85E-107±1.65E-106 |
| $f_4$ | 8.15E+01±3.41E+00 | 7.95E+01±3.29E+00 | 2.89E-237±0.00E+00 | 5.23E-54±2.38E-53 |
| $f_5$ | 1.01E+03±4.63E+00 | 4.48E+03±1.23E+03 | 9.99E+02±3.02E-02 | 7.70E+01±8.74E+01 |
| $f_6$ | 2.13E+02±1.57E+00 | 2.27E+02±6.44E+00 | 2.48E+02±7.47E-01 | 9.85E+01±8.10E+01 |
| $f_7$ | 8.00E-02±2.44E-02 | 4.71E-01±1.22E-01 | 6.97E-05±8.14E-05 | 3.06E-04±2.79E-04 |
| $f_8$ | -7.09E+04±1.51E+04 | -3.10E+04±2.02E+04 | -1.23E+04±3.53E+04 | -7.16E+05±4.77E+03 |
| $f_9$ | 5.51E+01±2.01E+01 | 3.08E+02±6.95E+01 | 0.00E+00±0.00E+00 | 0.00E+00±0.00E+00 |
| $f_{10}$ | 4.57E-03±8.76E-04 | 2.10E+01±1.04E-02 | 4.09E-15±1.08E-15 | 5.39E-15±1.60E-15 |
| $f_{11}$ | 4.22E-03±1.54E-02 | 5.96E-01±1.41E-01 | 0.00E+00±0.00E+00 | 0.00E+00±0.00E+00 |
| $f_{12}$ | 9.98E-01±7.50E-02 | 3.35E+00±1.14E+00 | 1.17E+00±8.91E-03 | 2.81E-01±3.16E-01 |
| $f_{13}$ | 1.08E+02±3.50E+00 | 2.61E+02±5.42E+01 | 1.00E+02±5.84E-03 | 6.75E-01±1.56E+00 |

**TABLE 11. Comparisons between GWO-WD and three GWO variants based on 10 fixed-dimension multimodal benchmark functions.**

| Function | mGWO (Mean±St. dev) | WdGWO (Mean±St. dev) | WDGWO (Mean±St. dev) | GWO-WD (Mean±St. dev) |
|---|---|---|---|---|
| $f_{14}$ | 3.19E+00±3.36E+00 | 1.85E+00±2.50E+00 | 1.19E+01±1.83E+00 | 1.26E+00±6.86E-01 |
| $f_{15}$ | 5.16E-03±8.54E-03 | 1.23E-03±3.62E-03 | 1.02E-02±8.29E-03 | 6.73E-04±3.40E-04 |
| $f_{16}$ | -1.0316±6.02E-08 | -1.0316±1.14E-05 | -0.9794±5.31E-02 | -1.0316±1.38E-05 |
| $f_{17}$ | 0.3984±2.94E-03 | 0.3979±5.64E-05 | 1.1865±6.21E-01 | 0.3989±1.15E-03 |
| $f_{18}$ | 3±2.95E-05 | 3±3.15E-06 | 2.02E+01±1.90E+01 | 3±4.78E-05 |
| $f_{19}$ | -3.8614±2.58E-03 | -3.8610±3.23E-03 | -3.3513±3.59E-01 | -3.8605±2.35E-03 |
| $f_{20}$ | -3.2519±8.88E-02 | -3.2077±9.32E-02 | -1.5177±4.37E-01 | -3.1795±3.71E-02 |
| $f_{21}$ | -8.9732±2.45E+00 | -7.8713±2.98E+00 | -2.1059±8.63E-01 | -10.1488±6.70E-03 |
| $f_{22}$ | -10.3958±4.36E-03 | -9.6005±2.09E+00 | -1.9816±7.16E-01 | -10.3946±1.67E-02 |
| $f_{23}$ | -10.2583±1.48E+00 | -10.3518±9.87E-01 | -1.8709±7.52E-01 | -10.5292±1.24E-02 |

GWO algorithm. In total, these two strategies simultaneously enhance and balance the exploration and exploitation capabilities of the GWO algorithm.

From the Wilcoxon signed-rank test results, the performance of the GWO-WD1 algorithm is better than that of the GWO-WD2 algorithm and worse than that of the GWO-WD

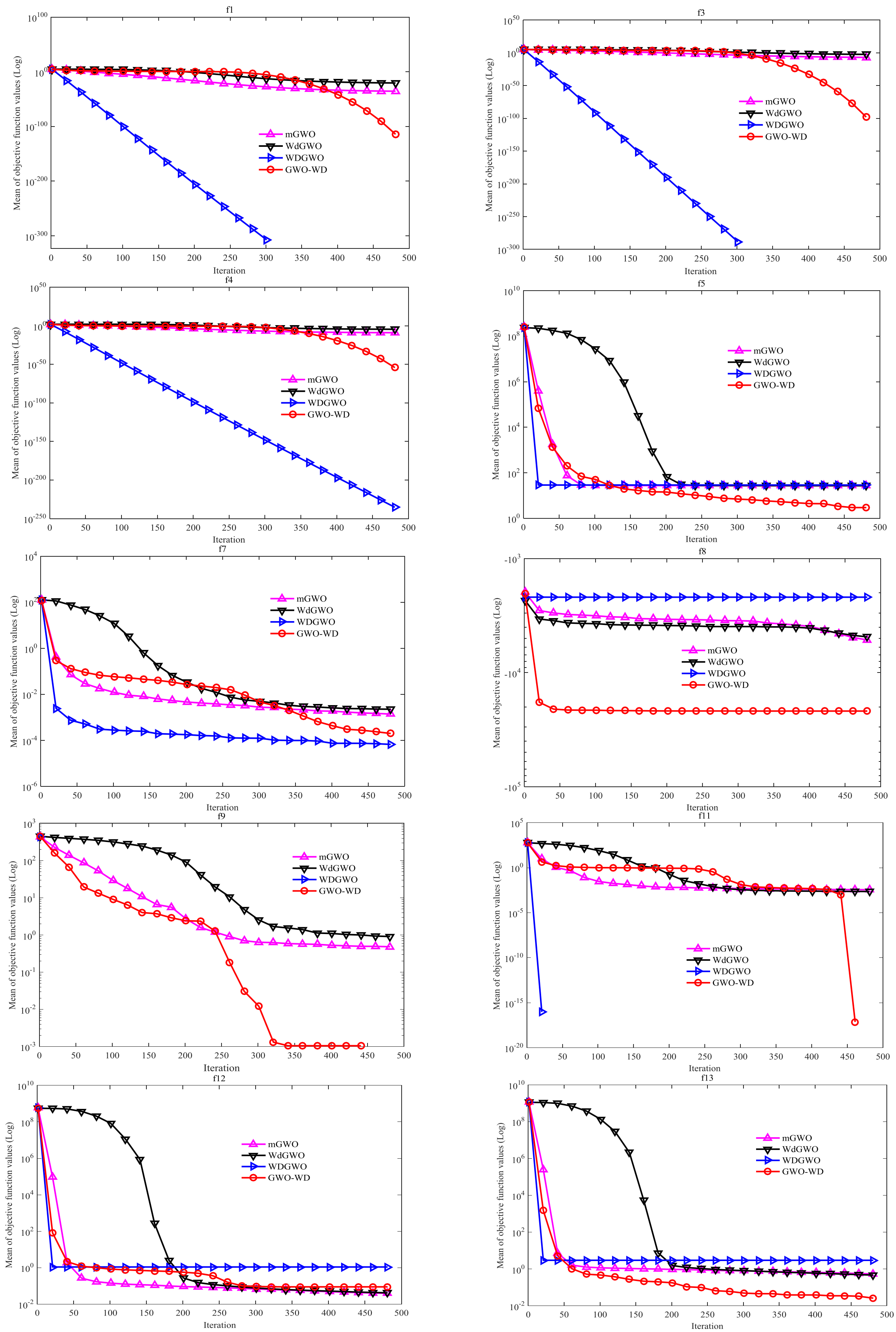



**FIGURE 2.** Convergence curves of GWO-WD and three GWO variants based on several typical functions with D=30.

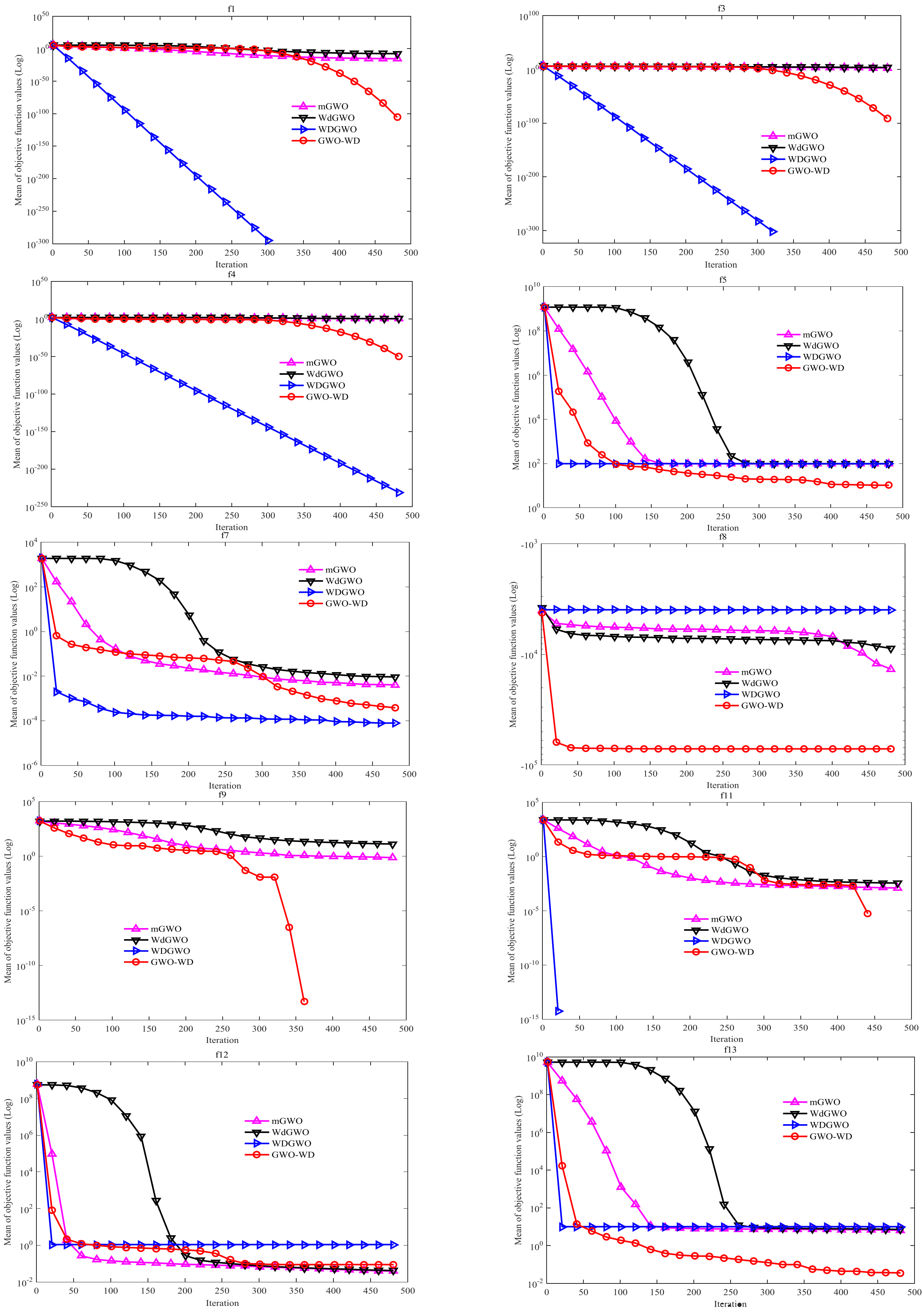



**FIGURE 3.** Convergence curves of GWO-WD and three GWO variants based on several typical functions with D=100.

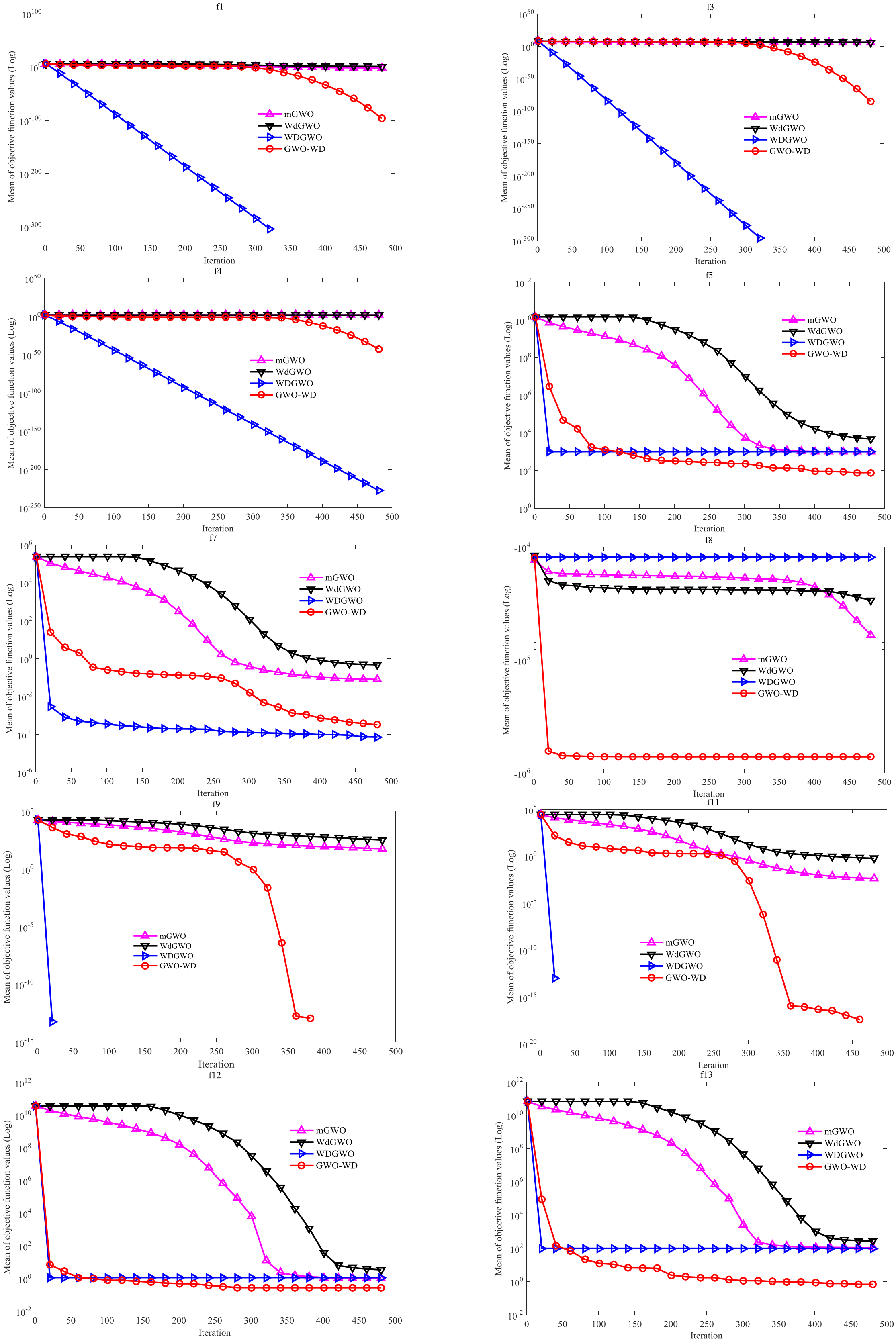



**FIGURE 4.** Convergence curves of GWO-WD and three GWO variants based on several typical functions with D=1000.

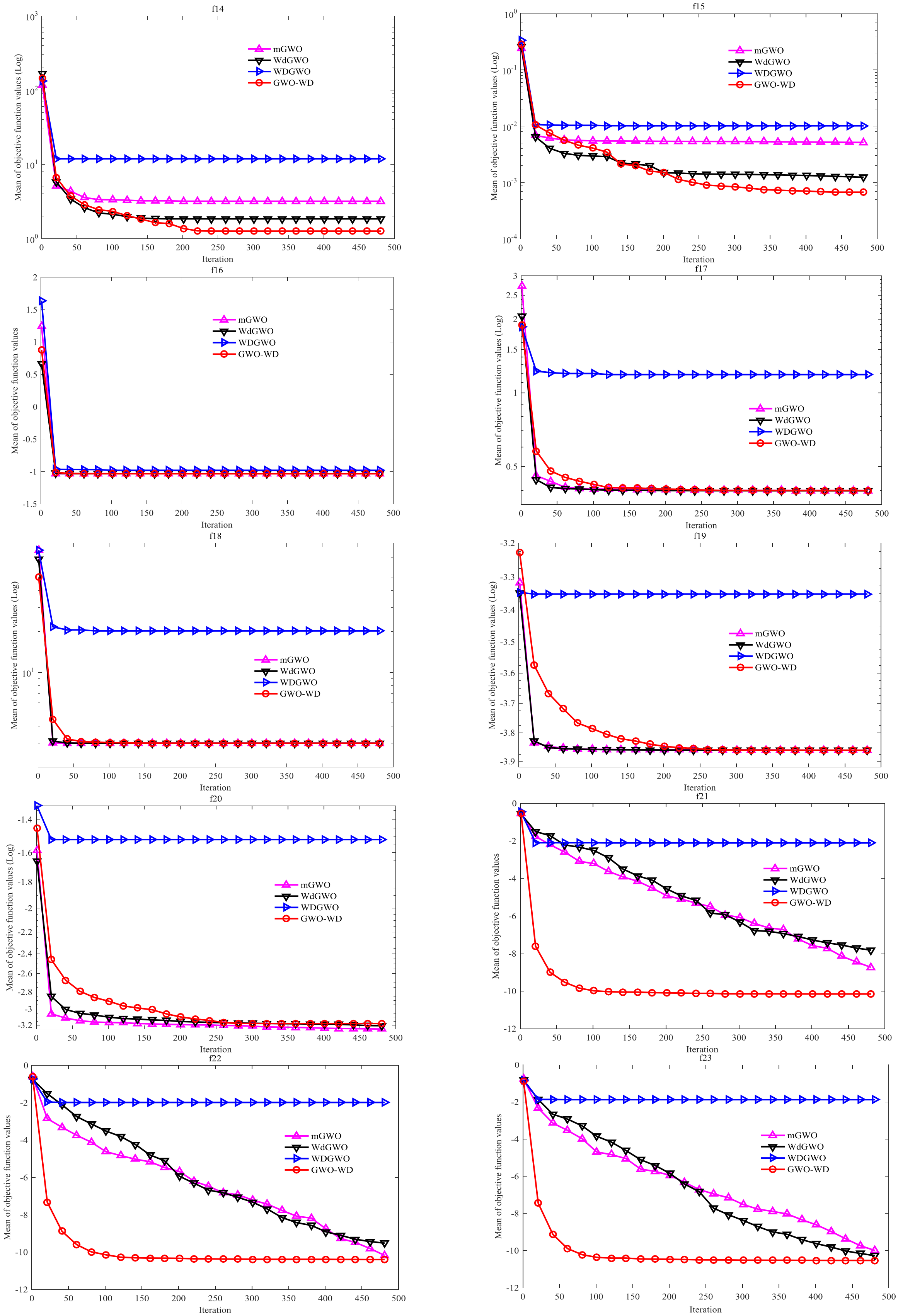



**FIGURE 5. Convergence curves of GWO-WD and three GWO variants based on several typical fixed-dimension multimodal benchmark functions.**

**TABLE 12.** Summarized Wilcoxon rank -sum test results between GWO-WD and three GWO variants.

| GWO-WD vs | mGWO | | | WDGWO | | | WdGWO | | |
|---|---|---|---|---|---|---|---|---|---|
| | Worse | Equal | Better | Worse | Equal | Better | Worse | Equal | Better |
| D=30 | 11 | 1 | 1 | 5 | 2 | 6 | 11 | 1 | 1 |
| D=100 | 12 | 1 | 0 | 5 | 3 | 5 | 12 | 1 | 0 |
| D=1000 | 13 | 0 | 0 | 8 | 2 | 3 | 13 | 0 | 0 |
| Fixed-dimension multimodal | 6 | 1 | 3 | 10 | 0 | 0 | 5 | 1 | 4 |
| Total | 42 | 3 | 4 | 28 | 7 | 14 | 41 | 3 | 5 |

**TABLE 13.** Experimental results of GWO-WD, GWO and GWO-WI.

| Function | GWO | GWO-WI | GWO-WD |
|---|---|---|---|
| $F_1$ | 6.59E-28 | 0 | 0 |
| $F_2$ | 718E-17 | 0 | 0 |
| $F_3$ | 3.29E-06 | 0 | 0 |
| $F_4$ | 5.61E-07 | 0 | 0 |
| $F_5$ | 26.81258 | 23.80734 | 1.38E-04 |
| $F_6$ | 0.816579 | 0 | 1.61E-05 |
| $F_7$ | 0.002213 | 1.03E-06 | 2.51E-06 |
| $F_8$ | -6123.1 | -11876.1 | -21452.2 |
| $F_9$ | 0.310521 | 0 | 0 |

**TABLE 14.** Comparisons between GWO-WD and the CLPSO algorithm, GSA, MFO algorithm, PSO algorithm, and WOA based on 23 test functions.

| Function | CLPSO (Mean±St. dev) | GSA (Mean±St. dev) | MFO (Mean±St. dev) | PSO (Mean±St. dev) | WOA (Mean±St. dev) | GWO-WD (Mean±St. dev) |
|---|---|---|---|---|---|---|
| $f_1$ | 1.26E+01±4.37E+00 | 2.90E-02±.45E-01 | 2.34E+03±5.04E+03 | 2.52E+00±9.48E-01 | 5.66E-74±2.64E-73 | 1.34E-140±2.71E-140 |
| $f_2$ | 9.05E-01±1.80E-01 | 4.02E-03±2.02E-02 | 2.68E+01±1.80E+01 | 4.10E+00±1.08E+00 | 3.56E-51±1.86E-50 | 1.64E-73±2.15E-73 |
| $f_3$ | 1.07E+04±2.33E+03 | 8.64E+02±2.89E+02 | 2.16E+04±1.23E+04 | 1.60E+02±4.67E+01 | 4.57E+04±1.31E+04 | 6.67E-123±1.83E-122 |
| $f_4$ | 3.24E+01±4.56E+00 | 7.10E+00±2.54E+00 | 7.07E+01±8.58E+00 | 2.06E+00±2.73E-01 | 4.05E+01±3.04E+01 | 3.96E-67±7.29E-67 |
| $f_5$ | 6.12E+06±1.50E+06 | 8.17E+01±9.41E+01 | 2.68E+06±1.46E+07 | 9.76E+02±5.55E+02 | 2.80E+01±5.00E-01 | 2.74E+00±5.64E+00 |
| $f_6$ | 1.37E+01±4.50E+00 | 1.02E-02±5.57E-02 | 2.33E+03±5.01E+03 | 2.54E+00±1.32E+00 | 3.26E-01±2.10E-01 | 1.75E+00±1.35E+00 |
| $f_7$ | 5.73E-02±1.75E-02 | 8.42E-02±4.02E-02 | 2.16E+00±4.43E+00 | 1.92E+01±1.48E+01 | 4.85E-03±5.80E-03 | 1.86E-04±1.86E-04 |
| $f_8$ | -1.16E+04±2.80E+02 | -2.57E+03±3.85E+02 | -8.26E+03±8.89E+02 | -6.08E+03±1.24E+03 | -1.08E+04±1.70E+03 | -2.17E+04±1.53E+03 |
| $f_9$ | 4.08E+01±6.59E+00 | 2.85E+01±6.91E+00 | 1.51E+02±3.79E+01 | 1.78E+02±3.76E+01 | 5.68E-15±2.29E-14 | 0.00E+00±0.00E+00 |
| $f_{10}$ | 2.27E+00±3.10E-01 | 1.18E-08±2.85E-09 | 1.58E+01±6.61E+00 | 2.55E+00±4.75E-01 | 3.97E-15±2.42E-15 | 4.44E-15±0.00E+00 |
| $f_{11}$ | 1.12E+00±4.22E-02 | 2.78E+01±7.07E+00 | 2.20E+01±4.54E+01 | 1.21E-01±4.32E-02 | 5.61E-03±3.07E-02 | 0.00E+00±0.00E+00 |
| $f_{12}$ | 6.44E+00±2.62E+00 | 2.29E+00±1.05E+00 | 8.53E+06±4.67E+07 | 6.19E-02±5.13E-02 | 1.73E-02±8.28E-03 | 8.93E-02±1.06E-01 |
| $f_{13}$ | 6.61E+00±3.38E+00 | 8.96E+00±6.51E+00 | 1.37E+07±7.49E+07 | 5.81E-01±2.74E-01 | 5.18E-01±2.29E-01 | 6.75E-01±1.56E+00 |
| $f_{14}$ | 1.16E+00±5.27E-01 | 4.61E+00±3.58E+00 | 3.16E+00±3.08E+00 | 2.94E+00±2.46E+00 | 2.80E+00±3.30E+00 | 1.26E+00±6.86E-01 |
| $f_{15}$ | 7.12E-04±1.94E-04 | 4.25E-03±2.99E-03 | 9.09E-04±3.07E-04 | 8.52E-04±1.70E-04 | 7.64E-04±6.45E-04 | 6.73E-04±3.40E-04 |
| $f_{16}$ | -1.0316±5.30E-16 | -1.0316±4.88E-16 | -1.0316±6.78E-16 | -1.0316±4.79E-16 | -1.0316±7.95E-10 | -1.0316±1.38E-05 |
| $f_{17}$ | 0.3979±0.00E+00 | 0.3979±0.00E+00 | 0.3979±0.00E+00 | 0.3979±0.00E+00 | 0.3979±3.82E-05 | 0.3989±1.15E-03 |
| $f_{18}$ | 3±2.87E-15 | 3±4.48E-15 | 3±1.61E-15 | 3±5.45E-15 | 3±3.88E-05 | 3±4.78E-05 |
| $f_{19}$ | -3.8628±2.67E-15 | -3.8628±2.34E-15 | -3.8625±1.44E-03 | -3.8628±1.99E-15 | -3.8526±2.56E-02 | -3.8605±2.35E-03 |
| $f_{20}$ | -3.2626±6.05E-02 | -3.3220±1.63E-15 | -3.2197±5.56E-02 | -3.2784±5.83E-02 | -3.2201±1.20E-01 | -3.1795±3.71E-02 |
| $f_{21}$ | -6.3923±3.46E+00 | -5.2770±3.44E+00 | -5.7219±3.12E+00 | -7.2202±3.09E+00 | -8.7869±2.54E+00 | -10.1488±6.70E-03 |
| $f_{22}$ | -8.4435±3.09E+00 | -10.1749±1.25E+00 | -7.0007±3.51E+00 | -8.9145±2.78E+00 | -8.1777±2.77E+00 | -10.3946±1.67E-02 |
| $f_{23}$ | -7.7374±3.78E+00 | -9.6095±2.44E+00 | -6.7675±3.63E+00 | -9.6980±2.21E+00 | -7.1458±3.59E+00 | -10.5292±1.24E-02 |

**TABLE 15.** Statistical results of the CLPSO algorithm, GSA, MFO algorithm, PSO algorithm, and WOA based on 23 test functions.

| GWO-WD vs | CLPSO | | | GSA | | | MFO | | | PSO | | | WOA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | W | E | B | W | E | B | W | E | B | W | E | B | W | E | B |
| | 17 | 1 | 5 | 19 | 0 | 4 | 20 | 3 | 0 | 17 | 2 | 4 | 13 | 4 | 6 |

algorithm. Therefore, the combination of the GWO-WD1 and the GWO-WD2 algorithms improves the performance. Specifically, GWO-WD1 improves the exploitation performance of GWO-WD2, and GWO-WD2 enhances the

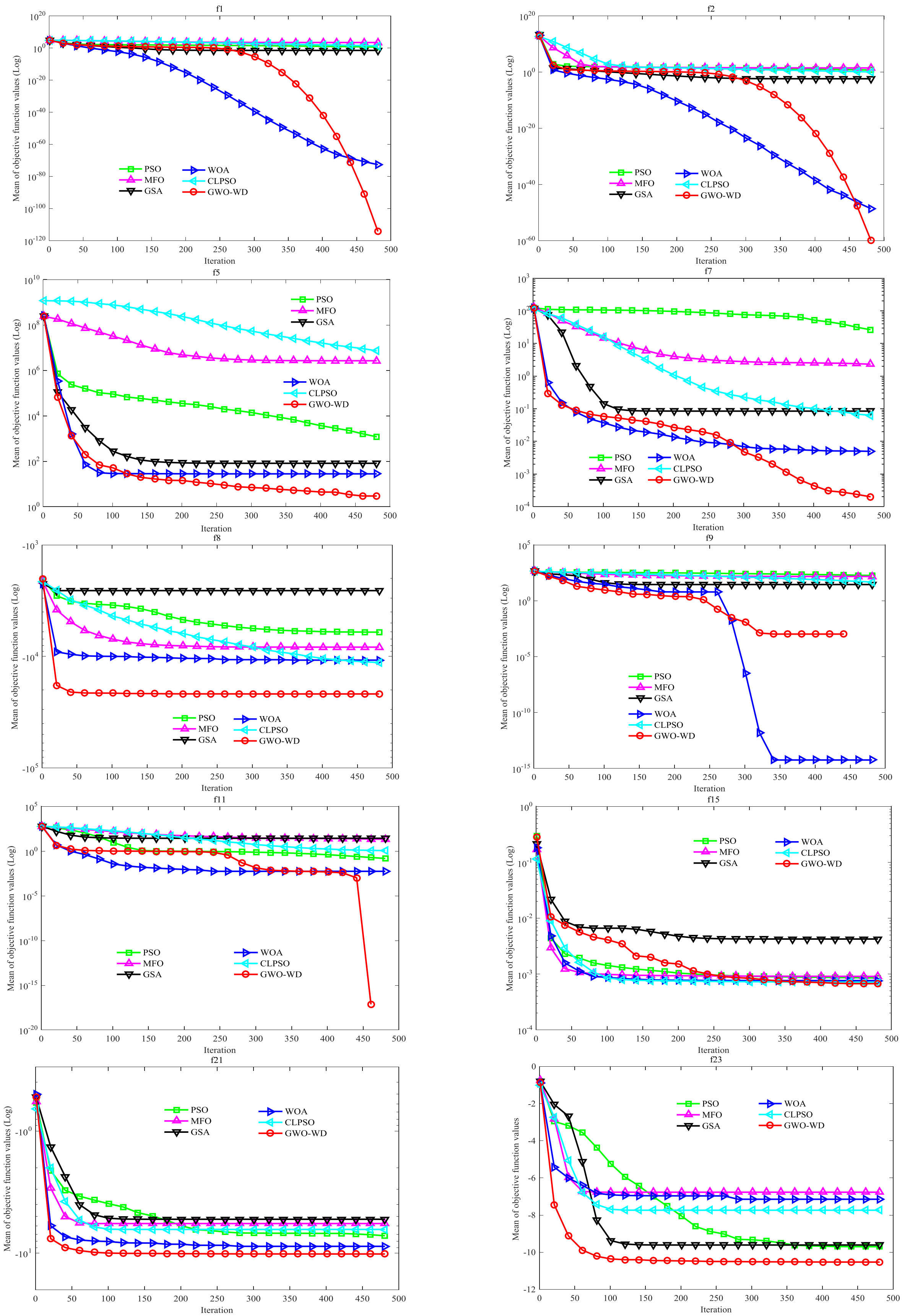



**FIGURE 6.** Convergence curves of GWO-WD and other selected algorithms based on ten representative benchmark test functions.

**TABLE 16.** Comparisons of the IPSO, TACPSO, DGWO, GWO, and DE algorithms on the IEEE CEC 2014 benchmark test suite.

| Fun | IPSO | | TACPSO | | DGWO | | GWO | | DE | | GWO-WD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | St. dev | Mean | St. dev | Mean | St. dev | Mean | St .dev | Mean | St. dev | Mean | St. dev |
| F1 | 7.75E+07 | 2.13E+07 | 1.30E+08 | 9.99E+07 | 5.79E+07 | 2.74E+07 | 8.59E+07 | 6.67E+07 | 1.72E+08 | 3.83E+07 | 6.79E+07 | 1.97E+07 |
| F2 | 2.55E+09 | 2.69E+09 | 1.64E+10 | 1.10E+10 | 2.18E+09 | 2.02E+09 | 2.31E+09 | 2.30E+09 | 2.11E+05 | 5.68E+04 | 1.71E+09 | 3.11E+08 |
| F3 | 8.77E+03 | 7.99E+03 | 5.18E+04 | 2.89E+04 | 3.56E+04 | 9.79E+03 | 3.39E+04 | 8.81E+03 | 6.66E+03 | 5.51E+03 | 2.86E+04 | 1.03E+04 |
| F4 | 6.64E+02 | 8.69E+01 | 1.98E+03 | 1.49E+03 | 6.63E+02 | 1.17E+02 | 6.55E+02 | 1.07E+02 | 5.64E+02 | 1.51E+01 | 6.67E+02 | 8.77E+01 |
| F5 | 5.20E+02 | 2.83E-01 | 5.21E+02 | 1.91E-01 | 5.20E+02 | 5.74E-02 | 5.21E+02 | 5.65E+02 | 5.21E+02 | 5.97E-02 | 5.20E+02 | 7.17E-02 |
| F6 | 6.19E+02 | 3.24E+00 | 6.24E+02 | 2.79E+00 | 6.13E+02 | 2.57E+00 | 6.14E+02 | 3.42E+00 | 6.32E+02 | 1.23E+01 | 6.29E+02 | 2.41E+00 |
| F7 | 7.26E+02 | 1.95E+01 | 8.79E+02 | 7.77E+01 | 7.10E+02 | 1.21E+01 | 1.17E+02 | 1.27E+01 | 7.00E+02 | 1.01E-01 | 7.15E+02 | 2.75E+00 |
| F8 | 8.76E+02 | 1.62E+01 | 9.23E+02 | 2.23E+01 | 8.67E+02 | 2.37E+01 | 8.77E+02 | 2.04E+01 | 8.81E+02 | 6.96E+00 | 9.77E+02 | 1.71E+01 |
| F9 | 1.02E+03 | 3.08E+01 | 1.07E+03 | 2.66E+01 | 9.76E+02 | 2.08E+01 | 9.98E+02 | 2.24E+01 | 1.10E+03 | 1.40E+01 | 1.10E+03 | 1.38E+01 |
| F10 | 3.68E+03 | 6.16E+02 | 4.49E+03 | 5.73E+02 | 3.11E+03 | 4.96E+02 | 3.18E+03 | 6.01E+02 | 2.92E+03 | 2.60E+02 | 7.02E+03 | 4.87E+02 |
| F11 | 4.52E+03 | 5.19E+02 | 5.22E+03 | 6.52E+02 | 3.79E+03 | 5.93E+02 | 3.94E+03 | 7.09E+02 | 7.68E+03 | 2.24E+02 | 7.89E+03 | 4.68E+02 |
| F12 | 1.20E+03 | 3.13E-01 | 1.21E+03 | 3.09E-01 | 1.20E+03 | 1.15E+00 | 1.20E+03 | 1.06E+00 | 1.20E+03 | 2.32E-01 | 1.20E+03 | 2.81E-01 |
| F13 | 1.30E+03 | 8.80E-01 | 1.30E+03 | 1.29E+00 | 1.30E+03 | 1.41E-01 | 1.30E+03 | 1.08E-01 | 1.30E+03 | 7.48E-02 | 1.30E+03 | 1.01E-01 |
| F14 | 1.41E+03 | 1.17E+01 | 1.45E+03 | 2.09E+01 | 1.40E+03 | 4.99E+00 | 1.40E+03 | 5.06E+00 | 1.40E+03 | 6.94E-02 | 1.40E+03 | 3.57E-01 |
| F15 | 1.64E+03 | 2.10E+02 | 4.17E+04 | 7.37E+04 | 1.57E+03 | 4.15E+02 | 1.67E+03 | 4.72E+02 | 1.52E+03 | 1.41E+00 | 1.53E+03 | 3.73E+00 |
| F16 | 5.17E+06 | 7.06E+05 | 1.61E+03 | 5.48E-01 | 1.61E+03 | 6.28E-01 | 1.61E+03 | 7.61E-01 | 1.61E+03 | 2.40E-01 | 1.61E+03 | 3.37E-01 |
| F17 | 6.91E+06 | 2.61E+05 | 4.14E+06 | 2.87E+06 | 2.13E+06 | 3.93E+05 | 2.31E+06 | 3.48E+05 | 8.64E+06 | 3.60E+06 | 4.47E+06 | 2.11E+06 |
| F18 | 1.94E+03 | 3.13E+01 | 9.93E+07 | 2.70E+08 | 2.14E+07 | 3.13E+06 | 9.28E+07 | 2.00E+07 | 8.37E+05 | 5.16E+05 | 2.56E+07 | 6.96E+06 |
| F19 | 5.26E+03 | 2.62E+03 | 2.01E+03 | 6.35E+01 | 1.94E+03 | 3.00E+01 | 1.94E+03 | 2.46E+01 | 1.92E+03 | 2.31E+00 | 1.98E+03 | 3.59E+01 |
| F20 | 5.26E+03 | 2.62E+03 | 1.74E+04 | 1.73E+04 | 1.95E+04 | 9.63E+03 | 2.01E+04 | 7.77E+03 | 1.83E+04 | 7.84E+03 | 1.63E+04 | 6.59E+03 |
| F21 | 1.68E+05 | 1.21E+05 | 7.15E+05 | 8.92E+05 | 1.42E+05 | 2.92E+04 | 7.15E+05 | 9.20E+04 | 1.89E+06 | 1.09E+06 | 1.20E+06 | 6.77E+05 |
| F22 | 2.71E+03 | 5.77E+00 | 2.86E+03 | 1.96E+02 | 2.56E+03 | 1.76E+02 | 2.60E+03 | 1.35E+02 | 2.38E+03 | 9.34E+01 | 2.83E+03 | 2.20E+02 |
| F23 | 2.63E+03 | 1.21E+01 | 2.68E+03 | 3.88E+01 | 2.63E+03 | 1.18E+01 | 2.64E+03 | 1.18E+01 | 2.62E+03 | 4.67E-13 | 2.52E+03 | 3.69E+01 |
| F24 | 2.64E+03 | 1.19E+01 | 2.68E+03 | 1.64E+01 | 2.60E+03 | 1.53E-03 | 2.60E+03 | 1.68E-03 | 2.63E+03 | 1.68E+00 | 2.60E+03 | 1.53E-05 |
| F25 | 2.71E+03 | 5.77E+00 | 2.72E+03 | 5.08E+00 | 2.71E+03 | 5.24E+00 | 2.71E+03 | 4.82E+00 | 2.71E+03 | 1.41E+00 | 2.70E+03 | 2.95E-13 |
| F26 | 2.73E+03 | 6.23E+01 | 2.72E+03 | 6.35E+01 | 2.75E+03 | 5.08E+01 | 2.74E+03 | 4.87E+01 | 2.70E+03 | 4.77E-02 | 2.70E+03 | 9.05E-02 |
| F27 | 3.53E+03 | 2.48E+02 | 3.78E+03 | 8.68E+01 | 3.35E+03 | 1.19E+02 | 3.35E+03 | 1.06E+02 | 3.36E+03 | 8.98E+01 | 3.29E+03 | 3.46E+02 |
| F28 | 4.36E+03 | 3.87E+02 | 4.50E+03 | 6.12E+02 | 3.96E+03 | 2.93E+02 | 4.03E+03 | 3.52E+02 | 3.65E+03 | 2.04E+01 | 3.42E+03 | 4.57E+02 |
| F29 | 2.18E+07 | 1.68E+07 | 1.64E+07 | 1.47E+07 | 1.01E+06 | 2.13E+06 | 6.46E+06 | 2.21E+06 | 6.90E+03 | 7.14E+03 | 5.12E+05 | 1.75E+05 |
| F30 | 6.04E+04 | 6.63E+04 | 1.14E+05 | 7.79E+04 | 4.07E+04 | 2.37E+04 | 4.09E+04 | 1.72E+04 | 6.78E+03 | 1.70E+03 | 5.79E+04 | 3.54E+04 |

**TABLE 17.** Statistical results of the IPSO, TACPSO, DGWO, GWO, and DE algorithms on 23 test functions.

| GWO-WD vs | IPSO | | | TACPSO | | | DGWO | | | GWO | | | DE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | W | E | B | W | E | B | W | E | B | W | E | B | W | E | B |
| | 13 | 5 | 11 | 20 | 4 | 6 | 11 | 10 | 9 | 17 | 2 | 4 | 14 | 2 | 14 |

exploration performance of GWO-WD1. This analysis confirms that both the modified position-updating equation based on the novel weighted distance and the elimination and repositioning strategy are effective.

### F. ADVANTAGES AND LIMITATIONS

As shown from the above experimental results, the GWO-WD algorithm yields better performance for the unimodal and multimodal benchmark test functions compared to the original GWO algorithm and the other selected algorithms. This advantage is mainly due to the proposed modified position-updating equation based on the novel weighted distance, and the elimination and repositioning strategy is effective for improving and balancing the exploration and exploitation capabilities of the original GWO algorithm. However, the proposed GWO-WD algorithm yields unsatisfactory results for fixed-dimension multimodal test functions, indicating a limitation. Therefore, more research is needed in this context.

## V. APPLICATION TO REAL-WORLD ENGINEERING PROBLEMS

To investigate the effectiveness of GWO-WD in real-world engineering applications, we tested the proposed algorithm on three classic engineering design problems, namely, pressure vessel design, welded beam design, and gear train design problems. The parameter values of GWO-WD were set as follows: the population size is 30, and the number of objective function evaluations is 2000. In addition, we adopted the parameter-free penalty function [49] to address constraints as follows.

$$f(Y) = \begin{cases} f(Y); & \text{if } Y \in S \\ f_w(Y) + \sum_{j=1}^{q} g_j(Y); & \text{if } Y \notin S \end{cases}$$

where $S$ denotes the feasible search space, $f_w$ represents the worst feasible solution and $q$ is the number of constraints.

**TABLE 18.** Statistical results of GWO-WD1, GWO-WD2, and GWO-WD on 23 test functions.

| Function | GWO-WD1 | | | GWO-WD2 | | | GWO-WD | |
|---|---|---|---|---|---|---|---|---|
| | Mean | St. dev | | Mean | St. dev | | Mean | St. dev |
| $f_1$ | 3.10E-147 | 5.59E-147 | Equal | 9.76E-20 | 2.00E-19 | Worse | 1.34E-140 | 2.71E-140 |
| $f_2$ | 2.77E-77 | 3.64E-77 | Equal | 3.17E-12 | 2.38E-12 | Worse | 1.64E-73 | 2.15E-73 |
| $f_3$ | 5.73E-126 | 1.95E-125 | Equal | 3.52E-02 | 1.71E-01 | Worse | 6.67E-123 | 1.83E-122 |
| $f_4$ | 2.35E-68 | 2.83E-68 | Equal | 7.49E-04 | 3.43E-03 | Worse | 3.96E-67 | 7.29E-67 |
| $f_5$ | 2.87E+01 | 3.05E-01 | Worse | 3.80E+00 | 7.80E+00 | Equal | 2.74E+00 | 5.64E+00 |
| $f_6$ | 5.53E+00 | 6.24E-01 | Worse | 1.40E-04 | 3.83E-05 | Better | 1.75E+00 | 1.35E+00 |
| $f_7$ | 2.15E-04 | 1.74E-04 | Equal | 2.83E-03 | 1.45E-03 | Worse | 1.86E-04 | 1.86E-04 |
| $f_8$ | -4.75E+03 | 2.75E+02 | Worse | -2.15E+04 | 3.01E+00 | Equal | -2.17E+04 | 1.53E+03 |
| $f_9$ | 0.00E+00 | 0.00E+00 | Equal | 5.49E-10 | 1.89E-09 | Worse | 0.00E+00 | 0.00E+00 |
| $f_{10}$ | 2.78E-15 | 1.80E-15 | Equal | 1.31E-10 | 3.64E-10 | Worse | 4.44E-15 | 0.00E+00 |
| $f_{11}$ | 0.00E+00 | 0.00E+00 | Equal | 3.30E-03 | 8.10E-03 | Worse | 0.00E+00 | 0.00E+00 |
| $f_{12}$ | 9.88E-01 | 2.34E-01 | Worse | 9.05E-03 | 9.91E-03 | Better | 8.93E-02 | 4.24E-02 |
| $f_{13}$ | 2.93E+00 | 8.68E-02 | Worse | 2.53E-04 | 1.53E-04 | Better | 2.63E-02 | 4.24E-02 |
| $f_{14}$ | 1.97E+00 | 2.48E+00 | Worse | 1.26E+00 | 6.24E-01 | Equal | 1.26E+00 | 6.86E-01 |
| $f_{15}$ | 4.12E-03 | 5.57E-03 | Worse | 1.72E-03 | 4.74E-03 | Worse | 6.73E-04 | 3.40E-04 |
| $f_{16}$ | -1.0316 | 6.84E-06 | Equal | -1.0316 | 2.51E-08 | Equal | -1.0316 | 1.38E-05 |
| $f_{17}$ | 0.3989 | 1.65E-03 | Equal | 0.3979 | 4.85E-05 | Equal | 0.3989 | 1.15E-03 |
| $f_{18}$ | 3 | 6.73E-05 | Equal | 3 | 2.11E-05 | Equal | 3 | 4.78E-05 |
| $f_{19}$ | -3.8597 | 3.11E-03 | Equal | -3.8627 | 2.25E-04 | Better | -3.8605 | 2.35E-03 |
| $f_{20}$ | -3.1809 | 6.28E-02 | Equal | -3.2658 | 6.11E-02 | Better | -3.1795 | 3.71E-02 |
| $f_{21}$ | -4.39539 | 1.18E+00 | Worse | -10.1520 | 9.65E-04 | Equal | -10.1488 | 6.70E-02 |
| $f_{22}$ | -5.22569 | 1.52E+00 | Worse | -10.4016 | 1.08E-03 | Equal | -10.3946 | 1.67E-02 |
| $f_{23}$ | -5.45269 | 1.73E+00 | Worse | -10.5351 | 1.08E-03 | Equal | -10.5292 | 1.24E-02 |

### A. PRESSURE VESSEL DESIGN PROBLEM

The pressure vessel design problem, which was introduced by Kannan and Kramer [3], has been widely employed to verify the optimization performance of algorithms for real-world problems. A basic description of the pressure vessel design problem is given in **Figure 7**. For this problem, the objective is to minimize the overall cost, which consists of material, forming, and welding costs.

As shown in **Figure 7**, this problem includes four decision variables, namely, $T_s$ ($y_1$, the thickness of the shell), $T_h$ ($y_2$, the thickness of the head), $R$ ($y_3$, the inner radius), and $L$ ($y_4$, the length of the cylindrical section of the vessel). The model of this problem is given as follows [3].

$$
\begin{aligned}
\text{Minimize}\, f(Y) &= 0.6224y_1y_3y_4 + 1.7781y_2y_3^2 \\
&\quad + 3.1661y_1^2y_4 + 19.84y_1^2y_3 \\
\text{s.t.}\ g_1(Y) &= -y_1 + 0.0193y_3 \le 0 \\
g_2(Y) &= -y_2 + 0.00954y_3 \le 0 \\
g_3(Y) &= -\pi y_3^2y_4 - \frac{4}{3}\pi y_3^3 + 1296000 \le 0 \\
g_4(Y) &= y_4 - 240 \le 0 \\
0.0625 &\le y_1, y_2 \le 99 \times 0.0625; \quad 10 \le y_3, y_4 \le 200
\end{aligned}
$$

**Table 19** shows the best results computed by the proposed GWO-WD algorithm and by various authors [3], [50]–[56]. The experimental statistical results were recorded after 30 independent runs, as shown in **Table 20**. **Table 19**indicates that the performance of GWO-WD is better than that of existing methods, with the proposed method obtaining the smallest overall cost for the pressure vessel design problem. From **Table 20**, the GWO-WD algorithm achieves better "Best" and "Mean" results than the other approaches and the second-best "St. dev." result.

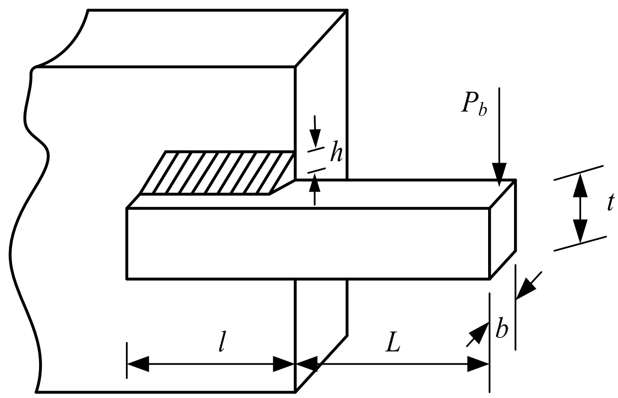



**FIGURE 7.** Structure of the welded beam design problem.

### B. WELDED BEAM DESIGN PROBLEM

The welded beam design problem, which was proposed by Coelho [51], is a well-known engineering design problem that has been commonly utilized as a test problem. The structure of this problem is illustrated in **Figure 8**.

**TABLE 19.** Comparisons of the best results for the pressure vessel design problem obtained by different algorithms.

| Methods | Design variables | | | | Cost |
|---|---|---|---|---|---|
| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $f(Y)$ |
| Sandgren [52] | 1.125000 | 0.625000 | 47.700000 | 117.701000 | 8129.1036 |
| Kannan and Kramer [3] | 1.125000 | 0.625000 | 58.291000 | 43.690000 | 7198.0428 |
| Coello [47] | 0.812500 | 0.437500 | 40.323900 | 200.000000 | 6288.7445 |
| Montes and Coello [51] | 0.812500 | 0.437500 | 42.098087 | 176.640518 | 6059.7456 |
| He.et al. [50] | 0.812500 | 0.437500 | 42.098445 | 176.6365950 | 6059.7143 |
| Coelho [48] | 0.812500 | 0.437500 | 42.09800 | 176.637200 | 6059.7208 |
| Gandomi et al. [49] | 0.812500 | 0.437500 | 42.0984456 | 176.6365958 | 6059.7143348 |
| Akay and Karaboga [46] | 0.812500 | 0.437500 | 42.098446 | 176.636596 | 6059.714339 |
| Present work | 0.784488 | 0.391587 | 40.426617 | 200.000000 | 5969.0261 |

**TABLE 20.** Statistical results for the pressure vessel design problem after 30 independent runs.

| Methods | Best | Mean | Worst | St. dev |
|---|---|---|---|---|
| Sandgren [57] | 8129.1036 | N/A | N/A | N/A |
| Kannan and Kramer [3] | 7198.0428 | N/A | N/A | N/A |
| Coello [52] | 6288.7445 | 6293.8432 | 6308.1497 | 7.4133 |
| Montes and Coello [56] | 6059.7456 | 6850.0049 | 7332.8798 | 426.0000 |
| He.et al. [55] | 6059.7143 | 6289.92881 | N/A | 305.7800 |
| Coelho [53] | 6059.7208 | 6440.3786 | 7544.4925 | 448.4711 |
| Gandomi et al. [54] | 6059.714 | 6447.7360 | 6495.3470 | 502.6930 |
| Akay and Karaboga [51] | 6059.4339 | 6245.308144 | N/A | 205.0000 |
| Present work | 5969.0261 | 6135.8818 | 6882.9915 | 161.1805 |

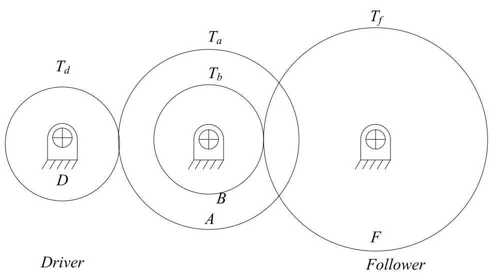



**FIGURE 8.** Structure of the gear design problem.

In this problem, the objective is to find the minimum cost by considering the shear stress ($\tau$), bending stress in the beam ($\sigma$), buckling load on the bar ($P_b$), end deflection of the beam ($\delta$), and side constraints. Four design variables are present in this problem: $h(y_1)$ (the thickness of the weld), $l(y_2)$ (the length of the welded joint), $t(y_3)$ (the width of the beam) and $b(y_4)$ (the thickness of the beam). This optimization problem is constructed as follows.

$$\begin{aligned}
&\text{Minimize } f(Y) = 1.10471 y_1^2 y_2 + 0.04811 y_3 y_4 (14 + y_2)\\
&\text{s.t. } g_1(Y) = \tau(Y) - \tau_{\max} \le 0\\
&\quad g_2(Y) = \sigma(Y) - \sigma_{\max} \le 0\\
&\quad g_2(Y) = y_1 - y_4 \le 0\\
&\quad g_4(Y) = 0.125 - y_1 \le 0\\
&\quad g_5(Y) = \delta(Y) - 0.25 \le 0\\
&\quad g_6(Y) = P - P_c(Y) \le 0\\
&\quad g_7(Y) = 0.10471 y_1^2 + 0.04811 y_3 y_4 (14 + y_2) - 5 \le 0
\end{aligned}$$

where $\tau_{\max} = 13600$psi is the maximum shear stress of the weld, $\sigma_{\max} = 30000$psi is the maximum bending stress, and $P = 6000\ lb$ is the load. The shear stress $\tau$ is modeled as follows.

$$\tau = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{y_2}{2R}\right) + \tau_2^2};\ \tau_1 = \frac{P}{\sqrt{2}y_1y_2};\ \tau_3 = \frac{MR}{J}$$

where

$$M = P\left(L + \frac{y_2}{2}\right);\ J = 2\left\{\sqrt{2}y_1y_2\left[\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2\right]\right\}$$

$$\sigma = \frac{6PL}{y_4y_3^2};\quad P_c = \frac{4.013E\sqrt{\frac{y_3^2y_4^6}{36}}}{L^2}\left(1 - \frac{y_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$\delta = \frac{6PL^3}{Ey_3^3y_4};\quad R = \sqrt{\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2}$$

The GWO-WD algorithm and the methods presented in the literature [50], [52], [53], [55], [57]– [60] were used to solve this problem, and the best results are listed in **Table 21**. The statistical results after 30 independent executions are summarized in **Table 22**. As shown in **Table 21**, the GWO-WD algorithm yielded much better results than the other methods. Furthermore, the statistical results presented in **Table 22** reveal that the GWO-WD algorithm provided the best "Best" and "Worst" results and the second-best "Mean" and "St. dev." results.

### C. GEAR TRAIN DESIGN PROBLEM

The gear train design problem was first proposed by Kaveh and Talatahari [61], and the objective is to find the most

**TABLE 21.** Comparisons of the best results for the welded beam design problem obtained by different approaches.

| Methods | Design variables | | | | Cost |
|---|---|---|---|---|---|
| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $f(Y)$ |
| Coello and Montes [53] | 0.205986 | 3.471328 | 9.020224 | 0.206480 | 1.728226 |
| He and Wang [56] | 0.202369 | 3.544214 | 9.048210 | 0.205723 | 1.728226 |
| Coello [48] | 0.208800 | 3.420500 | 8.997500 | 0.210000 | 1.748309 |
| Dimopoulos [54] | 0.2015 | 3.5620 | 9.041398 | 0.205706 | 1.731186 |
| Montes and Coello [51] | 0.199742 | 3.612060 | 9.037500 | 0.206082 | 1.73730 |
| Kaveh and Talatahari [57] | 0.205700 | 3.471131 | 9.036683 | 0.205731 | 1.724918 |
| Akay and Karaboga [46] | 0.205730 | 3.470489 | 9.036624 | 0.205730 | 1.724852 |
| Gandomi et al. [49] | 0.2015 | 3.562 | 9.0414 | 0.2057 | 1.73121 |
| Present work | 0.203682 | 3.36801 | 9.04577 | 0.205971 | 1.71117 |

**TABLE 22.** Statistical results for the welded beam design problem after 30 independent runs.

| Methods | Best | Mean | Worst | St.dev |
|---|---|---|---|---|
| Coello and Montes [58] | 1.728226 | 1.792654 | 1.993408 | 0.07471 |
| He and Wang [60] | 1.728024 | 1.748831 | 1.782143 | 0.012926 |
| Coello [53] | 1.748309 | 1.771973 | 1.785835 | 0.011220 |
| Dimopoulos [59] | 1.731186 | N/A | N/A | N/A |
| Montes and Coello [56] | 1.737300 | 1.813290 | 1.994651 | 0.070500 |
| Kaveh and Talatahari [61] | 1.724918 | 1.729752 | 1.775961 | 0.009200 |
| Akay and Karaboga [51] | 1.724852 | 1.741913 | N/A | 0.031 |
| Gandomi et al. [54] | 1.7312065 | 1.8786560 | 2.3455793 | 0.2677989 |
| Present work | 1.71117 | 1.739552 | 1.760025 | 0.012755 |

Note: "N/A" denotes data is not available.

**TABLE 23.** Comparisons of the best results for the gear train design problem obtained by different methods.

| | Sandgren [62] | Gandomi et al. [54] | Yan et al. [48] | Garg [63] | Present work |
|---|---|---|---|---|---|
| $T_a(y_1)$ | 18 | 19 | 19 | 19 | 19 |
| $T_b(y_2)$ | 22 | 16 | 16 | 16 | 16 |
| $T_a(y_3)$ | 45 | 43 | 43 | 43 | 43 |
| $T_f(y_4)$ | 60 | 49 | 49 | 49 | 49 |
| Gear ratio | 0.146667 | 0.144281 | 0.14 | 0.14428096 | 0.14428 |
| $f(Y)$ | $5.712\times10^{-6}$ | $2.701\times10^{-12}$ | $2.70\times10^{-12}$ | $2.7008751\times10^{-12}$ | $2.7008571\times10^{-12}$ |

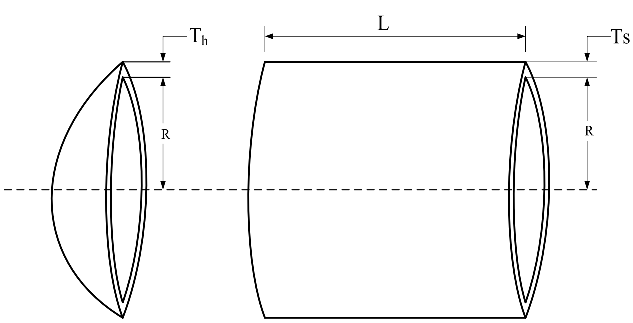



**FIGURE 9.** Structure of the pressure vessel design problem.

suitable number of teeth for a gearwheel between integer intervals of 12 and −60 to minimize the gear cost. The simple structure of this problem is plotted in **Figure 9**.

The mathematical model with decision variables $Y = (T_d, T_b, T_a, T_f) = (y_1, y_2, y_3, y_4)$ is constructed as follows.

$$\text{Minimize } f(Y) = \left(\frac{1}{6.931} - \frac{y_1 y_2}{y_3 y_4}\right)^2$$
$$\text{s.t. } 12 \le y_1, y_2, y_3, y_4 \le 60; \quad y_i \in Z^+$$

where the gear ratio is $\frac{y_1 y_2}{y_3 y_4}$.

The experimental results of the GWO-WD algorithm and those of the methods in the literature [47], [53], [61], [62] are reported in **Table 23**, and the statistical results are presented in **Table 24**. As shown in **Table 23**, the GWO-WD algorithm yielded the best objective function value, and **Table 24**indicates that the GWO-WD algorithm produced better "Mean", "Worst" and "St. dev." results than the other methods. Therefore, the results achieved by the GWO-WD algorithm are significantly better than those found by the researchers in [45], [49], [52], [58].

**TABLE 24.** Statistical results for the gear train design problem after 30 independent runs.

| Methods | Best | Mean | Worst | St.dev |
|---|---|---|---|---|
| Gandomi et al. [54] | $2.70\times10^{-12}$ | $1.98\times10^{-9}$ | $2.36\times10^{-9}$ | $3.56\times10^{-9}$ |
| Yan et al. [48] | $2.70\times10^{-12}$ | $5.65\times10^{-10}$ | $1.36\times10^{-9}$ | $5.72\times10^{-10}$ |
| Garg [62] | $2.70\times10^{-12}$ | $6.25\times10^{-9}$ | $3.30\times10^{-9}$ | $8.77\times10^{-10}$ |
| Present work | $2.70\times10^{-12}$ | $1.19\times10^{-10}$ | $1.26\times10^{-9}$ | $3.10\times10^{-10}$ |

## VI. CONCLUSION

In this paper, a new GWO variant named the GWO-WD algorithm is presented to solve GO problems. In this approach, a novel weighted distance based on the advantages of two different weighted distance strategies is proposed to modify the position-updating equation of the GWO algorithm, and a new strategy is introduced to eliminate and reposition some of the worst search agents. First, the proposed weight distance is applied to modify the position-updating equation of the standard GWO algorithm because this weight distance can provide useful information for solving complex multimodal problems. Then, the elimination and repositioning strategy is employed to remove and reposition several of the worst search agents and increase the probability of avoiding local optima. The performance of the GWO-WD algorithm is next benchmarked based on several GO problems, including 23 well-known benchmark test functions, the IEEE CEC-2014 test suite, and three classic real-world engineering design problems. The simulation results are compared with those of the standard GWO algorithm, three GWO variants and other approaches reported in the literature. The results indicate that the proposed algorithm is effective, robust and scalable when solving low- and high-dimensional, complex unimodal and multimodal problems but has room for improvement in applications involving fixed-dimensional multimodal problems. Convergence curves are also plotted for several classic test functions, and those curves illustrate that the GWO-WD algorithms exhibits a rapid convergence speed. In addition, the application of the GWO-WD algorithm to three classic engineering design problems validates its efficacy at solving practical problems and superiority over other methods. In future research, we will extend the proposed algorithm to solve multiobjective problems, and ELD problems and to train neural networks.

## REFERENCES

[1] D. Jitkongchuen, W. Sukpongthai, and A. Thammano, "Weighted distance grey wolf optimization with immigration operation for global optimization problems," in *Proc. 18th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, Jun. 2017, pp. 1–5.

[2] M. R. S. Malik, E. R. Mohideen, and L. Ali, "Weighted distance grey wolf optimizer for global optimization problems," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2015, pp. 1–6.

[3] 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," *Trans. ASME J. Mech. Des.*, vol. 116, no. 2, pp. 318–320, 1994.

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

[5] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Dept. Comput. Eng., Eng. Fac., Erciyes Univ., Kayseri, Turkey, Tech. Rep. TR06, 2005.

[6] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[7] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.

[8] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proc. Int. Symp. Stochastic Algorithms*, Oct. 2009, pp. 169–178.

[9] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.

[10] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016.

[11] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.

[12] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[13] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[14] L. Chaib, A. Choucha, and S. Arif, "Optimal design and tuning of novel fractional order PID power system stabilizer using a new Metaheuristic bat algorithm," *Ain Shams Eng. J.*, vol. 8, no. 2, pp. 113–125, Jun. 2017.

[15] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm Evol. Comput.*, vol. 44, pp. 148–175, Feb. 2019.

[16] R. Salgotra and U. Singh, "The naked mole-rat algorithm," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8837–8857, Dec. 2019.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[18] Seema and V. Kumar, "Modified grey wolf algorithm for optimization problems," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Aug. 2016, pp. 1–5.

[19] V. Kumar and D. Kumar, "An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems," *Adv. Eng. Softw.*, vol. 112, pp. 231–254, Oct. 2017.

[20] V. Kumar and A. Kaur, "Binary spotted hyena optimizer and its application to feature selection," *J. Ambient Intell. Hum. Comput.*, vol. 11, pp. 2625–2645, May 2019.

[21] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[22] V. Kumar and D. Kumar, "Data clustering using sine cosine algorithm: Data clustering using SCA," *Handbook of Research on Machine Learning Innovations and Trends*. Hershey, PA, USA: IGI Global, 2017, pp. 715–726.

[23] J. Xu, F. Yan, K. Yun, L. Su, F. Li, and J. Guan, "Noninferior solution grey wolf optimizer with an independent local search mechanism for solving economic load dispatch problems," *Energies*, vol. 12, no. 12, p. 2274, Jun. 2019.

[24] M. Pradhan, P. K. Roy, and T. Pal, "Grey wolf optimization applied to economic load dispatch problems," *Int. J. Electr. Power Energy Syst.*, vol. 83, pp. 325–334, Dec. 2016.

[25] R.-E. Precup, R.-C. David, and E. M. Petriu, "Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 527–534, Jan. 2017.

[26] B. Martin, J. Marot, and S. Bourennane, "Mixed grey wolf optimizer for the joint denoising and unmixing of multispectral images," *Appl. Soft Comput.*, vol. 74, pp. 385–410, Jan. 2019, doi: 10.1016/j.asoc.2018.10.019.

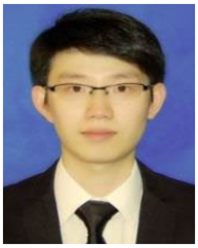
[27] S. Akash, K. Rajesh, and D. Swagatam, "$\beta$-chaotic map enabled grey wolf optimizer," *Appl. Soft Comput.*, vol. 75, pp. 84–105, Feb. 2019.

[28] S. A. Medjahed, T. Ait Saadi, A. Benyettou, and M. Ouali, "Gray wolf optimizer for hyperspectral band selection," *Appl. Soft Comput.*, vol. 40, pp. 178–186, Mar. 2016.

[29] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Adv. Eng. Softw.*, vol. 99, pp. 121–136, Sep. 2016.
[30] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, Jul. 2018.
[31] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, May 2016, Art. no. 7950348.
[32] J. Xu, F. Yan, O. Grace Ala, L. Su, and F. Li, "Chaotic dynamic weight grey wolf optimizer for numerical function optimization," *J. Intell. Fuzzy Syst.*, vol. 37, no. 2, pp. 2367–2384, Sep. 2019.
[33] L. Rodriguez, O. Castillo, and J. Soria, "Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3116–3123.
[34] K. Jaiswal, H. Mittal, and S. Kukreja, "Randomized grey wolf optimizer (RGWO) with randomly weighted coefficients," in *Proc. 10th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2017, pp. 1–3.
[35] A. Kishor and P. K. Singh, "Empirical study of grey wolf optimizer," in *Proc. 5th Int. Conf. Soft Comput. Problem Solving*, Mar. 2016, pp. 1037–1049.
[36] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1257–1263, Jul. 2015.
[37] Q. Luo, S. Zhang, Z. Li, and Y. Zhou, "A novel complex-valued encoding grey wolf optimization algorithm," *Algorithms*, vol. 9, no. 1, p. 4, 2015.
[38] 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 Convers. Manage.*, vol. 133, pp. 427–443, Feb. 2017.
[39] R. Salgotra, U. Singh, and S. Sharma, "On the improvement in grey wolf optimization," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3709–3748, Apr. 2020, doi: 10.1007/s00521-019-04456-7.
[40] J. Xu, F. Yan, K. Yun, S. Ronald, F. Li, and J. Guan, "Dynamically dimensioned search embedded with piecewise opposition-based learning for global optimization," *Sci. Program.*, vol. 2019, pp. 1–20, May 2019.
[41] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 63–80, Feb. 2018.
[42] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.
[43] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
[44] J. J. Liang, B. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep. 201311, 2014.
[45] Z. Cui, J. Zeng, and Y. Yin, "An improved PSO with time-varying accelerator coefficients," in *Proc. 8th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2008, pp. 638–643.
[46] T. Ziyu and Z. Dingxue, "A modified particle swarm optimization with an adaptive acceleration coefficients," in *Proc. Asia–Pacific Conf. Inf. Process.*, Jul. 2009, pp. 330–332.
[47] F. Yan, X. Xu, and J. Xu, "Dynamically dimensioned search grey wolf optimizer based on positional interaction information," *Complexity*, vol. 2019, Dec. 2019, Art. no. 7189653.
[48] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[49] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.
[50] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *J. Intell. Manuf.*, vol. 23, no. 4, pp. 1001–1014, Aug. 2012.
[51] L. D. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1676–1683, Mar. 2010.
[52] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, Mar. 2000.
[53] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A Metaheuristic approach to solve structural optimization problems," *Eng. with Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.
[54] S. He, E. Prempain, and Q. H. Wu, "An improved particle swarm optimizer for mechanical design optimization problems," *Eng. Optim.*, vol. 36, no. 5, pp. 585–605, Oct. 2004.
[55] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *Int. J. Gen. Syst.*, vol. 37, no. 4, pp. 443–473, Aug. 2008.
[56] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design," *J. Mech. Des.*, vol. 112, no. 2, p. 223, 1990.
[57] C. A. Coello Coello and E. Mezura Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Adv. Eng. Informat.*, vol. 16, no. 3, pp. 193–203, Jul. 2002.
[58] G. G. Dimopoulos, "Mixed-variable engineering optimization based on evolutionary and social metaphors," *Comput. Methods Appl. Mech. Eng.*, vol. 196, nos. 4–6, pp. 803–817, Jan. 2007.
[59] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," *Comput. Struct.*, vol. 89, nos. 23–24, pp. 2325–2336, Dec. 2011.
[60] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, no. 1, pp. 89–99, Feb. 2007.
[61] A. Kaveh and S. Talatahari, "An improved ant colony optimization for constrained engineering design problems," *Eng. Comput.*, vol. 27, no. 1, pp. 155–182, Jan. 2010.
[62] H. Garg, "A hybrid GSA-GA algorithm for constrained optimization problems," *Inf. Sci.*, vol. 478, pp. 499–523, Apr. 2019.

**FU YAN** received the B.S. degree in industrial engineering from Dalian Ocean University, Dalian, China, in 2014, and the M.S. degree in agricultural system engineering and management engineering from Northeast Agricultural University, Harbin, China, in 2017. He is currently pursuing the Ph.D. degree in management science and engineering with Harbin Engineering University, Harbin.

His current research interests include evolutionary computation, neural networks, machine learning, operations and management, and their practical applications.



**XINLIANG XU** received the M.S. and Ph.D. degrees in management science and engineering from Harbin Engineering University, Harbin, China, in 2011 and 2014, respectively.

Since 2015, he has been teaching and doing research with the School of Economics and Management, Northeast Agricultural University. His current research interests include upgrading of the whole industry chain, Internet + agriculture, and computation intelligence.

**JIANZHONG XU** received the M.S. degree in industrial economics from the Harbin Shipbuilding Engineering Institute, Harbin, China, in 1993, and the Ph.D. degree in management science and engineering from Harbin Engineering University, Harbin, in 2003.

He is currently a Professor with the School of Economics and Management, Harbin Engineering University. His current research interests include computation intelligence and frontier research in management.